Reason and Representation in Scientific Simulation

Matthew Spencer | PhD Thesis

Centre for Cultural Studies,

Goldsmiths, University of London

Declaration

The work presented in this thesis is the candidate's own.

Abstract

This thesis is a study of scientific practice in computational physics. It is based on an 18 month period of ethnographic research at the Imperial College Applied Modelling and Computation Group. Using a theoretical framework drawn from practice theory, science studies, and historical epistemology, I study how simulations are developed and used in science. Emphasising modelling as a process, I explore how software provides a distinctive kind of material for doing science on computers and how images and writings of various kinds are folded into the research process. Through concrete examples the thesis charts how projects are devised and evolve and how they draw together materials and technologies into semi-stable configurations that crystallise around the objects of their concern, what Hans-Jorg Rheinberger dubbed “epistemic things”.

The main pivot of the research, however, is the connection of practice-theoretical science studies with the philosophy of Gaston Bachelard, whose concept of “phenomenotechnique” facilitates a rationalist reading of scientific practice. Rather than treating reason as a singular logic or method, or as a faculty of the mind, Bachelard points us towards processes of change within actual scientific research, a dynamic reason immanent to processes of skilled engagement. Combining this study of reason with the more recent attention to things within research from materialist and semiotic traditions, I also revive a new sense for the term “representation”, tracing the multiple relationships and shifting identities and differences that are involved in representing. I thus develop a theory of simulation that implies a non-representationalist concept of representing and a non-teleological concept of reason.

Acknowledgements

This is dedicated to Victoria, who made the whole endeavour feasible and life doing it enjoyable.

Thanks to all the scientists at the Applied Modelling and Computation Group, without whom this thesis, and the journey its creation has involved, would have been impossible. It was the desire to do justice to their generous donations of time and attention that drove the project toward completion.

Thanks also to the Arts and Humanities Research Council for financing the project through a PhD scholarship.

Parts of chapter 6 have been previously published[1].

1 Spencer, ‘Image and Practice: Visualisation in Computational Fluid Dynamics Research’; Spencer, ‘Trouble with Images in Computational Physics’.

Table of Contents

1 Introduction.....9
1.1 Beyond simulacra.....9
1.2 Systems of investigation.....10
1.3 A changing world.....12
1.4 Towards a theory of practical reason.....13
1.5 Towards a theory of representation.....16
1.6 Methodological considerations.....17

2 Doing Science With Simulations.....23
2.1 Introduction.....23
2.2 What is simulation?.....23
2.3 The history of computer simulation.....26
2.4 Types of simulation.....29
2.5 Computational laboratories.....30
2.6 Sub-models and coupled models.....33
2.7 History of the Applied Modelling and Computation Group.....34
2.8 Interdisciplinarity.....36
2.9 The finite element method (FEM).....38
2.10 A multiplicity of simulations.....42
2.11 Visualisation and diagnostics.....44
2.12 Verification and validation.....45
2.13 Beyond verification and validation.....50
2.14 Applied modelling and the science of method.....51
2.15 Conclusion.....53

3 Practice in Theory.....54
3.1 Introduction.....54
3.2 Why practice?.....54
3.3 Humans and non-humans.....56
3.4 The significance of practice.....58
3.5 The reality of practice.....60
3.6 Skill.....65
3.7 Temporality.....67
3.8 Objectification.....69
3.9 Conclusion.....70

4 Reason in Practice....71
4.1 Introduction....71
4.2 Internalism and externalism....72
4.3 Empiricism and rationalism....78
4.4 Phenomenotechnique....82
4.5 Reified concepts....84
4.6 Phenomenotechnique in practice....85
4.7 Phenomenotechnique in context....88
4.8 Temporality and reason....90
4.9 Conclusion....91

5 Modelling and Representing....92
5.1 Introduction....92
5.2 Theories of theory: syntax/semantics....93
5.3 Substantive and deflationary theories of representation....95
5.4 From vicarship to realisation....97
5.5 What kind of thing might represent?....101
5.6 What makes a good representation?....104
5.7 Modelling as process....107
5.8 Representation in simulation....110
5.9 New kinds of representation....115
5.10 Conclusion....117

6 Images in/of Simulation....118
6.1 Introduction....118
6.2 Trouble with images....119
6.3 Beyond representation....123
6.4 Discontinuity and process....125
6.5 Images and intuition....126
6.6 Navigating error in the lock exchange....129
6.7 Becoming a scientist....133
6.8 The fate of the image....135
6.9 Conclusion....136

7 Code and Writing....138
7.1 Introduction....138
7.2 Concepts and inscriptions....139
7.3 The many texts of the laboratory....141
7.4 Putting text to work....154
7.5 Software in practice....157
7.6 Epistemology beyond the paper medium....159
7.7 Conclusion....161

8 Workability and Habitability....163
8.1 Introduction....163
8.2 The environment for doing computational science....164
8.3 Software as a material....165
8.4 The troubles with growing software....168

8.5 The scale of software....170
8.6 Making Fluidity robust....172
8.7 Programming systems products....174
8.8 Portability and code generation....177
8.9 Habitability....184
8.10 Conclusion....186

9 Stability and Surprise....187
9.1 Introduction....187
9.2 Regularity and technology....188
9.3 Taming unknowns: approximations....191
9.4 The epistemic thing....195
9.5 Discerning error and event....200
9.6 Surprising results in practice....202
9.7 The philosophy of unanticipated events....204

10 Conclusion....207
10.1 Revelation and invention....207
10.2 Reason and representation....209
10.3 A changed world....211

Bibliography....212

List of Figures

Figure 1: Example of unstructured mesh for complex topography, shown at Fluidity training event in 2011....40

Figure 2: Excerpt from poster about the functionality of the Fluidity code....40

Figure 3: Excerpt from poster displaying functionality of Fluidity....41

Figure 4: A diagrammatic representation of Rheinberger's spectrum....99

Figure 5: Three snapshots of turbulence in a simulation of the lock exchange....111

Figure 6: A visualisation of a time step in a water column collapse simulation....113

Figure 7: Visualisation of data generated by a simulation of the generation of internal waves, and their breaking, in stratified flow over a bump....120

Figure 8: A cross-section of the lock exchange simulation, at different time-steps....130

Figure 9: As in Figure 8, but with mesh overlaid....131

Figure 10: The duck-rabbit....134

Figure 11: Typical scribbles, sketched in order to facilitate explanation of the problem of the "supermesh" (top) and of discontinuous elements (bottom)....142

Figure 12: Changes in disciplinary interface anticipated by the code generation project....182

1 Introduction

1.1 Beyond simulacra

This thesis argues against the relevance of the concept of simulacra for our understanding of simulations in contemporary science. It shows that processes of modelling are not about making copies of the world, but rather about transformative, creative engagements. That the term "simulation" harbours connotations of duplications is a relic of representationalist world views, a present-day version of what Heidegger called the "world picture"[2]. In this misguided frame, simulations appear as the ultimate objectifications of the world: going beyond the drive to map out and collect data, describing in fine detail every inch of the world, simulations would add a new level of rationalisation, that of prediction, extending our grasp to the future and the possible.

While post-positivism in its various guises displaced visions of science as producing one true representation of the world, it provides no guarantee against the return of positivist themes under different names. My account, in grandest terms, is a critique of correspondence, but it extends this critique beyond merely a critique of correspondence theories of knowledge and of truth, looking to also displace the kind of real correspondence, or technical correspondence, that lies behind the view of simulations as simulacra. But more than the negativity of critique, which is often left implicit in what follows, what I wish to accomplish here is to give a positive (not positivist) account of science as practice.

What replaces simulacra is a view of simulation in terms of simulat*ing*, practical engagements that seek genuine novelty, transforming ideas of their object while they transform their own technical foundations. This thesis sets out a view of scientific practice that stresses ontogenesis over ontology, becoming over being. What is important is not the present existence of a state of knowledge that can be forced into a relationship of correspondence with some world, but rather the immanent transformative potentials of research in the making, the intrinsic temporalities that render research in a state of suspension, of hope and anticipation. The simulation itself cannot be extracted from the practical milieux without losing sight of these dimensions. A simulation cannot be an object of study

2 Heidegger, 'The Age of the World Picture'.

on its own, but rather must be understood in a wider frame of its creation, manipulation and multiplication in research.

1.2 Systems of investigation

This thesis is about research. It is about the research that gets done on computers, by scientists who build, run and analyse simulations. As a body of writing, it is the most tangible outcome of an 18-month ethnographic study of the work of a large group of computational scientists at the Applied Modelling and Computation Group at Imperial College in London.

While research may seem a blindingly obvious topic for study, for many philosophers of science it has harboured little intrinsic interest. The possibility of taking research seriously, as a practical process of enquiry, emerged in the wake of post-postivism, buoyed by a loose cluster of approaches that moved past grand theories of science and preferred to look at the smaller-scale worlds of practice. Some inferences may extend outwards from these small scale studies to wider issues about science in general, but such expansion is not the goal. It may be allowed to occur in an *ad hoc* manner. To the extent that this thesis speaks to wider issues, it is to questions of creative practice and questions of software development as much as it is to science. In studying research in computational science, I am primarily interested in what it is to do science on computers. If I can vividly bring those practices to life, I have fulfilled an initial goal.

Placing this enquiry in context: as a result of the focus on practice, I am not very much interested in grand pictures of science that span all the multitudinous disciplinary and interdisciplinary sites where research gets done. Scientific theories take a back seat here, as does knowledge, to a great extent. Taking their place is the *work* done by scientists. In this respect, this study is post-constructivist as much as it is post-positivist, participating what Karin Knorr-Cetina identified as a more recent interest “not in the construction of knowledge but in the construction of the machineries of knowledge construction”[3]. “Magnifying this aspect of science...”, she writes, “reveals the fragmentation of contemporary science; it displays different architectures of empirical approaches, specific constructions of the referent, particular ontologies of instruments, and different social machines”[4]. I am interested in the taking place of research in a major simulation laboratory. But I have to acknowledge that this remains a singular study. I do not have the comparative angle that Knorr-Cetina has been able to bring to bear. Comparison here remains largely implicit, and a project left for the future. This being said, the theoretical angles I bring to bear do lead me to pose, by the conclusion, some tentative but radical formulations of concepts of reality and truth, of reason and representation.

This thesis is a study of practical reason in research, the reason immanent to skilled engagements among people and things, worlds of intricately developed and deeply sedimented materials and

3 Knorr-Cetina, *Epistemic Cultures*, 3.

4 Ibid.

apparatuses. It is also a study of the very real, very concrete role that representation plays in these systems. It is empirical, but also draws on a great deal of extended theoretical discussion, drawing together social scientific theory, continental philosophy and philosophy of science.

If these conceptions of reason, representation, truth and reality comprise my grandest level of engagement, and the descriptive goal of portraying computational science a more humble and empirical level, there is also a third, middle level of enquiry at play, which is to portray simulation as a material practice, and thus to contest conceptions based on the Greek concept of “simulacra”, which tend to idealise simulation as a question of models and copies, an issue of commodification and capitalism[5]. Simulation, in my account, is not abstract, immaterial, or theoretical. It is treated akin to experimental practice, and indeed I draw inspiration from science studies theorists best known for their studies of experimental research (Rheinberger, Knorr-Cetina, Pickering, etc.). If we can understand simulation as *something that happens* in concrete engagements with worlds of materials, then this middle level will have found satisfaction.

Chapter 2, 'Doing Science With Simulations', starts with an extended sketch of simulations in scientific practice. Chapter 2 does not try to offer any theoretical analysis, but aims to give the reader a good foundation for proceeding to the rest of the thesis. From here we move towards two core theoretical chapters, outlining practice theory, science studies, and their relationship with Gaston Bachelard's “new rationalism”. It is by reading scientific practice through Bachelard, that I aim to revive his “non-Kantian” concept of reason in practice. While this will be important for understanding the work of simulationists, it is also an argument about the heritage of science studies.

By chapter 5, 'Modelling and Representing', we start to return to the empirical material, and go through a series of empirical chapters that work these theoretical sources through the ethnographic site. 'Images in/of Simulation' uses images as devices to gain a handle on the accomplishment of research. 'Code and Writing' introduces the question of software as a material for doing science, and deals with questions of its ideality and relationship with other texts. 'Workability and Habitability' follows this thread further by exploring the materiality of systems of research in computational physics, looking at how they mutate over time, and embody certain characteristic conditions for that practice.

I aim to achieve an optimal balance between letting the empirical material speak for itself, while bringing it together with, and allowing it to speak to, key theoretical issues presented earlier in the narrative. The final chapter before the conclusion, 'Stability and Surprise' returns us to more philosophical matters, raising the question of what is scientific and what is technical in research, and introducing a conception of the object of study, twinned with an affective approach to the drives and inspirations that give research its momentum and texture. This chapter aims to integrate the several strands of argument, but also to push their implications further, setting out a reformulated set of

5 The classic text in this respect is Baudrillard, *Simulations*.

concepts appropriate to the creative and immanent ontological viewpoint that lies behind this theory of practice, and which is often implicit elsewhere in the discussion.

1.3 A changing world

The questions that motivate my enquiry don't come from a hope of theorising Science as unity, or even science alone; the answers that emerge may bear on different objects altogether. Motivation instead arises from a changed world, a profoundly changing world. New forms of globalisation and connectivity have consequences across many domains of life. A changing climate threatens human existence through sea-level rise, desertification, ecological collapse. Rather than fall prey to an enlightenment preoccupation with science as a self-sufficient sphere, the pinnacle of human achievement, I turn to the demands of a wider world, an uncertain future, a rapidly evolving milieu, as a source of questions more appropriate to our times. The question is: “How do we act in a technically and environmentally precarious world?” “What is reason in such a world?” For the beginnings of such an enquiry I try to understand simulations, what they are and what they do.

Computational science is a loose association of practices situated at the heart of these matters. It is one of the most radically and rapidly transformed areas of science and engineering, conditional on new digital technologies and subject to the vicissitudes that these technical fields imply. The mediation of digital technologies opens the door to the generation and manipulation of vast mathematical systems, technical systems, and data-sets, but that which brings us closer to the gigantic also holds us apart, obstructing direct grasp of these things and placing new demands on manipulation and comprehension.

Computational science is essential to contemporary paradigms for the study of the climate, of energy generation, and of the study of natural disasters. It is a crucial tool in both policy and technology-driven preparations for the future. It is only going to become more central to the existential issues that define the 21st Century.

My study sticks closely to the work undertaken in one particular site: the Applied Modelling and Computation Group (AMCG). Many different elements are at work here: many materials, many traditions and many spheres of sociality. It is science riding and driving the cutting edge of digital technologies, pursuing old and new questions, developing methods for the next generation of large-scale, high-performance simulations.

What might a Science Studies look like in this context? Far from being an outsider among insiders, the locus of a difference that constitutes the inside and outside of the scientific field proper, the site reveals itself already to be internally divided. Many outsiders mingle among each other. Engineers, mathematicians and physical scientists, with very different backgrounds, histories, training and lexicons, together make up AMCG, a group defined by intense interdisciplinary collaboration. Far from finding “trading zones”[6] at the interstices of different scientific groups, the group here *is* a trading zone: creolisation not the exception but the rule. There is no straightforward identification of the ethnographer as the outsider, for he is one among many outsiders, and the object of study is defined by encounters across boundaries.

A radical version of this idea would position sociological and philosophical studies within rather than outside interdisciplinary (or as some would have it, “transdisciplinary”) science. The study of how research gets done in practice can be seen as one region of the general reflection on method that forms a hard core of many scientists' work (see section 2.14, 'Applied modelling and the science of method'). It may be a relatively exotic region of this territory, speaking often in strange dialects: far flung, but part of it nonetheless.

This posing of a continuum between the observer and the observed is consistent with a general move towards interdisciplinarity within science studies. Ronald Giere, for example, draws on on cognitive science to help him understand the nature of scientific practice[7]. Sociology, anthropology, philosophy, psychology, various sciences, may all have something to offer, depending on how the question is presented. Embracing disunity of science means also embracing the fragmentation of disciplinary divides on the part of the social sciences and humanities, and a methodological refusal of any *a priori* determination of what strategies and traditions are relevant to a problem. This multi-disciplinary openness is what science studies, at its best, can offer[8].

1.4 Towards a theory of practical reason

Having begun with the intention of shifting perspectives from the kind of enlightenment project that would sit science on a pedestal as the one true endeavour, it might seem strange that one of the central themes of this study is the affirmation of the dimension of practice that we might call its “reason”. For many recent theorists, reason warrants suspicion. From Nietzsche, to Freud, to critical theory, not to mention Foucault, Deleuze and Guattari, contemporary theory in the humanities is marked by a general reaction against the enlightenment, inflamed with a desire to look into the cracks of reason, the shadows of the rational subject, bringing madness, dreams, the unconscious and desire into the foreground, bringing to light the less polished side of the coin.

These critiques, however, are oriented towards a broad cultural trope of reason: reason as a great symbol, as Science, as Spirit, as human progress, the inexorable advance of history toward its end. This trope has had a huge impact on culture in general, most notably in its exclusion of “others” thus constituted as irrational. Women, non-Westerners, the old, the young, the ill, have felt in whatever diffuse way the effects of being marginalia in a modern episteme. But it must be stressed that such a critique, however politically important, does not foreclose the possibility of developing a much more

6 Galison, ‘Computer Simulations and the Trading Zone’.
7 Giere, *Explaining Science*.
8 Pickering, *The Mangle of Practice*, 215.

modest concept of reason, one which does not privilege a certain kind of person, a certain type of society, or a particular sort of institution.

I am not interested in treating reason as a faculty of the mind, as if there is some compartment of the cognitive apparatus within which lofty mental functions go about their business. In contrast, reason in this study is something worldly. It is the creativity of skilled practical engagements, working upon themselves toward transformative effect. The manipulation of concepts is here treated on a par with the manipulation of things. A broad historical trend points in this direction, from Bachelard's theory of phenomenotechnique[9], Heidegger's treatment of being-in-the-world[10], to Merleau-Ponty and Bourdieu's writings on the body[11], as well as more recent theories of distributed cognition and the extended mind[12].

Writing of a fellow ethnomethodologist's immersion in a chemistry lab, Garfinkel and colleagues point out that practical competence must be understood as the very instantiation of reasoning rather than a process separate from it. “Schrecker's handling of equipment was not merely an athletic accompaniment of chemical reasoning, it was part and parcel of chemical reasoning (in the same way that playing a musical instrument is not an athletic accompaniment of music but the existence-in-the-production of everything that music could be)”[13]. The key point here is that in displacing intellectualist treatments of reason, we reinsert action itself into a complex nexus of practice, leaving no space for what could be described as “brute doings” considered apart from the meaning and reason their accomplishment embodies.

Practical reason, in this vein, is something embedded in complexes of technology and social interaction. Reason does not admit of a universal logic or a structure of thought. It is material, technical and bodily as much as it is cerebral. To study reason, in this case, is to study that complex domain I would call “practice”. Far from reason being a question of a commerce between concept and reality, practice undermines the very basis of this dualism, as Rouse has provocatively asserted:

> “Cultural studies of science play a pivotal role... because of the importance of *scientific practices* within the contemporary dualism of intentionality and nature. If Descartes's account of mind and body is the paradigmatic dualism, then scientific practices are the pineal gland of contemporary philosophy: the proposed site for the magical reconciliation of what has been conceived irreconcilably”[14].

It is necessary to point out from the start that the kind of rationalism advanced here does not provide any great grounds for intervention in the debates over rationality and relativism[15]. Such classic exchanges generally revolved around the giving of reasons for belief in certain propositions, from

9 Bachelard, *The Formation of the Scientific Mind*.

10 Heidegger, *Being and Time*.

11 Merleau-Ponty, *Phenomenology of Perception*; Bourdieu, *Outline of a Theory of Practice*.

12 Clark and Chalmers, ‘The Extended Mind’; Hutchins, *Cognition in the Wild*.

13 Lynch, Livingston, and Garfinkel, ‘Temporal Order in Laboratory Work’, 227.

14 Rouse, ‘Understanding Scientific Practices’, 443.

15 See, for example, Hollis and Lukes, *Rationality and Relativism*.

witchcraft accusations among the Azande to the deductions of Western logicians[16]. Treating reason as a property of skilled practical engagements does not require commitment to any particular relativism or universalism because it is not first and foremost a question of belief or language. There may be universal features of certain forms of practice. There will certainly be elements of practice that are relative to their circumstances. But in either case, reason is part of the generative milieux, rather than an external method or faculty in a position of adjudication.

Furthermore, while a practical rationalism allows no absolute division between the rational and the irrational, it does permit relative distinctions to be made. Practical reason is strengthened by experience. Not by subjective experience, what some would call “sense perception”, but rather by experience in the sense of “being experienced”, what Germans would call *erfahrenheit*. “Being experienced enables us to literally embody the judgement in the process of making new experiences, that is, to think with our body... *Erfahrenheit*, that is, acquired intuition, is a form of life”[17]. Scientists' work works in, and thus upon, their fields of practice, a deep conditioning of intensities, reflexes and intuitions. This same work equally works upon the grain of the material and social environment, something not usually captured in the term “experience”. We could thus talk of well-developed and less-developed fields of practice, not in the sense of there being some external metric against which they can be held up and compared, but rather in the sense that some fields have a persistence and coherence to them, developing internal dynamics unfolding trajectories and intensities, while other have no such longevity, are less coherent and much more fleeting.

This view of practical reason does not divide things up in the way that we are used to. For a start, while most circumstances of science in action involve highly developed fields of practice, so too would many other arenas of life. Art, craft and ritual equally often exhibit highly developed systems of practical reason[18]. But this is important, and it raises an open question: a practice theory must embrace cultural diversity, and avoid *a priori* assumptions that science is the primary or only forum in which reason is realised. Distinctions between science and non-science are made all the time, not least by scientists themselves, but while it is the business of social scientists to understand these distinctions, how they are made, what their effects might be, we also need to be open to the possibility that the continuities and commonalities between scientific practice and the practical engagements manifest in art, craft and ritual practice may provide the greatest source of insight into how experience, technology and the body form wider cultural complexes with intrinsic dynamics of their own.

16 Evans-Pritchard, *Witchcraft, Oracles and Magic*; see also, Grieffenhagen and Sharrock, ‘Logical Relativism’.

17 Rheinberger, *Epistemic Things*, 77.

18 Cf. Rheinberger, ‘Experimental Systems’, 68, n. 11.

1.5 Towards a theory of representation

If there is any concept that rivals reason for its subjection to sustained fire from Cultural Studies and Continental Philosophy, it is representation. It should be obvious that I don't endorse a picture of science in which science on some grand level produces a representation of reality. With the displacement of scientific theory as the key locus for study, achieved by the “experimental turn” in philosophy of science[19], it is no longer clear what part of science would be doing this singular task of representing. Once condemned as producing a “world picture”[20], it is now clear that science can only be grasped in such a fashion through a framing that strips away the diversity and specificities of its practices, stamping it with an essence.

When continental philosophers such as Heidegger, Foucault, Ranciere and Baudrillard provide their various critiques of representation, their enemy is the grand cultural trope in which science is a powerful unified apparatus for the creation of a maximally adequate picture of reality, a “mirror of nature”[21], a “world picture”, or what Paul Teller later came to dub “the perfect model model”[22]. Continental critiques often play on the political connotations of the term “representation”, offering a critique of the legitimating strategies of democratic political forms which draw on the notion of a government representing its citizens[23]. Ranciere, for example, would not only point out that there are gaps in representation, people who happen to be un-represented or mis-represented, but would go further and stress that a fundamental lack is necessarily present in any form of representation, that representation works through exclusion[24]. Similarly, in Heidegger's view of modern science as the culmination of metaphysics in its production of the “world picture”, it is not the specific adequacy of the picture that is to be scrutinised, but the dimensions of existence, that it excludes *a priori*: *poeisis*, for example[25].

Continental philosophers may, however, stand to learn something from their English-speaking brethren. The problem of representation as it occurs within Anglo-American philosophy of science is rarely posed according to a big-picture narrative about pervasive cultural tropes spanning art, politics, science. We can find among some of these writers a concern constituted alongside that of the scientist, a general guiding principle that philosophy must stay true to the ways in which the concept is used in practice, rather than studying what is essentially the philosophers' own abstraction. Hence Mauricio Suarez will critique structuralists like Steven French for failing to account for the full breadth of kinds of representation we find in the world[26]. The object of study is

19 See, for example, Hacking, *Representing and Intervening*.
20 Heidegger, ‘The Age of the World Picture’.
21 Rorty, *Philosophy and the Mirror of Nature*.
22 Teller, ‘Twilight of the Perfect Model Model’.
23 For a history of the term’s usages, see Pitkin, *The Concept of Representation*; see also Latour, *We Have Never Been Modern*, 27.
24 See, for example, Ranciere, ‘Democracy, Republic, Representation’.
25 Heidegger, ‘The Question Concerning Technology’, 34–35.
26 Suárez, ‘An Inferential Conception of Scientific Representation’, 768.

not an historical *a priori* or cultural unconscious. It is the diverse assembly of human endeavours that get called representation.

Within the context of scientific computing, it is very hard to see how you can understand everyday practices of description, depiction, simulation, modelling and visualisation, without getting to grips with the way in which many things represent each other, and how these relations are put to work. On this very much more mundane level, representation has little to do with producing a universal representation of the world. For a start, representations are not “outside” the world, and in most cases they work precisely because they are not true[27]. The laboratory does not produce, or contribute to, a single great picture of reality, but rather is the site of production of many different representations, which are folded into its circuits of practice in many different ways. Their formation and manipulation is an important axis of work in the laboratory and their multiplication and dissemination renders them one of the most important elements of the scientific material culture in which research gets done.

In chapter 5, I develop what could be called, following Suarez, a “deflationary” and “pragmatic” view[28], but I try to combine this with insights from the “material semiotic” tradition in STS. Rather than offer some general theory of representation I find it more appropriate to give a theory of represent*ing*, one which refuses to take as its object a binary relation between source and target, and instead regards its object as a complex nexus of many elements: source and target, but also agents, communities, projects, uses, intentions, conventions. It is here that real representations are devised, posited, developed, sustained, worked through, passed on and discarded. It makes little sense to say that representations are “true”, but they can turn out to be “good” in many ways, in their various roles as materials for research. This being said, once I develop, in chapter 5, an account of representation, I go on to study images and texts in the following chapters as always *more* than representations. This excess is what the notion of practice is supposed to embrace. Representations in practice, just like objectifications of practice, imply an escape, an excess, and for this we turn beyond representation to materiality (chapter 8), temporality (chapter 6), and lack (chapter 9).

1.6 Methodological considerations

This thesis is based on an 18-month period of ethnographic research with the Imperial College Applied Modelling and Computation Group, from August 2010 to March 2012. While I call this research “ethnographic” it must be noted that the intensity of contact was far less than what would be conventional for anthropological ethnographic studies. I did not live with scientists, nor did I spend every day with them. I visited once or twice a week, and attempted to become involved in all manner of routine activities. I attended weekly developers' meetings on Wednesday lunchtimes for a period of 8 months, and outside this time I attended less regularly, but enough to keep up to date.

27 See, for example, Knuuttila, ‘Models, Representation, and Mediation’, 1263.

28 Suárez, ‘An Inferential Conception of Scientific Representation’; Suárez, ‘Scientific Representation’.

These meetings gave me a good insight into the longer term fluctuations in the mood of the group, the comings and goings of different staff members, and the seasonal differences in intensity between release dates, funding deadlines, term-time pressures, and vacations.

I also attended project-specific meetings for the multiphase flow research group and the nuclear transport research group over the summer of 2011. In November 2010 and November 2011 I went to the annual three-day training event for the “Fluidity” fluid dynamics code, on the first occasion as an observer, while on the second I enrolled as a new trainee, alongside external collaborators and scientists new to the group, during which I learned to create meshes, to compile the code, to run a variety of example simulations, to manipulate visualisations of the data I thus generated, and to modify the simulation set-ups I had been using. I also attended training sessions for the developers of Fluidity, in which I learned the architecture of the software and the basics of how it works.

From the beginning I was connected to the community through digital means, and received daily updates from their email lists, for support and social issues. I logged in, on occasion, to the AMCG chat channel. All sorts of texts were available to me. I browsed the source code of their open source software, looked at descriptions of new commits and branches of the code. I examined bug reports and testing updates. All of these things helped me gain a good comprehensive knowledge of their research. I was also helped by having worked for a software company for a year prior to starting my own research, so I was already familiar with many of the software technologies and management systems, as well as with the general practical issues that surround working with code.

I went to about 15 public seminars in which members of the group presented their research, and attended a conference in which a few were doing the same. These were good places to hear about new projects, as well as to understand how research was being presented to outsiders. It was also very interesting to follow comments and conversations in these public events, and to reflect on how and why different features of the research are flagged up for comment or critique. I attended a seminar in which the group presented their work to the Natural Environment Research Council (NERC), as part of a project run by the research council in which they wanted to visit research groups to gather information on next generation methods. At yearly meetings to introduce the group to new PhD students and postdocs I was invited to introduce myself, which made me feel, after a few months, more of a part of things.

As well as these academic occasions, I went to lunch many times with members of the group, during which they would often discuss academic matters, as well as the social issues consequent of being part of a large group. I also joined in with coffee breaks when I was around. And I was very pleased to be invited, and able to attend, one of the group's Christmas dinner parties. These occasions helped greatly with my general understanding of this kind of science, but they were also hugely important in getting as many people as possible out of this very large research group to know who I am and to accept me as an unthreatening presence. After a few months I became a somewhat normal presence within the group, rather than the curiosity I must have seemed at the beginning.

When I first started doing this research I was invited to give a seminar on my work, in which I could propose my project to the group, and hopefully muster enough enthusiasm that I would be invited to become a kind of resident ethnographer. Though I had thought this would be a small occasion, when I arrived and saw the size of the room and for the first time appreciated the size of the research group, I realised it was something much larger. It was daunting to address scientists at that stage in my research, but it was soon apparent that an exercise of this nature was absolutely essential for me to begin my empirical study. Because so much of my observation was done in communal settings, I felt it was necessary for everybody to know who I was and what I was doing. There are so many new PhD students in the group at that time of year that I would otherwise just be assumed to be one of them, and the scientists around me would be unaware that someone in the room had them as his object of study, an ethically uncomfortable covert presence.

I call this study ethnographic because it was on the basis of these long-term and diverse experiences that I started to feel that I understood something about what it is like to do computational science. The majority of the empirical “material” presented in this thesis, however, is excerpts from transcripts of interviews, of which I did 32, amounting to a total of over 50 hours of recorded material. I use this material in order to give voice to my informants and because it seems better to allow them to speak in their own words wherever possible. But I do not think of interviews as data-collection enterprises. I do not think that interviews on their own are particularly effective as devices for gaining empirical understanding of research practice. It is only on the basis of the wider sweep of my ethnographic encounter that I could appreciate what was said in the interview circumstances.

Methodologically speaking, I want to distance myself from qualitative approaches in which interviews are the core focus, and in which transcripts are analysed for modes of expressions, words and phrases counted, the numbers crunched (this kind of “coding-based” approach is common in cognitive studies of science and grounded theory). For me, so much is already lost before we even get to the transcript. Each interview is wholly different in type and tenor. The different scientists I interviewed had very different stances towards me and towards each other, something that is only appreciated when observed over many months.

When I cite interviews I provide the text in a somewhat artificial format. Interviewees do not really speak for themselves, and I don't think it is right to present them as if they do. I speak for them, or at least, I give them the space for their expressions. I control these spaces, and I must take responsibility for this. I provide these snippets of conversation without the context of my questioning because I mostly quote small sections at a time, and because my interviews were very loosely structured. I tried to avoid questions that required a yes or no answer, and focussed on getting my informants talking on a topic, interjecting only to keep them going, to show that I understood what they were talking about, and occasionally to guide the conversation towards other topics that I wanted to approach. I quote without giving my question because the interesting parts of the dialogue are often those in which the interviewee has gone “off topic”, so for most of the text I use, the question doesn't really help to give the context. I also do this because I want to avoid giving the impression of objective recounting of events. As the initiator and appropriator of these conversations, I take responsibility for their use, and it seems only right to embrace the authority that the ethnographer adopts in presuming to “speak for” his informants[29].

My interviews do not form a set; they do not all follow the same pattern, and did not involve identical questions being asked. The scientists at AMCG are very varied in their professional interests, in their methods of working, their backgrounds, and in their philosophical interests. I was not particularly interested in comparing different views on the same question (as if the same question in different circumstances is ever really the same question). Rather, I wanted to explore a range of questions, tailored to the interviewee. I began all interviews by taking five minutes to explain my project and clarify any questions the interviewee might have about what I was doing. Then I went on to ask them what kind of research they are involved in, the history of their work on these kind of topics, what the work involved on a practical day-to-day level, and further open questions, for example about what the major issues or problems are that they are facing in their research at present or in the future.

It was not easy to find an appropriate empirical site, and it took some work to get it established. I began my fieldwork less than a year after the famous “Climategate” scandal involving the hacking of scientists' emails at the University of East Anglia. While the scientists at AMCG are not working in exactly the same field, there was a general sense of defensiveness about computational science, a result no doubt of the very public challenging of this kind of science in mainstream media.

My appearance at the group was made no easier by my informants' lack of prior knowledge of ethnographic studies of science. It has now been over four decades since laboratory studies became widespread, and it is not uncommon that in any scientific context, someone within the laboratory will have met, or will have known someone who has met, a social scientist involved in this kind of work. A preliminary interview with a scientist at University College London, which I conducted before selecting AMCG as my field site, was made much easier by the fact that this interviewee's father had been involved with Harry Collins' ethnographic studies many years previously. He therefore knew very well what it was that social scientists did when they studied scientific practice.

So despite having to begin from scratch in explaining what it was that I was actually proposing when arriving at ACMG, and despite a general wariness of public opinion of their science, and about what it was that I might be making public about their science, I was lucky to quickly gain acceptance. This was largely thanks to the effort of a small number of individuals who were prepared not just to allow

29 See, for example, Clifford and Marcus, *Writing Culture*, 13.

me in, but to positively encourage me, and who went to the trouble of clearing my presence with the Head of Department, getting me an identity card, and thereby giving me some semi-official status as part of the group.

There was some resistance to my work from some quarters. One or two scientists clearly couldn't see what they would gain from my presence, but were well aware of the trouble I could cause. If I were to be in the business of criticising their practices, or misrepresenting them somehow, it could adversely impact their ability to gain funding and collaborators. There were some parts of the group to which I never gained much “behind the scenes” access, and in these cases, because they were in the minority, I didn't push too hard. Among the largest part of the group, those involved with the “Fluidity” fluid dynamics code, my involvement was facilitated by the fact that this software was just becoming open source while I was there, which meant that all the documentation, all the email lists, all the chat interactions, and so on, were to be made open to external users and developers. So the inner workings of the group would be publicly available anyway, and if anything, an ethnographer hanging around would help them come to terms with their new openness.

In some cases, individuals seemed slightly hostile to me, probably because of my audacity to think I could legitimately have something to say about their work. In these cases I found the best policy was to contact them for an interview as soon as possible, to give them a chance to interrogate me in private about what I am doing. Most scientists clearly enjoyed talking about their work, and to have an interviewer there who is primarily interested in them talking about what interests them (rather than reeling questions off a pre-set questionnaire) seemed to be compliment enough that no hostility persisted.

My strategy for gaining acceptance within the group involved reassuring my informants about my intentions. The most important dimension of this reassurance was the promise from the outset that I would use pseudonyms so that it would be very hard for anybody to track down who had said what. This tactic also had its downside, however, because it is only a small leap to conclude that if pseudonyms are necessary, I must be interested in scandal. I hedged against this by pointing out that it is standard practice in social scientific studies to maintain the anonymity of sources.

I developed a system of false initials, which I use throughout the thesis. This is not as personal and evocative as giving false names, but I don't stake much of my analysis on the ability to evoke personalities. Where names were mentioned in text taken from interviews or other sources, I have swapped them for these initials, without square brackets because the substitution is obvious.

Anonymity could present a problem in some cases for the attribution of genuinely academic opinions to the person who has voiced them. This does not present too much trouble here, but it is worth bearing in mind that further studies into the nature of modelling and into the practices of simulating may well cross a line in which the ethnographic account blurs together with the kind of meta-reflection on their practices that is occasionally published by computational scientists in their own computational science journals[30]. In my case, I am satisfied that where the merits of scientists' reflections on their own practices are apparent, it is enough that credit be attributed to AMCG, which is not a pseudonym. It is the real name of the real group, and it stands for the arena in which these opinions have been nurtured and developed.

Computational science conventionally recognises the social nature of research with what are sometimes absurdly long lists of authors on their published papers. The scientists at AMCG are in some sense co-responsible for what I write here too, but we have no similar convention in which to embrace their participation, except for paying methodological lip service. Nevertheless, it is important to firmly maintain that this account emerges from a very specific set of interactions with a particular group of people. At points it perhaps speaks to a wider domain, to computational science in general, or to science in general, but the real centre of this study is that one group, that one place and time, and I don't doubt that any similar study conducted elsewhere would on so many pages put a somewhat different spin on things. But this particularity is not to be acknowledged as a limitation, confining the text to the particular rather than the universal. It is an asset, a grounding, a point of departure to which it is important to remain true.

30 See, for example Randall and Wielicki, ‘Measurements, Models, and Hypotheses’.

2 Doing Science With Simulations

2.1 Introduction

This chapter sets the scene by introducing in broad outline what is involved in doing science with simulations. I aim to give an overall picture of the kind of science that has been the object of my study. I provide an overview of different kinds of simulations, how they are used, and the history of scientific computing. I also introduce AMCG, the work these scientists do, and the basic components of their laboratory. The theoretical chapters that follow will be more illuminating in light of this extended empirical introduction, and when we get to the empirical analysis that forms the latter half of the thesis, this chapter will have provided background needed so that we can jump straight in to more complex issues.

2.2 What is simulation?

Stephan Hartmann listed five main reasons why scientists run simulations:

> “1. Simulations as a technique: Investigate the detailed dynamics of a system
> 2. Simulations as a heuristic tool: Develop hypotheses, models and theories
> 3. Simulations as a substitute for an experiment: Perform numerical experiments
> 4. Simulations as a tool for experimentalists: Support experiments
> 5. Simulations as a pedagogical tool: Gain understanding of a process”[31]

Hartmann's categories give a good overall sense of the range of roles that simulations can play. They are not particularly useful as a typology of different simulation-situations, for the items in the list are found in combination. It is this combination that helps us gain an introductory grasp of the “epistemic culture” of a research group or research area[32]. The simulations I studied were most obviously used in the senses 1 and 3, which are often combined. Simulations allow the investigation of the detailed dynamics of a system because they are not limited by the practicalities of data collection that often constrain the set of data outputs of an experiment. For example, in fluid dynamics, an experimentalist may place a handful of temperature sensors at a certain points,

31 Hartmann, ‘The World as Process’, 85–86.

32 Knorr-Cetina, *Epistemic Cultures*.

typically at the edge, of a tank of fluid. On the other hand, a simulationist can obtain more detail, generating readings for temperature throughout the space. The investigation of detailed dynamics is also a question of exploring the many different possible states the system might go through under different initial conditions, and this is where simulation comes into its own in the third sense. Simulations can be performed where experiments are impractical. This often means cases where experiments can't be done, such as systems that are too large or too energetic to have experimental scale analogues. But it also includes cases where the experiment is to be run many hundreds of times, something that can be automated for simulations, but can be extremely costly for experimentalists.

The other three roles of simulations can also be found at work at AMCG. The theories, hypotheses and models that are developed in the second role can be “fundamental theories”, such as the Navier-Stokes equations for fluid dynamics. But for my informants these are not usually at stake. They are taken for granted, a kind of resource used to explore other kinds of problems. They may well, however, be developing more empirically grounded theories, what often get called “phenomenological” theories, such as theories of the role of gyres in driving deep ocean currents. They may equally be used to develop theories of the apparatus, theories for example about the efficacy of certain algorithms in modelling. Many new techniques that are implemented in simulations are still largely hypothetical until they are realised in the computational apparatus, and it is in simulation projects that their properties can be explored.

In some cases, scientists turn to simulations where their experiments prove impractical. But at AMCG, most scientists are full-time simulationists. They do, however, in some projects, collaborate with experimental scientists. For example, in industry funded projects for designing reactors, turbines, or coolant systems, AMCG provides the simulations which will form the preliminary stages of the engineering project, which the industrial partner will use in order to refine designs before going to the prototyping stage. In other cases, the relationship is reversed, and it was the simulationists who recruited experimentalists in order to supplement their practices.

> “For instance they [two postdocs who ran experiments as part of an oil reservoir conductivity project] would be doing experiments with water of different salinities and they would be measuring how the electrical potential they measure changes as they change the salinity of the brine. That is a key parameter that I need to know when I run my model. And I literally just use the number from the lab experiments.” (QH)

Hartmann's fifth function of simulations is important at AMCG because it is by playing around with simulations that scientists hone their skills. It is often pointed out that scientific models play pedagogical roles. For example, models of molecules are commonly used to teach students of chemistry to think about the relationships between atoms and bonds[33]. But even a complex research-

33 Justi and Gilbert, ‘Models and Modelling in Chemical Education’; see also Chadarevian and Hopwood, *Models: The Third Dimension of Science* for several good case studies of the role of physical models in science.

oriented computational modelling framework has a pedagogical role to play, as it provides a primary medium for thinking and manipulation[34]. Skills in using it can be developed, thus enhancing and extending the kinds of engagements possible. In chapter 8, 'Workability and Habitability', we will look at how software systems may differ or change in their capacity to allow freedom for such manipulations.

It is worth, at this early point, making some comments about terminology. I reserve the term “simulation” for computer simulations. Various commentators have discussed non-computational simulations[35], but for present purposes for “simulation” read “computer simulation”. I use the term to describe concrete computational processes, the actual electronic happenings that occur in space and time, so I use the term for something very specific[36]. This is helpful because it ties our discussion down to the events in the laboratory around which this kind of scientific practice is organised. While simulations are the computational processes, practices of simulating are diverse and complex. The process of creating a simulation involves many different kinds of social processes, abstract reasoning, idealisation. But the simulation itself is a concrete empirical event.

The main difficulty encountered in using the term “simulation” for actual electronic occurrences is that these happen at a scale and speed that is not amenable to direct perception. They are the kind of thing liable to incite debates about realism and anti-realism vis-à-vis unobservable entities. However, while acknowledging that we only have empirical knowledge of what has actually taken place, these electronic events we call “computation” are so thoroughly instrumentalised, and incorporated into so many kinds of scientific practice, that they are strong candidates for the kind of realism Ian Hacking proposed in *Representing and Intervening*[37]. To use a term that will be introduced in chapter 9, computation is not, in computational science, an *epistemic thing*. It is an aspect of reliable apparatuses, not a semi-mysterious thing motivating research. For simulationists, computation is mundane, part of the complex of apparatuses brought to bear on other kinds of problems; the locus of motivation is elsewhere.

Regarding a simulation to be an event in a digital computer requires us to distance ourselves from discussions of simulations in terms of the Greek concept of simulacra[38]. A simulacrum in this sense is a copy of something, and the term “simulation” has been enlisted into a philosophy of images and commodities which prefers to retreat into abstract philosophical territory, with no real concern for how real simulations figure in fields such as science. One recent treatment defines it thus:

34 The term ‘framework’ is used here to describe a technical system, so is used in a very different way to the usage established by Carnap and his followers to talk about theories. Carnap, ‘Empiricism, Semantics, and Ontology’; see also Galison, ‘Computer Simulations and the Trading Zone’.

35 For example Parker, ‘Does Matter Really Matter?’.

36 Similar views are expressed by Humphreys, *Extending Ourselves*, 109; Hartmann, ‘The World as Process’, 83.

37 Hacking, *Representing and Intervening*, 22–25.

38 For example, Baudrillard, *Simulations*.

> “Simulation: a copy without a source, an imitation that has lost its original. The theory of simulation is a theory about how our images, our communications and our media have usurped the role of reality, and a history of how our reality fades”[39].

Firstly note that it makes no sense to begin from a presumption that the etymology of the term will guide us on a productive path of enquiry. Whether simulations might be regarded as copies is an issue that will be dealt with in chapter 5, 'Modelling and Representing'. They are certainly not copies of anything in any straightforward sense, and to the extent that they represent, they represent many different things in many different ways, something that must be understood through their position in a practical nexus of concern. If representation is already multiple, the question of the original is no longer pertinent. And finally, the concept of reality will also be something we should return to, following an analysis of practice, rather than pre-empting it. While my account is a 'cultural studies' account, and thus leans more heavily on the theoretical element of analysis than on the empirical, it is important to temper this orientation by refusing to take voguish theoretical trends as an initial guide, situating my study primarily in the empirical and working from there.

I have no similar desire to narrow down the meaning of the terms “model” and “modelling”. I don't think it is necessary to start with an idea of metaphors or analogies, as has been a strong tradition in philosophy since Mary Hesse's early studies[40]. Modelling, conceptualised as a practical process, is something much broader, and captures the complexity of what it is that scientists are doing when they are building and running simulations, and help to tie these kinds of scientific work to the variety of kinds of physical and ideal models that are also used throughout the sciences[41]. I keep the term modelling broad because it captures something of what we want to talk about when we talk about “practice”, and as we will see in chapter 3, this term is also characterised by considerable, and useful, semantic flexbility.

2.3 The history of computer simulation

The history of simulation is closely associated with the history of the digital computer. The foundations of computing were laid at the interface of engineering and the study of logic. Gottfried Wilhelm Leibniz made an early attempt at a mechanical multiplying machine in 1672, a precursor to Charles Babbage's famous Difference Engine of 1822[42]. Both relied on cogs and gears to produce their mathematical calculations. It was only, however, with further formalisation of logic and algebra by George Boole, that the foundations were laid for figures such as Charles Peirce, Claude Shannon, John Atanasoff and Konrad Zuse to make the connection between Boolean algebra and the workings of electronic circuitry[43]. By the late 1930s, circuits had been constructed that could perform addition,

39 Cubitt, *Simulation and Social Theory*, 1.

40 Hesse, *Models and Analogies*.

41 For a similar view, espoused for a different kind of agenda, see French, ‘Keeping Quiet on the Ontology of Models’.

42 Kaufmann III and Smarr, *Supercomputing and the Transformation of Science*, 26–27.

43 Ibid., 28.

subtraction, multiplication and division. It was, however, the Second World War that provided the greatest stimulus to this endeavour, as newer weaponry required greater and greater numbers of calculations for artillery shell trajectories, and for cracking enemy ciphers[44]. John von Neumann and Stanislaw Ulam's famous work on Monte Carlo methods also emerged from these military concerns[45]. Following the war, the US nuclear programme pushed further the development of computing, and another breakthrough came in the 1950s with the replacement of vacuum tubes with the smaller, more reliable, and more efficient transistors[46].

Computer simulations are among the oldest applications of digital computers. Some of the earliest computers were devised to solve complex mathematical equations in the fields of nuclear physics and climatology. Norman Phillips performed some of the earliest simulations of the earth's climate system in the mid 1950s, on a computer with only 5 kilobytes of memory[47].

It took, however, several decades before computational methods had been adopted across the many disciplines and sub-disciplines of the scientific fields, and before “computational science” was recognised as an approach that potentially spans all the sciences, rather than a niche area of computing[48]. The term “computational science” is often attributed to Nobel Laureate Kenneth Wilson, said to have coined the term in the early-mid 1980s[49].

In the early days, all computers were extremely bulky and expensive. Today's fastest computers are still bulky and expensive. But computing has diversified, and with this diversification comes more ways of integrating computers into research. At AMCG, researchers run simulations on the full range of hardware, all the way up to supercomputers. But much computational science around the world does not use advanced hardware, and is done using the much more modest equipment of a desktop PC. As several commentators have noted, while much talk about “e-Science” or computational science treats it as a homogeneous approach, in fact the ways in which computers are used in different epistemic cultures is highly variable[50].

The growth of computational science is not solely a technical story. It is also social. One of the major factors influencing the pace of uptake of these new methods was a generational divide, with many professors remaining suspicious of simulation until they retired and were replaced by a younger generation who had grown up with it. Sherry Turkle has studied scientists' attitudes towards simulation from the 1980s to the present day, and the generations feature as an important theme in her account. “For physicists [in the 1980s], using simulation when you could be in direct touch with the physical world was close to blasphemy”[51]. “An older generation,” she writes:

44 Ibid., 29–30.

45 Goldsman, Nance, and Wilson, ‘A Brief History of Simulation’.

46 Kaufmann III and Smarr, *Supercomputing and the Transformation of Science*, 34.

47 McGuffie and Henderson-Sellers, *A Climate Modelling Primer*, 63.

48 Hine, ‘Computerization Movements and Scientific Disciplines’.

49 See, for example Denning, ‘Computing Is a Natural Science’, 13.

50 Merz, ‘Embedding Digital Infrastructures’; Fry, ‘Coordination and Control of Research Practice’.

51 Turkle, *Simulation and Its Discontents*, 30.

> "fears that young scientists, engineers, and designers are "drunk with code." A younger generation scrambles to capture their mentors' tacit knowledge of buildings, bodies, and bombs. From both sides of a generational divide, there is anxiety that in simulation, something important slips away"[52].

For my informant NK, the divide maps onto ongoing tensions between computational methods and established theoretical and experimental approaches to science. The most prevalent view among my informants is that these different approaches are not really in competition, that they can never replace each other, and that they have settled into a position of complementarity. But there remains a sense in which simulationists are the usurpers of traditional methods.

> "Some theoreticians are quite negative about computation. They look down on it... Discussions have been had in the past about whether computation could in principle completely do away with experiments. From the experiment side people say there are always bugs, you can't quite trust the computational model. But then from the numerical [i.e. computational] side people say that they have verified it, they know the equations, they know they have solved it, and that it is going to be more accurate than something done in a laboratory. If you want to model a system the idea that that model might be more accurate than a laboratory [experiment] is not often obvious and does not make laboratory people very happy..." (NK)

NK continued with an example:

> "If you said you wanted to study the flow in a cup of 10cm size I can model it to an insane amount of accuracy. But if you want to do it in a lab, there is always a manufacturing tolerance on these things. Some people would argue that the computational model is more accurate. But you could debate that for a hell of a long time. One of the problems is that with the laboratory and the computer things can go wrong in both and things do go wrong in both" (NK)

Today, it is rare to hear claims that simulation will do away entirely with other ways of doing science. The picture is more one of complementarity, and it is common to hear computational science referred to as "the third pillar of science", what Galison calls a "*tertium quid*"[53]. A recent review of computational science in the UK defines it like this:

> "Computational science, the scientific investigation of physical processes through modelling and simulation on computers, has become generally accepted as the third pillar of science, complementing and extending theory and experimentation... Computational Science and Engineering (CSE) today serves to advance all of science and engineering, and many areas of research in the future will be only accessible to those with access to advanced computational technology and platforms."[54]

52 Ibid., 7.

53 Galison, *Image and Logic*, 747.

54 Engineering and Physical Sciences Research Council, 'International Review of Research Using HPC in the UK', 1; see also, for example, Kaufmann III and Smarr, *Supercomputing and the Transformation of Science*, 4; Rohrlitch, 'Computer Simulation in the Physical Sciences'; Ostrom, 'Computer Simulation'; Godfrey-Smith, 'The Strategy of Model-Based Science'.

But despite its widespread use across the sciences, simulation has often been regarded as a neglected area of study in science studies[55]. This is at least partly because computational science is more difficult to ethnographically observe than laboratory sciences, which formed the bastion of early sociological accounts. Similarly, it was only in the late Twentieth Century that instrumentation and experimentation started to be given adequate space in philosophical accounts of science[56]. The philosophy of simulation, as another practical method for carrying out research, has been bolstered by this trend, yielding many philosophical treatments attempting to philosophically compare simulations with experiments[57]. As Sismondo put it: “In their focus on examining the warrant for fundamental theories, philosophers of science have almost completely neglected the processes involved in applying such theories”[58]. Because of this new trend looking at processes of applying theories, “modelling” has become a concept of great philosophical interest, and simulation as one prevalent way of doing modelling has consequently received a great deal of attention too[59].

2.4 Types of simulation

To gain a handle on the breadth of types of simulation, it is useful to think about some basic distinctions. Simulations of discrete problems versus continuous problems form one axis. Uses of simulation frameworks that also extend those frameworks versus those that just put pre-made software to use is another. Thirdly, the kind of scientific practice in which the simulation is used depends on whether the software in question is a big project, involving many people and a lot of code, or whether it is a smaller enterprise.

All simulation is discrete, in the sense that all digital computational processes are described in discrete mathematics. A simulation, inasmuch as it can be thought of as a series of logical operations, consists of a finite set of such operations. As CE put it, “there is no continuum on a computer chip”. An initial sub-classification of simulations would divide them into those that simulate processes that are themselves described by discrete mathematics, and those that are described by continuous mathematics involving infinities/infinitesimals. The latter are what concern me in this thesis[60]. They include the vast array of simulations of problems that are theoretically described by differential equations, which have to be “discretised” to run on a computer. But it is worth pointing out at the start that there are also problems that more “naturally” fall into a digital computer because of their discrete nature, and these include, for example, agent-based modelling and network analysis.

The simulations that I studied are under active development, being used for research by the same people who write their code. Developers and users are for the most part the same people. With

55 Merz, ‘Multiplex and Unfolding’, 294.

56 Radder, ‘Toward a More Developed Philosophy of Scientific Experimentation’.

57 See, for example Morgan, ‘Experiments Without Material Intervention’; Guala, ‘Models, Simulations, and Experiments’.

58 Sismondo, ‘Models, Simulations, and Their Objects’, 256.

59 Morgan and Morrison, *Models as Mediators*; Magnani and Nersessian, *Model-Based Reasoning*.

60 But see Rohrlitch, ‘Computer Simulation in the Physical Sciences’, 515–516 for an interesting discussion of potential crossover between the domains.

Fluidity, however, the biggest model developed at AMCG, the group is just beginning to see uptake by a new category of scientist, which are often referred to as “pure users”. These people use the software but don't develop it. They need to understand what it does and how it does it, but they are sheltered from the intricacies of the source code, and from the “bleeding edge” of its development. They work largely through other programmes which provide interfaces through which the code is set up and its outputs analysed. But at the moment pure users are a minority group. They have generally appeared as a result of the Fluidity software becoming open source, freely available over the internet. This also means that the group has little knowledge of who they are and what they are doing with the code, except in those cases (at a rate of about one or two a week) in which an external user uses the mailing lists to make a technical query. But while Fluidity has very little by way of a pure user base, other software packages are very well established in this respect, having many pure users. In the geophysical fluid dynamics field, for example, the American model the MIT GCM (Massachusetts Institute of Technology General Circulation Model) is widely used in this way, as is the French model NEMO (Nucleus for European Modelling of the Ocean).

Connected with the division between developer-based science and user-based science is a distinction between small and big software. This division is hard to make exactly because there are no absolute measures of size for software (see chapter 8, 'Workability and Habitability'). Much developer-based science is carried out with small software applications written by the research team for a particular project, and this kind of “cottage industry” software production is very different to the bigger scale endeavours needed to develop a large and complex software system on the scale of Fluidity, not least because the latter requires a much bigger team.

2.5 Computational laboratories

Here I provide a brief outline of the core equipment that furnishes the computational science laboratory. This will give an initial picture of the resources that are essential to this kind of work. Accounts of scientists and external commentators can easily place such stress on the simulation software that it can seem like this somehow exists independently and works on its own. It is too easy to think of a computer as just one thing, and equally easy to lump together very different software systems. Part of the goal of this thesis is to produce a much more nuanced account. A complex web of apparatuses exists in and between computers, deployed in many different ways at different stages of the research process. The following table describes the major software equipment that is used with the Fluidity code.

Software	
Modelling framework – e.g. Fluidity	I call the core code a “modelling framework” because it can be used to create many different simulations, and many kinds of simulations. A modelling framework like Fluidity is not a model in any straightforward sense. It has

Software	
	many different options that can set it up to create simulations that model physical systems, ideal systems or mathematical systems, but it is always bigger than any particular simulation. It has the potential to model such a vast number of very different systems that calling it a model in the singular does not to do it justice.
Code editor – e.g. gedit	Working on the source code requires an interface. Programmers have their favourite editors, which are essentially text editors with additional capacities designed to help work with code, highlighting syntax, for example, or comparing different versions of the same file.
Set-up interface – e.g. Spud/Diamond	Setting up a modelling framework like Fluidity requires a lot of options, parameters and boundary conditions. Spud is the GUI (Graphical User Interface) for Diamond, the options set-up system. Together they allow a user to easily edit and validate an FLML file (FLuidity Markup Language), a format based on the widely used XML format, which is fed to the Fluidity programme in order to run a simulation. A “schema” file contains the information about what format and structure that XML needs to have in in order to be a correct set of options for the particular modelling framework that is being used (what settings are dependent on other settings being set in certain ways, which settings are mutually exclusive, what is necessary and what optional). It also includes meta-data about settings to inform a user what they are used for, problems to watch out for, and so on. Spud takes this information and creates a user interface that displays all the options as a tree, allowing them to be changed via text fields, drop down menus, highlighting settings that must be given values, and so on.
Mesh generator – e.g. GMesh	One of the main things that is needed to run a simulation is a mesh file. This describes the shape of the domain, as well as the way in which the domain is to be divided up into a grid. Software like GMesh allows you to generate and manipulate these meshes by specifying rules and by manipulating them through a visual interface.
Compiler – e.g. GFortran	In addition to the set up files, for Fluidity to be run, it is necessary to obtain a compiled version of the code. This is an executable binary file that has been generated from the source code. Compilers are highly complex pieces of software that, via a series of stages, turn source code into something that can be executed on a computer. They can be set up in many different ways, allowing for different forms of optimisation. You have different compilers for different kinds of target platform. For example Fluidity is compiled with Gfortran for running on Linux workstations, but is compiled with the Intel Fortran compiler for running on the Imperial College cluster.

Software	
Adjoint software – e.g. Libadjoint	The adjoint of a differential equation is an alternative formulation that rearranges its terms in a way that is useful for certain kinds of problems, including certain kinds of optimisation problems. Once those equations have been discretised for the computer, this becomes a very complex operation. An adjoint system writes key data to disk while a simulation is processed, and computes the adjoint afterwards.
Parallelisation library – e.g. Zoltan	The modelling framework will draw on a variety of libraries. These are external software systems with which it interfaces in order to use a variety of techniques. For parallelisation, Fluidity uses algorithms from the Zoltan library.
Solver library – e.g. PETSc	A solver library contains algorithms used to solve the matrices that are generated as part of the simulation. Which solvers are used will depend on how the model is set up and the properties of the matrices in question.
Debugging tools – e.g. gdb	When a simulation goes wrong, which can mean it causes a crash or that it produces erroneous results, scientists usually re-run the program, but this often requires re-compiling it to make its operation easier to follow. De-bugging tools help them to follow the computational process and identify where it is getting into trouble, ideally thereby indicating what part of the set-up or source code is at fault.
Post-processing tools – e.g. Paraview	Once the simulation has been run and an output generated, it is necessary to look at that output. Only in a small minority of cases is the output data meaningful when read directly from the file. Post-processing techniques such as visualisation suites convert that data into a visual representation through a user interface which presents the user with an array of settings and filters. Diagnostic tools similarly convert the data into a more usable form by calculating averages, comparisons with other data-sets, and so on. But diagnostics are often “hand-coded”, little sections of code written by the scientist in an interpretive code like Python, rather than relying on an externally developed product.
Operating system – e.g. Ubuntu	All simulations are run on computers, which require operating systems to manage their hardware. Fluidity, for example, is supported on current versions of Ubuntu Linux, but it has also been run successfully on Debian, Red Hat and Suse. When run on HECToR, the UK National Supercomputer, it runs on CLE (Cray Linux Environment). Each of these has their own characteristics which can affect the simulation, and each is regularly updated, producing a stream of

Software	
	new versions which developers and users need to keep up to date with.
Source code repository – e.g. Bazaar	The source code for a modelling framework is not simply saved on a hard disk. It is managed by a software system that records all changes made to it, so that particular simulations and bugs that occur in them can be referred to a particular version of that code and so that any changes can be dated and traced to the developer who made them.
Automated testing suite – e.g. Buildbot	Other software systems couple with the repository in order to facilitate development. One of these is automated testing, which runs a set of tests on new versions of the code, in order to identify problems with it. These tests can test the code on different hardware set-ups, with different compilers, or can test that the functionality within the code is still operational.
Other software – e.g. bug tracking software, chat clients, etc.	Further software facilitates the smooth working of the group. One example is bug tracking, which is essentially a database of problems, with associated metadata (description, symptoms, which person is responsible, conversations about the bug, etc.) and a user interface. Code review systems put new sections of code up for review by other scientists. Chat channels and email lists facilitate communication. Online wiki software organises a website with "how to" guides for the software.

2.6 Sub-models and coupled models

There is diversity within and among modelling frameworks. The core algorithms of a modelling framework like Fluidity give approximate solutions to partial differential equations. This core, however, is supplemented by a great many other components. Software libraries have already been discussed above. Parameterisations can be regarded as sub-models which provide approximate solutions to parts of the system in question that for one reason or another are not resolved by the core algorithms. The system needs to be tweaked to compensate for this basic deficit. This is a fact of life for most kinds of simulations, but it has also been regarded as a sign of weakness, an admission of failure.

> "It is (unfortunately) necessary to represent a distinct part, or more usually many distinct parts, of the complete system by imprecise or semi-empirical mathematical expressions. Worse still is the need to neglect completely many parts of the complete and highly complex system. This process of neglect/semi-empirical or imprecise representation is termed parameterization"[61].

But for most simulationists, parametrisations are inevitable, and such charged language is not necessary if they are done well. They are add more approximations on top of the approximations

61 McGuffie and Henderson-Sellers, *A Climate Modelling Primer*, 72.

already generated by the core of the code. They are not necessarily troublesome, but clearly need close attention to make sure that they are behaving properly when integrated into a given code.

> "Because you discretise things on a grid there are certain things that you can't represent... In that jump from the continuous system to the code some things are lost, such as turbulence. If it is important, you need to have sub-models [i.e. parameterisations] that represent stuff that isn't being explicitly solved for" (NK)

Turbulence parameterisations are a good example, because they are needed in a lot of fluids models. They are like little models within the main code which compensate for the effects of the discretised simulation being "grainy" at some level, that there are processes going on at smaller levels which cannot be explicitly resolved, but whose effects on higher level mixing needs to be factored in.

On a larger scale, bringing different models together is a common tactic for simulating phenomena that involve different kinds of processes. Climate models are examples where this has been pushed to an extreme, with oceans, ice, atmosphere, ecological models, etc., all coupled together. AMCG has long been at the forefront of fluids/radiation coupling. The geophysical applications of Fluidity are also being extended in collaboration with the Scott Polar Research Institute in Cambridge, coupling the ocean model with their ice shelf model. Processes of modelling can thus involve the combination of several models in interaction and nested within each other.

2.7 History of the Applied Modelling and Computation Group

Having sketched an outline of computational science, we turn now to introduce in detail the group that has been the focus of my research. The Applied Modelling Group at Imperial College has been around since the 1960s. Its early story is best told by GT, who was the director of the group for many years, before retiring from the job in the early 2000s to become a part-time senior investigator. At this point CK took over as director.

> "The group began as part of a nuclear power section in the Mechanical Engineering department back in the 1960s. But the critical point was [in the late 1980s] when two people came together who had general applications interests in two complementary areas, both using the same underlying mathematical tools, and that was the particular felicitous thing that happened. These were CK and he was preceded by a guy called CW. They were both enthusiasts for this finite element method, which was really a structural modelling tool developed from fairly empirical roots by civil engineers, for designing bridges and things like that, with an emphasis on models which conform to the structure, and putting that first and foremost in the approach. It turned out that here we had CW with an interest in applying this to nuclear radiation... and CK who was a bit of a pioneer in applying it to fluids modelling. At the time it wasn't really accepted generally for modelling fluids; other techniques called finite difference models were much more generally employed." (GT)

The group was still very small at this point: around a dozen researchers. One of the most significant events for its future path was its movement between different departments during the 1990s. The group's journey was driven by wider trends favouring different disciplinary areas, from a significant decrease in funding for nuclear power research, to the changing fortunes of environmental science.

> “The group started in explicit nuclear power... In the early 90s it was clear that nuclear was quite out of favour [with funding bodies]. So with the help of the then deputy rector we engineered a move of the group from Mechanical Engineering to the Centre for Environmental Technology as it was called at the time. This was in the early '90s. Then that led us to be incorporated into a new environmental department called the T. H. Huxley School of the Environment...” (GT)
>
> “Then things happened and that collapsed completely... and we decided to attach ourselves to [Earth Science and Engineering] so we could continue our diversification and bring application of nuclear modelling tools to environmental subjects... We managed fortunately to attach ourselves to it because we helped to get a 5* RAE rating for the department almost straight away. And the pure environmental department [the T. H. Huxley School] was actually dismantled by the college and the residue is the Centre for Environmental Policy which still exists: the residue of a difficult time when the pure environmental work had to be downsized from a departmental scale to a centre scale.” (GT)
>
> “So what started out as a nuclear power section in mechanical engineering and decided it would call itself AMCG, and followed this “wandering tribe” path from mech. eng. to the Centre for Environmental Technology to the T. H. Huxley School to Earth Science and Engineering; that was quite fortunate for AMCG. We ended up with the best of all worlds in a department with the explicitly earth science part of the college's environment interests.” (GT)

While all this structural change was shifting the disciplinary coordinates of AMCG's research, a major research project was being formulated that was eventually lead to a massive expansion in the population of the group towards its current situation, with 58 full time staff, and 36 affiliates working from other departments at Imperial College or at other universities.

> “Towards the end of our stay in mech. eng., CW and CK and I met the only oceanographer in Imperial College, at least the only numerical oceanographer, who worked in the mathematics department. He is now dead: an enormous tragedy. And NK was to be his chosen protégé. He was diabetic and suffered a diabetic collapse at a conference in France. He was very young. His legacy was a wonderful joint proposal with CK, CY and myself aiming to support NK as a postdoc... and this was awarded just before he died but before it actually started... We all met around about 1990 in the quadrangle out there and talked about making a finite element model of the world. That illustrates really the role of the chance in all this, the element of chance in including people of complementary abilities... We had a bit of trouble with the research council because we had lost the principle investigator but we managed to convince them that if I was the nominal PI we would get an eminent advisory committee together, of people from Southampton, Reading and Liverpool to oversee the project.” (GT)

From this initial foundation, the group gradually increased in size as more grants were won in order to further develop the ocean model.

> "It all has been built up from scratch, to a large extent due to CK's ferocious dedication, at that time working 24/7 really to create the ocean model, which we then called ICOM [Imperial College Ocean Model] and which has been patiently ramped up through various NERC [Natural Environment Research Council] grants and getting people like NK and TT and others involved, to consolidate the whole thing and get it well established." (GT)

Chapter 8, 'Workability and Habitability' will return to this story at the end of the thesis, showing how this big project unfolded and the new pressures it put on the group. Note that "ICOM" is another name for Fluidity, which is also sometimes called Fluidity-ICOM. The reason for this plurality is simply that it is unconventional for a code to be used for both small scale and large scale fluids processes. The ICOM brand was created in order to better promote the large scale work that the group was doing.

The major codes developed at AMCG are:

Fluidity/ICOM/Fluidity-ICOM: Computational Fluid Dynamics (CFD)/Geophysical Fluid Dynamics (GFD)

EVENT: Radiation Transport (an older formulation of the transport equations)

RADIANT: Radiation Transport (a newer formulation of the transport equations)

FETCH: Coupling code linking EVENT with an older version of Fluidity to solve coupled fluids/radiation problems

FETCH 2: New coupling code linking RADIANT with the current version of Fluidity to solve coupled fluids/radiation problems

iSALE: Multi-material impact modelling code

2.8 Interdisciplinarity

Some writers have claimed that rather than treat this kind of science as interdisciplinary in the conventional sense, a more radical term, "transdisciplinarity" is required[62]. At AMCG, research is transdisciplinary where not only the techniques, but also the problems themselves, exist outside conventional disciplinary frameworks. But it is important not to get carried away. While transdisciplinarity reveals one face of research, a great deal of computational science remains interdisciplinary, putting computers to work in order to study problems that can be traced to conventional disciplines: oceanography, engineering, or physics, for example. But even if these problems have a history that stretches back beyond the introduction of computational techniques, the way that computational scientists tackle them has a distinctive character which is stamped upon the future evolution of these problems as objects of scientific concern.

62 Gibbons et al., *The New Production of Knowledge*.

> "I did hear him [NA – a professor from a traditional geology background] say once that it was a kind of revelation to him how people [in computational science] thought... people like AF, people like CK, TT: the way they approach problems, working in a synthesis including the mathematical, numerical and application sides." (GT)

What GT highlights is the mixture of disciplines in the work of computational scientists, something that is largely social rather than just being an attribute of single individuals; it gives the group its distinctive style of research. People from very different backgrounds bring their different competencies to bear on problems studied through simulations. This diversity exists not just because the group recruits from a diverse range of disciplinary areas, but because the work is necessarily interdisciplinary. It is necessary to bring together mathematicians and computer scientists with applications specialists such as oceanographers and geologists. While each consequently learns a lot from the other, there is no expectation that the difference will be overcome. No computational science generalists exist. Everyone is specialised in complementary ways. But it can be hard to learn to interact with people from such different backgrounds. KT, an oceanographer, explains his entry into the group:

> "I got involved in this because I was interested in a problem and I thought that the model could be used to solve this problem. So I started talking to CK and NK and through a series of conversations we eventually learned to speak each others' language. I think it is like an artillery range: you are firing shells and it was about three of four meetings before we finally started having good conversations. We discussed a project, which got funded, and while we were doing that we had an MSci student who did the proof of concept and we got that published. CK asked me to get involved with the consortium proposal and eventually I got heavily involved in the group, because I can find out an awful lot about the oceans from this computational approach." (KT)

It was interesting to see that almost all my informants tended to affiliate themselves with their original discipline from their undergraduate years, even if they had spent many years since then working in this field. "Computational scientist" is not a label as easily adopted as "I am a geologist who works in computational science" or "I am a mathematician who works in computational science". The reason for this is probably simply that there are many different kinds of scientists working together in computational science, so saying you are a computational scientist doesn't clarify what kind of work you actually do nearly so much as referring to your background in your original discipline.

AMCG involves a mixture of mathematicians, engineers, computer scientists, physicists, geologists and oceanographers. By far the biggest group are the mathematicians, who at any one time account for between a third and a half of the group. Computer science and physics are less well represented than one might expect, with more applications specialists coming from the applied physical sciences (geology, oceanography, engineering), and most developers having a mathematics background. Fewer than half of new arrivals have programming experience before joining the group, and those that have worked with software have generally worked with much smaller scale systems than Fluidity or RADIANT. Once scientists have been there a while, they tend to line up on one side of the principle divide, between those who consider themselves programmers and those who don't.

> "I think of myself as a programmer and a scientist, and a mathematician. You can't separate them. They are all interrelated. I am a very specialised, very educated kind of programmer. But if you ask UU or CK are they programmers, they will say "no": they don't know how to program. They happen to program in the course of their duties but they don't do it "right" and that is a big divide" (KU)
>
> "I try and attend all the development meetings, but I also try and keep up to date with the earth and planets side of things. That is really where my research lies but I know I can't really do that without the tools, so I am kind of stuck in between two worlds" (IW)

The interface of different disciplines is an extremely productive location, something that is recognised by philosophers who are sceptical that science is a meaningfully unified enterprise. Both Galison and Hacking stress that disunity is not a problem for science. It does not undermine its efficacy. Quite on the contrary, "it is precisely the disunification of science that brings strength and stability"[63]. "It is precisely the disunity of science that allows us to observe (deploying one massive batch of theoretical assumptions) another aspect of nature (about which we have an unconnected bunch of ideas)"[64]. This is also something Rheinberger has stressed: "To bring alternative spaces of representation into existence is what scientific activity is about, and this is why the question of reality as an attribute of alternative representations, and the question of representation as an attribute of its alternative uses, will continue to stay at the centre stage of the scientific enterprise"[65].

The articulation of relatively independent bodies of knowledge, practice and skill can be very productive. A research group such as AMCG thrives on such a difference. It draws together scientists of radically different persuasions, who have very different backgrounds and who publish in entirely different fields. But in the actual time of research, prior to their publications, they work very closely together. The articulation of different fields is not limited to linguistic exchange (as in Galison's trading zones[66]), but is also achieved through the techniques and things they have in common, what Star and Griesemer named "boundary objects"[67]. The objects of study, but also the models and modelling technologies, are hinges between the different orientations of the community.

2.9 The finite element method (FEM)

AMCG is distinctive for its specialism in the finite element method (FEM). This is a method for the approximate numerical solution of partial differential equations. It is the method used by both Fluidity and RADIANT. The spatial domain is divided into a mesh, usually of triangles (if two-

63 Galison, *Image and Logic*, 781.

64 Hacking, *Representing and Intervening*, 183.

65 Rheinberger, *Epistemic Things*, 113.

66 Galison, 'Computer Simulations and the Trading Zone'.

67 Star and Griesemer, 'Institutional Ecology'.

dimensional) or tetrahedra (if three-dimensional). The solution to the equations is represented by a series of “basis functions” on this mesh. Although the mesh divides the domain into discrete units, these basis functions provide a variety of ways to represent (for each variable) what is going on between the vertices. It may be represented by linear, quadratic, or higher order polynomials. Which is chosen will depend on the application. The solution is commonly represented by “continuous” basis functions, which means that neighbouring functions have the same value at the vertex they have in common. But for some cases, particularly those in which large changes in a variable occur below the scale of the mesh, it is preferable to use “discontinuous” basis functions, in which the solution is not forced to “join up” when approached from different sides of a vertex.

The difference between numerical methods is something that can be studied on a fundamental mathematical level. All I can deal with here is a more heuristic level of understanding. The main rival to the finite element method for geophysical problems is the finite difference method, which is well known because it is currently used in the majority of climate models. A rough and ready distinction between the two can be expressed in terms of the approximations they make. Finite difference decomposes the partial differential equations into a set of difference equations, thus primarily working from an approximation of the equations to be solved, and working through to a solution. Finite element, in contrast, starts by approximating the solution to the equations, setting up a “solution space” by positing a mesh of basis functions with a given number of degrees of freedom, and algorithmically searching within this space for the best approximation.

On the other hand, it is common to hear finite element enthusiasts point out that, mathematically speaking, finite difference is equivalent to a sub-set of finite element discretisations. Any finite difference model could theoretically be rewritten as a finite element one, while the inverse translation is not always possible. Mathematical differences and relations aside, the difference between the two methods is usually immediately obvious because finite element models tend to utilise triangular or tetrahedral meshes, whereas finite difference requires square or quadrahedral (or higher).

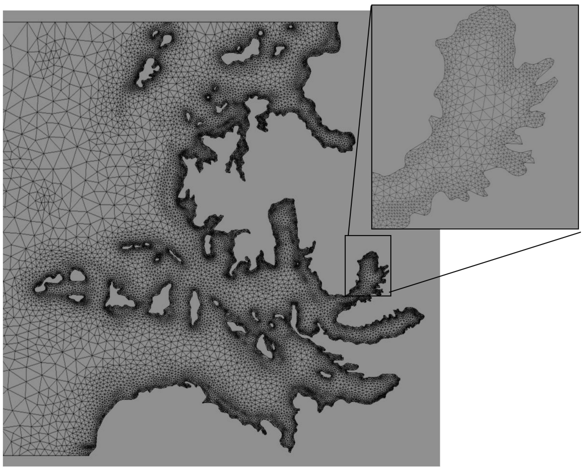


Figure 1: Example of unstructured mesh for complex topography, shown at Fluidity training event 2011

The use of tetrahedral meshes is particularly good for representing complex topography, where the mesh is unstructured. Triangles/tetrahedra can be arranged in all sorts of complex patterns to represent the domain, as well as adapting to represent the complexities of the solution within that domain. “The underlying principles of Fluidity is that we use an unstructured mesh that can represent things like coastlines and bathymetry better than squares – which is intuitively obvious – and that gives you a better answer” (QS).

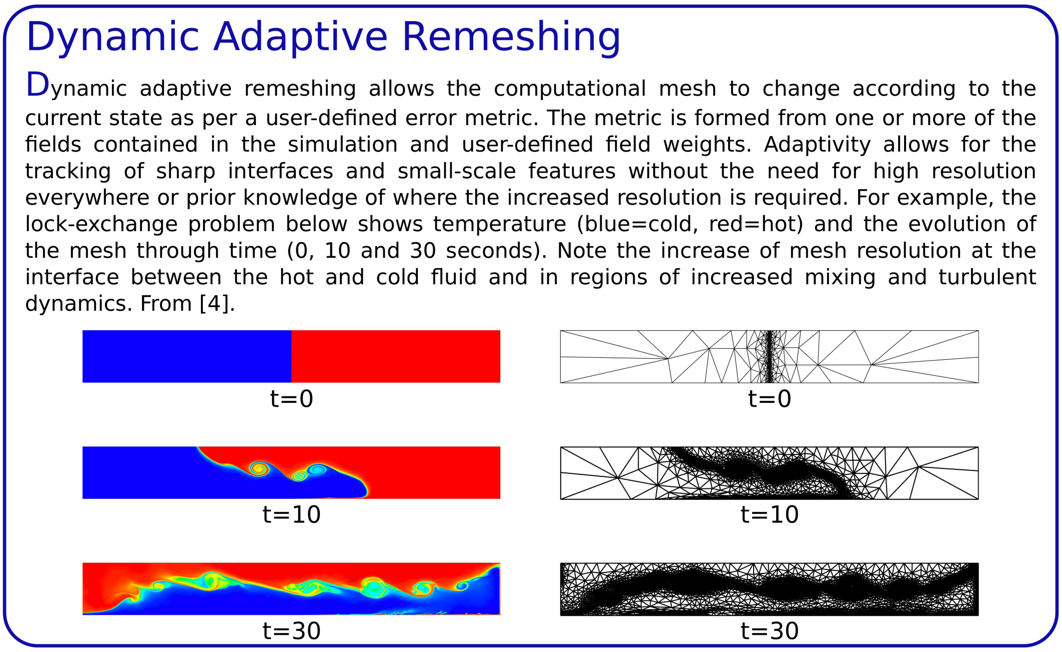


Figure 2: Excerpt from poster about the functionality of the Fluidity code

One of the other advantages of the finite element method is that it is amenable to being used with adaptive meshes. These meshes evolve as the simulation runs, using predefined rules to concentrate resolution where it is needed. This means that areas in which complex phenomena are being resolved get more elements and thus more computer resources, whereas less interesting areas of the simulation can be represented by coarse, large elements. The ability of adaptivity to make simulations efficient goes some way toward compensating for the complexity of FEM. “Fluidity,” says QS, “has the advantage of speed because of adaptivity but the downside is that because we are using finite element that is inherently slower than finite difference that most other models use. So we are already a factor of ten-ish slower for the same resolution”. Part of the overall mission of the group is to demonstrate that this trade-off is worthwhile.

The combination of unstructured and adaptive meshes allows resolution of small-scale phenomena within a domain, such as gyres within an ocean, which are very important oceanographic phenomena, areas of powerful upwards and downwards mixing which are extremely small compared with the ocean as a whole. It is hard to say where these phenomena will emerge so it is extremely helpful to have a computational system that can automatically concentrate resolution when and where such phenomena appear.

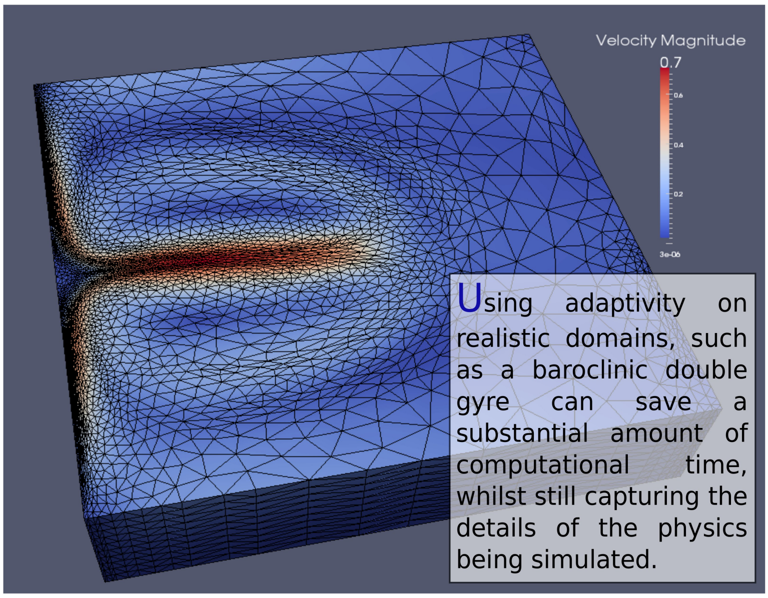


Figure 3: Excerpt from poster displaying functionality of Fluidity

Many of the same kinds of advances that have made Fluidity a significant code have also been worked into the radiation transport codes developed at AMCG, which are also implementations of the finite element method. RADIANT also has adaptive unstructured meshes, but it is a much younger code developed by a smaller group. The older code EVENT, which RADIANT is to replace, is no longer under active development and as such is no longer a focus point for these researchers' enthusiasms for the future.

2.10 A multiplicity of simulations

I now give a preliminary account of how simulations get made and run. We will return to this question throughout the thesis, but it is important to give an initial sketch so that the relationship between modelling frameworks and simulations is made very clear.

One of the first things to note is that a simulation on its own is wholly out of place. NK, introducing Fluidity to new users at the 2011 training event, said: “Remember that a single simulation on its own is almost certainly useless”. Simulation is always multiple in scientific practice. There are several axes along which simulations multiply.

Firstly, a set of simulations is usually run under different initial conditions to explore the behaviour of the system that is modelled in different circumstances, to build up a picture of its behaviour in terms of a set of possible scenarios. For example, in industrial coolant pipe engineering, the temperature of the fluid might be changed, the velocity of the fluid, and the amount of drag on the pipes' walls.

A second set of simulations is generated by a variation in the properties of the model that do not correspond to any empirical reality of the target system. There is no analogy in these cases to the variation in initial conditions of an experimental set-up. For example, the error metric used in order to specify the rules for adaptivity might be modified, or the rate at which adaptivity is to be applied, or a maximum or minimum limit imposed on how fine or coarse the mesh can get. The exploration of these sets of simulations tells the scientist something about the behaviour of parts of the modelling framework, about how sensitive its results are to the specifics of its set-up. The most common set of simulations in this respect is generated by varying the number of elements in the simulation, to see how coarse it can get, and thus how fast the computer can process the simulation, while maintaining the desired degree of accuracy.

Further sets of simulations are generated by the process of building the software and generating the model set-up. Scientists rarely dive straight in to complex problems, but seek a route through simpler, often idealised or experimental-scale problems in order to find the settings that they are confident with. Where new bits of software are being developed in order to create the new simulation, this process is even more important, in order to test out the new components in a range of well-studied cases before applying them to new problems. This trajectory will be fleshed out in more detail with a case study in chapter 4, 'Reason in Practice'.

On a small scale, multiple simulations are generated by the kind of “tinkering” that software developers routinely engage in as they try things out, gain confidence with them, and move on to put their software to work in ever new ways. While we need not subscribe to her views of simulations as mathematical models, Dowling eloquently writes about this “hands on” aspect of working with simulations. “A sense of direct manipulation encourages simulators to develop a "feel" for their mathematical models with their hands and their eyes, by tinkering with them, noticing how they behave, and developing a practical intuition for how they work”[68]. The key thing here is the forwards path forged by processes of generating and manipulating simulations. In Sismondo's view, “[m]odels become a form of glue, simultaneously epistemic and social, that allows inquiry to go forward...”[69].

But more than simply being a feature of individual research projects, the whole enterprise of building a modelling framework, and doing research with it, is one of many steps paving the way for further studies. Not only do individual research projects take an iterative approach to developing software and exploring possibility spaces. So too does the research of the group as a whole, insofar as it is bound to the iterative development of software, and the iterative validation of that software's functionality.

> “Problems are like stepping stones or rungs in a ladder. You really can't sit with a group of people for ten years and work out an ocean model and then apply it. You need to apply it all the way along with different problems” (HP)

Funding tends to be tied to smaller projects, so these piece by piece add up to a larger trajectory of model development. “There is [in AMCG] a model developmental strategy in that there are some things that can't be done until other things are done, so there are some things that have to be done in order. And then there are other things that can just be done whenever the money is available” (KA). While some extensions can be added whenever there is funding, the more complex applications can only be funded if many necessary components of their realisation are already in place, studied and tested, serving as proofs of concept for future extensions.

In 2010 the group was commissioned by an oil company to build a simulation of a particular oil reservoir they were interested in. They had some flexibility in how they went about it. The oil reservoir simulation would require “multiphase” capabilities, the interaction of different fluids; for example, the interaction of the oil with gasses, and of the interaction of fractions of oil with different densities. This multiphase functionality had been designed and planned. It had actually been part of the older version of Fluidity, but had been dropped during the early 2000s when the group instigated a major rewrite (see chapter 8, 'Workability and Habitability'). When this project was won, senior staff met to discuss the idea of using it as an opportunity to finally integrate multiphase capabilities back into the main Fluidity code. Not only would this functionality then be usable by a much wider range of users, it would also be integrated with the cutting edge element types, solvers and parallelisation routines.

68 Dowling, ‘Experimenting on Theories’, 269.

69 Sismondo, ‘Models, Simulations, and Their Objects’, 258.

> “We had been talking about this for ages. It is something CK is obsessed with, for years even before I arrived... So we talked about this for a long time, and several parts of the group had been working on different bits of the grand scenario for doing the project: the new DG [discontinuous Galerkin] formulation WS works on, the new element types he and CE work on, the new discretisation formulations that I have been working on for some time... Then when we had this green light, money for one project that we weren't expecting. We all sat down and planned out how it would go.” (QT)

Because research is funded by these small to medium sized projects, the larger vision for the group and for the technologies they create has to be created and maintained out of smaller building blocks. The multiphase capability could have been tacked on just for this project, but in using the space of this research to achieve full integration, the way is paved for future projects that will rely on it to expand the framework in still further directions.

A final aspect of the iterativity of simulating is pedagogical. It is these processes of multiplying simulations along a temporal axis that gives simulationists their practical knowledge of their software and of the systems that its simulations represent. Building simulations from smaller component parts is regarded as an epistemological good in itself.

> “If we included everything we would learn nothing. You have got to build it up or else you really don't know what is controlling what: what the model is doing. What you really don't want to do is chuck a whole load of parameters in, get a result, and say “look this is the result” **-** you want to be able to say “this is the result *because*...”” (IW)

From the point of view of science in practice, we must take IW's remark not just as recognition of the role of multiple simulations in justifying claims that can be made about them, but also as an expression of the way in which repetition and multiplication establish the kinds of sensitivities through which scientists learn to work, through which they learn their trade and through which they end up making any such linguistic claim in the first place (we return to these ideas in chapter 6).

2.11 Visualisation and diagnostics

Visualisations, graphs and diagnostic variables are the key modes in which scientists are able to encounter the data they generate in a form which is amenable to manipulation. They are the ways in which scientists encounter their simulation, the way in which they are confronted with their own practices. Chapter 6, 'Images in/of Simulation' will address these themes in much greater depth. This section merely points to the absence of direct encounter with an original “output” of a simulation, in order to press home the point that no simulation is encountered outside of wider systems of practice.

> “In doing the work I have to take the big data file and reduce it to something that means something. You can't just look at a picture of a flow field and know whether what you have got is what you want. If it is a pipe there are obvious things [to look at] like the velocity profile across the pipe. Or with turbulent flows you can do various statistical measures to find out if you have a statistically steady flow and what fluctuation there is about that. Then

> there is my favourite test which I have been working on for about a year now – the backward facing step – the flow leaving the edge of something and reattaching to the surface downstream. So you measure the distance from the step to the reattachment and that is a well measured, well reported number in the literature and you can say if it is longer or shorter you can say what that means" (QY)

Simulations do not have a simple and unambiguous output. Computational processes do not write to disk every step they take. There is no record of the computation outside of the results of specific instructions to write data to disk at certain points in the process. These are already determined by the scientist according to what kind of reduction of that data (visual, numerical) is going to be appropriate. You might need different data outputs depending on whether you are wanting to generate a rich graphic of the whole domain, or a statistical measure of a certain part of it. Which of these is appropriate depends on the application, with many outputs in published literature being driven by precedent: you create a graph analogous to one in a previous paper to facilitate easy comparison of your results. Graphs in general are considered a reliable way of seeing key mathematical relationships such as convergence.

> "If I say I have a graph for error on this axis and number of elements on this side then you think it should go in that direction, and if it does, you are fine, and if it doesn't then you know its wrong. So a simple graph is often much more intuitive to understand than some much more complex beast. Like when someone has tried to cram everything onto one graph and you just can't interpret it; it is just impossible" (QS)

It is important to give proper emphasis to these post-processing stages. There is no raw form of encounter between scientist and simulation. The closest they tend to come is during some debugging processes and in some analytical projects when they write to disk the maximum data from every step of the simulation: every variable, at every time step, for every point on the mesh. But this is rare because it tends to be enormously expensive in terms of disk space, limits processing speed by constant writing to disk, and because for most purposes such a vast data-set is too unwieldy to be useful. Most simulations will never be encountered like this. They will be encountered through translations of data into other forms, which themselves will have been written into the model set-up from the start.

2.12 Verification and validation

Verification and validation is one of the core topics in the theory of numerical simulation. In essence, they are the operations that are supposed to demonstrate that the simulation is a good representation of the mathematical equations (verification) and the target system (validation). They are often treated together, but it is worth pointing out that there are differences of meaning in how the terms are applied in practice. There are three distinctive kinds of verification, and two ways of treating validation. I go through each in turn.

The first sense of verification is a mathematical activity which is carried out on paper, with no computers involved. At AMCG there is a handful of mathematicians whose research revolves around the development of numerical methods and the study of their properties. For the finite element method, this is often a matter of developing new element types to solve partial differential equations, which themselves are often reformulated or simplified for this purpose. It is possible to prove various results solely “on paper”, about the relationship between the continuous mathematical system and the discretisation you have come up with. We will see one such paper proof in chapter 9. These proofs are always partial with respect to any actual simulation because there is a lot more input required to implement that discretisation in code, which might also influence the goodness of the solution. There is no straightforward equivalence between what we could call the “mathematical model” and the computational system used by simulationists[70]. But by addressing the question of the basic methods at the heart of the software this kind of verification exercise represents a first step towards assuring that the simulation itself is doing what it is supposed to be doing.

The second sense of verification is the one more familiar to computer scientists, but which actually gets very little mention among computational scientists. Here verification is the question of whether the software that has been written really encodes what it is supposed to encode (or whether there are errors in its implementation). In this case verification is a question of the relationship between some specification for the program, and source code, and it is a general issue for software development, rather than anything specific to scientific software or simulation. My informants rarely concerned themselves with this side of verification because they do not work to specification. They simply do not have one side of the relation. While the broad outlines of projects are agreed in advance, the vast majority of the details of implementation are worked out during the process of coding, and are never separated from that process such that a comprehensive description of the software could be obtained. The user guide is more about how to use the software and its descriptions of the actual processes are schematic. There is thus no statement of the software's function that is remotely detailed enough to be compared against the source code. So computational scientists tend to turn to the third sense of verification in order to check their code.

The third sense of verification is the comparison of the simulation output with mathematical results. This is only possible where what is being modelled is a mathematical function with a known analytic solution. In the process of developing a model, therefore, it is standard practice to set up a simulation to solve mathematical equations with known analytic solutions. In one sense these checks are very strong because there is very high confidence in the mathematical data the simulation output is compared to. In HU's words, “the best thing you can have is an analytical solution”. But in another sense it is weak because in something like fluid dynamics you have to model a highly idealised system in order for there to exist a known analytic solution against which to compare. So verification only applies to a very special sub-set of systems you might want to model.

70 This point has been argued by Gramelsberger, ‘What Do Numerical (Climate) Models Really Represent?’.

Some philosophers have taken all verification to be a mathematical operation, a matter of "purely mathematical problems, which, as such, have to be solved at a formal level"[71] and it is easy to see that this territory is that of the first and possibly also second senses of verification. But there is an empirical side to the third sense of verification which is quite distinct from its purely formal dimensions. The first sense compares with the mathematical theory of the equations to be solved, the second sense with the specification. But in actually building software, a lot of additional input is required, drawn from a variety of sources. In contrast with the first two, the third sense requires real working code, and the actual running of a simulation. Once the software is operational, it is not so easy to conceptualise it as a formal structure on analogy with theory or specification. This dimension has been well emphasised by Eric Winsberg:

> "While these models generally incorporate a great deal of the theory or theories with which they are connected, they are usually fashioned by appeal to, or inspiration from, and with the use of material from, an astonishingly large range of sources: empirical data, mechanical models, calculational techniques (from the exact to the outrageously inexact), metaphor, and intuition. In the end, the model that is used to run the simulation is an offspring of the theory, but it is a mongrel offspring. It is also substantially shaped by the exigencies of practical computational limitations and by information from a wide range of other sources"[72].

When TT describes verification as a "test that your software does exactly what your PDEs do", the term "test" is well advised. It is a test of the software, of the equipment. Some of the raw material for this test comes from formal mathematical proofs of the answer to the case to be compared with. But the testing itself is about finding out about what it is that has been made when the scientists have built their code. In this sense, HU suggested this third sense of validation could be called "analytical validation" rather than verification.

We turn now to validation. Validation involves using statistical measures of fit between simulation output data and empirical data to build confidence that the simulation is correct, or at least that no major errors are affecting results.

> "When we started doing this work... there was not much in terms of applications and validation. There were lots of numerical proofs but there wasn't so much in the way of real world validation. It was never stated in this way but I took it as being part of the bargain that if those guys help me to do my science then I would make sure that my science helps them as validation" (KA)
>
> "There are different types of validation – the best is where someone has done an experiment and tells people what the experiment involved but you don't tell them what the result is, and then they go away and run their models and come back with their results and you compare them – in some fields that might work depending on how simple the model is and how many numerical stability or numerical accuracy knobs you have in your model." (TX)

71 Frigg and Reiss, 'The Philosophy of Simulation', 602.

72 Winsberg, *Science in the Age of Computer Simulation*, 30–31.

One of the fundamental tensions in validation is captured in this statement by TX. In his view, the data against which the simulation output is to be compared should ideally be completely cut off from it until the very last moment. That data is a precious resource, and must be kept separate to get the maximum validation. Randall and Wielicki, both atmospheric physicists, complained for this reason about "tuning" and "calibration" of simulations, which in their eyes "artificially prevents a model from producing a bad result"[73]. They are critical of attempts to adjust a model that is producing a wrong result, so that it will produce a correct one. But in many cases the current state of knowledge about the thing to be modelled is not nearly exhaustive enough for a scientist to be able to build a working model without going through an iterative process of testing, comparison, tweaking and indeed calibration. And those cases where such exhaustive knowledge is available are rarely the ones generating the most scientific interest. What Randall and Wielicki point out is an overstatement, but it is quite true that where processes of calibration have been part of a simulation's genesis, stronger justification is needed, a stronger collection of validation and verification results[74]. The question is not merely whether the simulation is producing the right results, but rather whether it produces the right results for the right reasons. This is where it is important to distinguish between two different meanings of validation.

The first sense of validation is the validation of a simulation, which is direct, holistic and specific to that simulation. This is the straightforward, more intuitive use of the term: what is validated is the simulation. On the other hand, what scientists are usually interested in when they do validation is not just that simulation, but rather the modelling framework that has been used to build it. This is indirect and partial validation. Two simulations built with the same framework may use different parts of its functionality and what functionality they do have in common is likely to be stressed in different ways in each case. So the validation of the modelling framework is indirect, operating through the intermediaries of the various simulations that are generated using it. It is partial because each validation only tests part of the framework, and does so in its own way. And it is piecemeal, built up from many different overlapping validation projects that all add up to different levels of confidence in the different aspects of the framework. Some are common to many projects, and thus will have been thoroughly tested many times. Others may only recently have been developed, and have only been deployed in one or two projects. "It is a cumulative process of validation," says IW. "You can increase your confidence in your results and your confidence will increase over time as you try different things and validate different aspects."

73 Randall and Wielicki, 'Measurements, Models, and Hypotheses', 404; see also Pickering, *The Mangle of Practice*, 14, 20–21 for a more positive spin put on the metaphor of tuning.

74 Data that could be used for validation is also a precious resource for building the simulation in the first place. In cases where the phenomena being modelled are not thoroughly understood, the builders need all the resources they can get their hands on. But this creates a potential problem for validation. A colleague of TX's, PH, accepted that it would be better if data was hidden until the last minute, but it is not practical in the nuclear engineering community, where information about the systems in question is often quite scarce. As a half measure, some scientists partition their data, such that half is used in order to build the simulation, and half are left unseen to be compared with for validation once things are up and running.

> "The validation stage is quite a critical stage I believe for Fluidity-ICOM. If I publish this validation, and SS has published her validation, you can say that "Fluidity-ICOM can do this". And they are necessary steps in the development of a model, because from this, from flow past a cylinder [a key test case], people can say "MA did this, so I can then do a more complex model with the wind and temperature and so on"." (MA)

The wider sense of validation is not directed towards particular simulations, but rather towards the parts of the software framework and the various techniques embodied in it. For Fluidity, validating its adaptivity algorithms was a key outcome of the lock-exchange project (discussed at some length in chapter 6, 'Images in/of Simulation'). Just as modelling frameworks are loose assemblages of techniques, put to work in different combinations for different problems, and tested to different degrees by different problems, validation and verification in general is a matter of bringing a loose assemblage of successes to bear on the future reliability of that system. This is something Winsberg expressed eloquently:

> "Whenever these techniques and assumptions are employed successfully – that is, whenever they produce results that fit well into the web of our previously accepted data, our observations, the results of our paper-and-pencil analyses, and our physical intuitions; whenever they make successful predictions or produce engineering accomplishments – their credibility as reliable techniques or reasonable assumptions grows. This is what I meant when I said that these techniques have their own life; they carry with them their own history of prior successes and accomplishments, and, when properly used, they can bring to the table independent warrant for belief in the models they are used to build"[75].

In this quote we can see the piecemeal picture of validation extended even further. Further to all these different validations, there are many different sources of legitimacy, which are drawn together around a modelling framework and attach to the elements within it. These range from predictive successes, to mathematical proofs, to the reputation of the community that built it. We will encounter many factors, through the argument that follows, that have a bearing on the legitimacy of modelling frameworks, and it is important to recognise that these do not add up to a single homogeneous whole, but rather a miscellaneous assemblage of different sources of justification.

The final point to make in this section is about benchmarking because that has a slightly ambiguous status.

> "I have... been working with people across the globe to run all our codes for the same problem and see if we all converge towards the same solution: a kind of community benchmarking" (IW)

While validation and verification confront the simulation with mathematical results and empirical data, benchmarking uses these encounters in order to bring a modelling framework into articulation with other computer models. These usually involve comparing their respective verifications or validations. There are many different ways of doing benchmarking, from attempting to reproduce a

75 Winsberg, *Science in the Age of Computer Simulation*, 122.

result that another group has already published, to a combined project in which several groups attempt to simulate the same experiment at the same time, with the experimental results revealed only when their various simulations have been run. Benchmarking is in these cases is a question of validation, but what matters is the co-ordination of the validation of different simulations and modelling frameworks so that they can be compared.

2.13 Beyond verification and validation

In an influential critique, Oreskes et al. showed that no verification and validation could be complete, against the truth-implying connotations of the terms[76]. But rather than take this to undermine the legitimacy of simulations, we should take it as inspiration to look instead at the dynamic of simulation practice, in which we can see validation as an ongoing, partial, and piecemeal process, one which strives to move beyond its domain, and makes chancy moves out into what cannot be validated.

> “If you are talking about a really complex system then it is more important just to validate each one of the separate components, make sure they match up with what you know experimentally. Then you can go on from there and do something much more complicated. Typically your ambition there is to go beyond what could possibly be done by experiment” (TT).

All validation and verification is limited because it only compares to specific cases, and these are often not the ones that my informants tended to most want to find out about. There is a generalising inference, or as HU puts it below, a “hope”, that as successful validations are accumulated, grounds are established for more confidence that simulations can carry some of this legitimacy with them into the domain of problems that are not so easily tested.

> “I run several simulations and I check whether the solution converges to the analytical solution – so if somebody decreased the level to an infinitely small number then eventually you would get the right answer. You can't show that this is therefore valid for all domains; you only show that for that specific example it converges. One hopes that it implies something more for other domains” (HU)

This “hope” is the bread and butter of computer modelling; the straightforwardly validated cases are those that have experimental analogues, and are thus already amenable to experimental study. It is in pressing beyond these that simulation goes further than experiment. But even while simulationists push beyond what can be validated, there is a creative requirement to think of analogous cases that can provide the best route to take you out into the unknown.

> I always get my PhD students to try and think of a scaled down version of their projects that can be or has been produced in the lab. For example I had a student who was looking at how much heating you get when two asteroids bang together. You can't do this in the lab. But what you can do is take a powder and whack it with a lump of metal, and you can work out

76 Oreskes, Shrader-Frechette, and Belitz, ‘Verification, Validation, and Confirmation’.

> how fast you need to whack the powder in order to melt the powder. So I got my student to try to reproduce that experiment. (TX)

Having validated based on an experiment in which powders were melted by a projectile, a “rung in the ladder” is established, in the words of HP, one which validates the key aspects of the model that will be stressed in the asteroid simulation. Quite what indirect validation is appropriate for a given case is an important question. In a recent project of KA's, he makes the claim that validation for simulation of ancient oceans is better done by running another simulation of the present oceans because while there is lots of empirical data about the ancient seas in the study of rock formations, there are aspects of oceanography that will not be reflected in the sediments at the bottom of the ocean. “We are arguing that if you look at the rocks you can't validate because the rocks aren't telling you what the water body was like” (KA).

> “I would say that the way to validate for an [ancient] ocean problem is that you validate in the modern, which I think is the most important... so I am not too worried that we can't validate in the ancient... That is the point of using a model; sometimes the model should be telling you things that the rocks can't” (KA)

Simulation science moves out on a limb, bringing with itself a collage of different kinds of supports. It is not an exact science, but a practical endeavour, developing its techniques in a variety of domains, hoping they hang together, and taking its chances.

2.14 Applied modelling and the science of method

> “Everybody is always looking for new methods because that new method gives you a new insight – there is this whole class of problems that they can't get at because they haven't had the device” (KA)

The final feature of scientific practice in a simulation laboratory that I want to treat in this introduction is the divide between two orientations of research: towards applications and towards methods. When I talk about applications, I mean studies of empirical or ideal problems (for example, ocean problems, or cases like flow past a cylinder, backwards-facing step, and many others we will encounter). When I talk about methods, I mean studies of the techniques used in making simulations, which are also often studied through running simulations.

> “It is a classic chicken and egg situation with the mathematical/numerical/computational methods and applications: which drives which. There is obviously a lot of intellectual curiosity about things at each of these levels because the basic mathematical algorithms have their potential originality: the computational numerical methods integration with parallel computers, they have their originality. The applications have their own highly original elements as well. The nature of academics is such that if they have the ability there will be the drive to exploit them.” (GT)

Applications and methods are interdependent, methods being tested out by being applied, applications being realised through the trying and testing of methods. The shifting of techniques from being objects of analysis to being tools to be deployed as "black boxes" in the investigation of some other object, is the kind of thing Merz calls the "multiplex" nature of models. She points out that it is therefore always necessary to pay close attention to how the research is oriented[77]. It has been too easy to follow the intuitive view, which would allow the applications side to stand for simulation science in general, making the false assumption that what scientists want to find out about when they run a simulation is the system that their simulation represents. This is indeed true in many cases, but in many other cases, the research is better understood as being set up in order to use this relationship between the simulation and its target to find out about the properties of the algorithms and techniques that have been integrated into the model. For a model under development, such as Fluidity, this work on simulation methods is a major output, and the contribution that the researchers who build Fluidity are making is not just to the applications areas that Fluidity can be applied to. It is to the wider community of simulation builders, who are interested in what techniques work well in what circumstances. Techniques can travel, apart from the modelling frameworks in which they are implemented, and can come to be integrated into many different systems for doing computational science.

The main divide at AMCG is between "applications people" and "methods people". Those who identify with the empirical sciences tend to fall into the former category, while the mathematicians tend to fall into the latter. There is a significant amount of tension between the two orientations because while a model under development is very much under the control of the methods people, there was a general resentment that the funding for all computational science research largely revolves around applications instead.

> "This is one of the biggest issues of the computational science area and we discuss it a lot round here... What we [methods people] are really passionate about is getting nice software and then studying the algorithms and proving the results and things like that. We enjoy looking at applications but we are not experts in the applications, yet we spend all our time writing proposals that are about the heart, or about the earth, etc. It is frustrating because it is much easier to get funding for people to run your code than to develop new ones... It is very hard to provide the supportive infrastructure to study a new finite element method. You have to say "I want to make a better wind turbine". It is cool to be involved in those kind of projects and it is really important but it is just that it is not where most of the work is going in. It is not recognised by the scientific community or the management of the college as being important. Quite often what happens is that these proposals get written, then funding arrives for postdocs and PhDs, and then we try to hire good programmers and hope that they can get to grips with the application and do that and do some of what we are interested in as well." (CE)

One of the advantages of looking at science in practice is that it helps to undermine quick assumptions that would otherwise condition the analysis, premises that are shared between philosophers who study simulations in terms of their applications, research councils who fund

77 Merz, 'Multiplex and Unfolding'; Sismondo, 'Models, Simulations, and Their Objects', 256.

research, and management structures which concern themselves with the distribution of people and things in the research organisation. Instead of following this premise, we instead start with an open perspective which takes into account the duality of orientation that defines what kind of thing is the locus of interest, and what counts as merely a practical device used to investigate it.

2.15 Conclusion

This chapter has built up a general picture of the kind of scientific work that is involved in making and running simulations. What is left under-studied here, however, is what exactly it is that we mean by “practice” in scientific practice. We are studying research in computational physics, but more needs to be said about the status and significance of this domain that is the centre of attention. These issues are fleshed out in the next two chapters, which will prepare the way for the remainder of the thesis.

3 Practice in Theory

3.1 Introduction

This chapter provides an initial theoretical discussion of "practice", a concept which will be indispensable in what follows. The first section locates practice within science studies. The middle of the chapter goes into some detail about the philosophical stakes of a commitment to practice. The final few sections draw out some of the consequences for our analysis.

The concept of practice is extremely useful for formulating what it is we are interested in when we study science. One of its most significant assets is the fact that no single school of thought or discipline can claim ownership of it. A whole constellation of approaches loosely gather around its theme. This is an asset, but it is also a potential problem, for these approaches are not unified in any straightforward way. There are many different theories of practice, and many routes we can take. The good thing is that there is no dogma to be accepted, and thus great freedom to use the concept of practice according to the exigencies of the empirical site.

This chapter sets up practice as a category through which to understand science. The chapter that follows this, chapter 4, will identify ways in which we can read a theory of practice as a certain kind of rationalism, and thus revive a concept of reason as a tool with which to understand simulation.

3.2 Why practice?

Science studies is in a state of theoretical aftermath. During the 1990s, this field of study was defined by big debates. Battles were fought over issues of social construction in what became known as the "Science Wars", while more private but often very hard fought struggles carved out the contours of stylistic and theoretical schools of thought[78]. Although science studies has its origins at least two decades earlier, with the first big wave of scholarship in the 1970s[79] and much theoretical

78 See, for example, Collins and Yearley, 'Epistemological Chicken'; Callon and Latour, 'Don't Throw the Baby Out with the Bath School!'; Gingras, 'The New Dialectics of Nature'; Pickering, 'In the Land of the Blind'; Gingras, 'From the Heights of Metaphysics'.

79 Barnes, *Scientific Knowledge and Sociological Theory*; Barnes and Shapin, *Natural Order*; Bloor, *Knowledge and Social Imagery*.

novelty emerging through the 1980s[80], the 1990s saw it come to maturity as an emerging discipline in its own right. These tumultuous times have now died down, their initial force weakened, not necessarily because scholars moderated their positions or compromised, but rather because they came to accept their differences as irresolvable.

Most of the ideas that drive contemporary work in science studies have their origins in the 1980s and '90s. With much less novel theory in play at the moment, it must be asked whether the discipline no longer has quite the vigour of the years of its coming of age, for with maturity comes heritage, and current writing must situate itself within a tradition, flesh out the consequences of inheritance, or forge something new. But if science studies' youthful vigour has waned slightly in the last few years, at the same time it became the focus of increasing outside interest, as a source of fresh theoretical inspiration for many scholars not directly interested in science. Science studies approaches have enjoyed such a prolific diffusion across disciplinary boundaries that they seem now to rival post-structuralist philosophy as a generic source of “theory” in the humanities and social sciences[81].

In emphasising the concept of practice, I want to take a step back and advocate a wider perspective on science studies theory. The danger with any theoretical movement is that it overplays its hand and portrays itself as a radical break with the past, a complete overturning of traditional categories. In time, it will always be possible to reflect on such moments and show them on the contrary in their continuity with established modes of thought. This is very much the case now with Latour's posthumanism, his radical rethinking of what it means to be modern[82], and to a certain extent with Pickering's desire to initiate an appreciation of a “new ontology”[83]. These theoretical moves, however radical, can always be read as part of an historical tradition, not as a break from it, or, if a break, a break within it.

While the big debates between the Edinburgh School (Bloor, Barnes, etc.), Bath school (Collins, Pinch, Yearley, etc.) and the French “actor-network” theorists (Latour and Callon) raged over questions of the meaning of the social, the construction of knowledge and the nature of interests and constraints, a quieter consensus was being built in the background, not quite equated with either, but stimulated by the insights of both. Similarly, while philosophers found themselves at odds with social scientists on the question of reason and construction, an approach started to emerge which promises, if not reconciliation, at least some common ground on which to agree or disagree. This was the foregrounding of “practice” as the central category of analysis, “scientific practice” as an object of study both for philosophers and for sociologists[84]. The concept of practice successfully navigates between the twin perils of post-positivism, the cultural construction of science and the

80 Shapin and Schaffer, *Leviathan and the Air-Pump*; Latour and Woolgar, *Laboratory Life*; Knorr-Cetina, *The Manufacture of Knowledge*.

81 For example, Bennett, ‘The Force of Things’; Harman, ‘The Importance of Bruno Latour’.

82 Latour, *We Have Never Been Modern*.

83 Pickering, ‘New Ontologies’.

84 Pickering, ‘From Science as Knowledge to Science as Practice’.

theory ladenness of observation. It allows us to appreciate each, but without reducing the real complexity of doing science to simplistic viewpoints. Pickering reflected on the emergence of this current in the 1990s:

> “In retrospect, then, I can see much of my own work as an exploration of this neglected side of pragmatism, an inquiry into practice *in its own right*, without a pregiven presumption that the end of enquiry has to be an argument about knowledge. And, to put it simply, the upshot for me was a gestalt switch into what I call the performative idiom. The argument of *The Mangle of Practice* was that if there is a sun around which all else revolves, it is performance, not knowledge – knowledge is a planet or maybe a comet that sometimes participates in the dynamics of practices and sometimes does not, and the discovery, for me, was that practice has its own structure that one can explore and talk about – as a dance of agency, for example”[85].

This chapter introduces practice as the major theoretical frame for the thesis. It will also set up chapter 4, 'Reason in Practice', for it is on the basis of an appreciation of practice that we might turn back to the great ancestor of science studies, Gaston Bachelard, and see in his writings not just an early glimpse of what would come some decades later, but a wider interpretation of scientific practice as a rationalism, an interpretation of science in practice in terms of what he would call “phenomenotechnique”. With his recent studies of historical epistemology, Rheinberger has already suggested this as a productive line of enquiry[86]. I would claim that not only does it give us some excellent tools with which to approach science in action, practice also provides us with a broader frame with which to appreciate the greatest advances of science studies, opening up the question of what scientists actually do to critical scrutiny, and thus inviting us to seek reason in the actual sites of accomplishment. Putting more recent scholarship in the context of this older tradition is not a matter of doing it down by claiming it was reiterating what was said before, but rather of showing it to be part of a bigger, more powerful historical movement in thought, a general move to practice.

3.3 Humans and non-humans

One of the key features of the practice-based view is that it moves from an “anthropocentric” epistemology in which knowledge is textual and theories linguistic, to one in which technologies and bodies share the stage as indispensable aspects of what it means to do science. This is particularly important for present purposes because simulations raise questions of anthropocentrism with great force. These vast technical systems easily eclipse the cognitive capacities of individual humans, of their creators and manipulators. For Paul Humphreys this is simulation's most philosophically significant aspect. “The computations involved in most simulations are so fast and so complex that no human or group of humans can in practice reproduce or understand the process”[87]. He consequently argues that humans are displaced from the centre of the scientific endeavour:

85 Pickering, *The Cybernetic Brain*, 380–381.

86 Rheinberger, *On Historicizing Epistemology*; Rheinberger, *An Epistemology of the Concrete*.

87 Humphreys, ‘The Philosophical Novelty of Computer Simulation Methods’, 620.

> "For an increasing number of fields in science, an exclusively anthropocentric epistemology is no longer appropriate because there now exist superior, non-human, epistemic abilities. So we are now faced with a problem, which we can call the *anthropocentric predicament*, of how we, as humans, can understand and evaluate computationally based scientific methods that transcend our own abilities"[88].

Simulations cannot be directly understood. They are too fast and too complex. In chapter 7 I will show how big software places us in a situation in which conventional paper media are irredeemably insufficient to "capture" the movement of this research. We move, with simulation, to a world of big technologies, of directly transmitted systems of code, humans working inside such a wider frame, not masters of it, but dwellers within it.

It is important on this point to acknowledge a major theoretical distinction that is implied by the concept of practice. While we may draw a lot from the actor-network school of thought, practice theory remains, despite its de-centring of the human, too anthropocentric to be classed as part of that movement. The human, in a theory of practice, may be enmeshed in a concrete world of bodies and things. But the human does remain special. The key thing to grasp is the difference between the extreme symmetry of the semiotic orientation of Latour and Callon[89], and the practice orientation of other scholars such as Pickering. Pickering outlines the difference:

> "Semiotically, as the actor-network insists, there is no difference between human and nonhuman agents. Semiotically, human and nonhuman agency can be continuously transformed into one another and substituted for one another. I am not alone in thinking that there are serious problems with these ideas when it comes to the analysis of science... I find it hard to imagine any combination of naked human minds and bodies that could substitute for a telescope, never mind an electron microscope, or for a machine tool, or for an atom bomb (or for penecillin, heroin, …). Semiotically, these things can be made equivalent; in practice they are not"[90].

In the narratives provided by Latour, perhaps most notably in his *We Have Never Been Modern*, practice theory will appear to be the less radical option, more modest and indeed more "modern", than semiotic approaches[91]. But from the point of view of practice, we are interested in the fundamental constituents of human existence, and semiotics, however eloquent its analyses of signs, however flat its networks of hybrids, can only provide one component. As Schatzki puts it, "the nominalism of actor-network theory bars recognition of any wider entity that actions make up or of any constitutive context in which actions take place. Actor-network theory thereby fails to capture a key feature of human social life, namely, the practices that are tied to arrangements and help constitute social phenomena"[92]. This is not to say that the semiotic tradition cannot provide deep insight into the role of relations and things in science. In chapters 5 and 6 I draw substantial help

88 Ibid., 617.

89 Callon, 'Some Elements of a Sociology of Translation', 200.

90 Pickering, *The Mangle of Practice*, 15.

91 Latour, *We Have Never Been Modern*.

92 Schatzki, 'Materiality and Social Life', 135.

from semiotics, but always return to practice, for the former can only really tell us about the relations of things, and cannot give insight into the motivations, inspirations and rhythms of research in the making.

It is worth following this discrimination from actor-network theory with a comment on heterogeneity, a term commonly used in science studies. Scientific practice is extremely complex, involving many different elements. It is this that makes it interesting. But it is important not to fetishise heterogeneity, as if saying that a situation is heterogeneous is to say anything of interest, beyond the fact that it is interesting. As analytical tools, the term suffers from a troublesome ambiguity. A truly mixed up hybrid is homogeneous, and an understanding of the system of differences that underlies its composition requires additional empirical insight, such as the observation of the processes of its creation. On the other hand, anything whose constituents are distinctive enough to be identified as different elements in a heterogeneous whole is not really a hybrid at all[93].

Another key co-ordinate in the contemporary theoretical landscape, which helps us to locate practice, is “assemblage theory”, espoused by thinkers such as Paul Rabinow and Manuel de Landa[94]. de Landa's recent book on simulation attempts to draw out an ontological vision of reality based on emergent properties, of the kind that are commonly accessible to simulation studies[95]. From this kind of viewpoint, there is little place for practice. The world is a complex and chaotic system of systems and there is to be no privileged level of analysis. While this project may have admirable goals for understanding reality, de Landa singularly fails to get to grips with the kind of science through which we explore such emergence. The irony is that while he wants to develop a sophisticated emergentist realism, he does so only by adopting a naïve realist standpoint towards the simulation science he relies on. There may be, as will be clear in what follows, a space for understanding the contemporary social field in terms of assemblages, because this concept foregrounds the materiality of the social. But again, this materiality must be nested within a broader consideration of practice.

Having gained an initial feel for where practice sits within science studies, we now look further afield, and start to identify its advantages for a comprehensive analysis of research.

3.4 The significance of practice

One of the initial advantages of the concept of practice is the possibility it offers for drawing on resources in the philosophy of science as well as in the sociological traditions of science studies. The concept gained wide currency in philosophy of science in the second half of the Twentieth Century. Indeed, the general movement away from formal studies of scientific theories in the philosophy of

93 Gingras makes similar points using the metaphor of cake. Gingras, ‘Following Scientists Through Society?’.

94 Rabinow, *Anthropos Today*, 49–56.

95 De Landa, *Philosophy and Simulation: The Emergence of Synthetic Reason*.

science has been described as a turn towards looking at “science in practice”, a more “case study” inclined kind of philosophy. The “Society for the Philosophy of Science in Practice” counts among its founding members several very significant figures in contemporary philosophy of science, of which Nancy Cartwright and Hasok Chang are perhaps the most famous[96]. While by no means uncontroversial this turn toward practice is now a mainstream position. Furthermore, its salience to the present study is clear from the fact that a lot of the trend towards practice was equally a trend towards foregrounding the role of models and modelling in philosophy of science[97].

What is not so clear in the philosophy of scientific practice, however, is what we might call the “ontology of practice”. For too many of these philosophers, practice is treated as something of a self-evident concept with little need for philosophical scrutiny. It is simply “what scientists actually do”.

But “what scientists actually do” is an empirical question, requiring empirical study. As should be evident to any philosopher, there is no straightforward way to understand any empirical phenomenon, and this is especially the case when that phenomenon, being social, is highly complex in nature and deeply affected by methods of observation and intervention. The great benefit of the social sciences is that it has long been recognised that methods require as much attention as the objects to which they are applied. Perhaps it is because philosophy has not traditionally conceptualised itself as empirical study, that we have seen less similar attention to these reflexive methodological questions. But this has to change if a philosophy that looks at “what scientists actually do” is to be sustainable.

It is also necessary to point out to philosophers that the availability of the term “practice” cannot be taken for granted. Its currency is relatively recent in social science, becoming prevalent only after the 1970s[98]. This history is significant to its use. Social scientists have not always studied practice and many still would resist conceptualising their object of study in these terms. It does not sum up empirical research in general. The most notable alternative tradition, of which Durkheim, Parsons and Luhman are often evoked as representatives, involves a much more “holistic” view of society, in which “what scientists actually do” is only a passing concern, one significant only for what it tells you about the logic and function of social systems or social structures[99]. Structuralism involved similar kinds of holism. The other major rival to practice theory is methodological individualism[100]. Like practice theory, individualisms also favour a local rather than holistic level of explanation, but they go much further in this direction, and tend to idealise complex contexts of action in order to model them according to simple mechanisms of decision and choice[101].

96 See http://www.philosophy-science-practice.org/

97 Morgan and Morrison, *Models as Mediators*; Gelfert, ‘Model-Based Representation’.

98 See, for example, Ortner, ‘Theory in Anthropology Since the Sixties’.

99 Schatzki, ‘Introduction: Practice Theory’, 14.

100 Schatzki, *Social Practices*, 6.

101 See, for example, Becker, ‘A Theory of Social Interactions’; and more recently, works such as Bueno de Mesquita and Lalman, *War and Reason*.

Practice doesn't name a specific school of study, but rather a diverse constellation of approaches united by their twin refusal of holism and individualism, and a desire to focus attention on an irreducible level found in between. Somewhere between the abstractions of the individual and society is the field of practice, and this is the primary reality to which practice theorists address themselves. Bourdieu and Giddens are often named as prominent practice theorists, as is Foucault, when he isn't being tarred with the brush of “postmodernism”. Charles Taylor, Ernesto Laclau, Chantal Mouffe and Jean-Francois Lyotard are also commonly mentioned[102]. It is, in fact, as a strong alternative to postmodernism that practice theory gains its greatest contemporary relevance. But its greatest asset is also its greatest challenge: the sheer diversity of the resources at its core. This is something apparent from its two most commonly named ancestors: (late) Wittgenstein and Heidegger, both of whom provide in their own ways some of the most profound resources for thinking about the accomplishment of action within concrete circumstances, and about the background against which such actions are performed, while also exhibiting deep differences in style and in substance.

Practice theory is not just one theoretical option for studying the same basic object as its rivals. For a practice theorist, there is no such reality as “society”, no transcendent social realm. Similarly, there is no reduction of the embeddedness of action in concrete circumstances that will provide a meaningful concept of the “individual”. The difference between these theories there is a fundamental disagreement about what kind of thing it is that we are studying. This choice is not about choosing one option over another for studying the same object. It is a matter of choosing the basic orientation that defines the field in the first place. Because the question is so fundamental, we need to be able to ask: What kind of reality is there to practice?

3.5 The reality of practice

> “It is my belief that all theories of social life either explicitly contain and maybe also discuss, or constitutively presuppose ontological understandings”[103]

The fundamental ontological issue at the heart of practice theory is hard to avoid. One of the central strands of the critique of holism, often the reason for dissatisfaction in the first place, was the question of the ontological status of society. For Durkheim society exists on a level transcending everyday reality, a level from which we gain our categories and institutions, which is the source of laws, sanctions, prohibitions, and hence of religious reverence[104]. To him, there is a reality called “society”, existing above and beyond the interactions and psychologies of individuals. Sociology as a discipline is distinct from other social sciences (especially from psychology) because its object and its explanations are located at this transcendent level: how is society possible? Structuralists, such as Claude Leví-Strauss, would reformulate Durkheim's project, but would retain this higher level: What

102 Schatzki, *Social Practices*, 11.

103 Schatzki, ‘Materiality and Social Life’, 124.

104 Durkheim, *The Elementary Forms of the Religious Life*.

structures govern language, myth or kinship[105]? What you are studying is the system, and this is the reality of concern. The systematicity of structure and the functional interdependence of institutions provided explanations for cohesion at all levels.

When Bourdieu develops his theory of practice, he questions this transcendence, asking what kind of reality these systems and structures actually have[106]. In seeking to answer this question, he looked to the theoretical practices of the analysts who derived these concepts. Holistic transcendent domains of order, said Bourdieu, are an effect of the abstractions made by a distanced observer: “in taking up a point of view on the action, withdrawing from it in order to observe it from above and from a distance, he constitutes practical activity as an *object of observation and* analysis, a *representation*”[107]. Such objectifications are second-order phenomena, not a primary reality of social life. They are real in the sense that they are genuinely to be found in the armchair or office of the social scientist, but not real in the sense of operating independently in the field, with a transcendent reality. Bourdieu will accept the fact that these kinds of wider systematicity can be derived from empirical analysis of social life, but for him this potentiality to yield systematic description is something to be accounted for, not something to be assumed already in operation under the surface of events.

The major challenge for any holistic social science is to provide an adequate account of the ontology of the whole. Latour's famous critique of the concept of society is in this respect not particularly novel, and this critique is not specific to his theoretical approach[108]. His rhetoric is eloquent and stirred many scholars up towards a fresh wave of critique of the historical legacy of their disciplines. But what is for him was principally an attack on the Durkheimians of the Edinburgh School is a reconfiguration of this older challenge. If society is a thing about which it is appropriate for us to talk, which admits of explanations and questions, then what kind of thing is it? In what sense does it really exist outside of the objectifications of the social scientist? Having raised these challenges, any theory of practice must equally rise to the challenge of explaining the ontology of this quite different sort of thing, this field of practice, which it holds up instead as a more solid ground for social scientific enquiry. This ontological challenge is to be embraced, for it is here that we might explore some of the most profound features of a theory of practice.

The emphasis for me here will be on the reality of the field of practice, the concrete multiplicity through which action is accomplished[109]. Several prominent theorists of practice tend to take a slightly different stance, in which the primary objects are practic*es*, conceptualised as distinct items

105 Lévi-Strauss, *The Elementary Structures of Kinship*.

106 Bourdieu, *Outline of a Theory of Practice*; I am more interested in Bourdieu’s classic analysis of practice than in his later applications of his ideas to science. For example, Bourdieu, ‘The Peculiar History of Scientific Reason’.

107 Bourdieu, *Outline of a Theory of Practice*, 2.

108 For example, Latour, *Reassembling the Social*, 27; cf. Ingold, *Key Debates In Anthropology*, 55–98.

109 An interesting analogy may be found here with the concept of the ‘situation’ in pragmatism. For example, Brown, ‘John Dewey’s Logic of Science’, 268–273.

within a social universe[110]. These practices, however, seem to be already abstracted from their site, and this is not the kind of approach I would take. I would talk about practice, not about practices, as if they are entities. What is real is the field of practice, not objectifications such as practices.

A first element of the reality of the field of practice is comprised by the bodies that dwell within it, through which interactions occur. The greatest effect of Bourdieu's theory of practice was to bring the body centre stage. It allowed us to start talking about the role of embodiment in social interactions. There is an important parallel with Foucault's writings on discipline, on how social institutions act upon the body[111]. It opened up a whole field of research into how different kinds of discourses, situations and institutions make people into certain kinds of people[112]. But for Bourdieu, it is not just that power acts upon the body. The body is fundamental to the regularity of action, it is generative. Embodiment captures the real processes of sedimentation of complex strata of dispositions that condition what is done, how it gets done, and how situations are felt and perceived[113].

It is, however, necessary to push Bourdieu further. It is too easy, in his account, to regard the body as a substrate onto which social forms are stamped. This is encouraged by the schemes that Bourdieu drew up in his early work, diagrammatic systems that were supposed to represent the orientations of a culture[114]. There are two problems with this. Firstly, there is no “blank slate” of the body (in the manner than Locke had proposed for the mind)[115]. We might like to draw on Deleuze's account of bodily organisation and disorganisation for a more nuanced view. For him, the unity of the organism is undone by its limitless folds[116], or by productive flows of desire that connect and disconnect in manners subversive to unification[117]. It is always intricately textured. Secondly, as Stephen Turner has forcefully pointed out, practice theory cannot rely on any theory of dispositions that requires that dispositions somehow be “shared” by many individuals[118]. Were such an assumption to be made, and it is arguable whether it is or is not implicit in Bourdieu's account, we would be forced to account for some sort of mechanism by which dispositions can be transferred between individuals, and it is hard to see what such a “copy and paste” mechanism would be.

The beginnings of a consistent view of practice emerge when we start from the principle that the kind of sedimentation we are talking about does not exist on the level of the individual body. If this is granted then the above problems fade away. Individual bodies are always part of larger configurations of learning. While individuals can practice skills on their own, there is always a

110 Schatzki, *Social Practices*.
111 Foucault, *Discipline and Punish*.
112 Hacking, ‘Making Up People’.
113 Bourdieu, *Outline of a Theory of Practice*, 78.
114 Ibid., 157.
115 Thrift, *Non-Representational Theory*, 61.
116 Deleuze, *The Fold: Leibniz and the Baroque*, 8–9.
117 Deleuze and Guattari, *Anti-Oedipus*.
118 Turner, *The Social Theory of Practices*, 44–77.

communal component. Actions are done and demonstrated in many ways from childhood onwards. Habits are formed together, and they are varied down to the smallest scales. The reality of practice is not the reality of a singular body, but a field of many bodies. Skills and dispositions are not transferred between bodies, but co-constituted in a field of becoming. Such fields will clearly overlap and change, and have very fuzzy edges. Practice is something that occurs in an arena of bodies that is always multiple and mutable[119]. Their coherence will vary, as some co-ordinations are long-established, others emerging on the fly.

The most substantial reality fields of practice have is one and multiple. It is always possible to find other contexts, to divide and multiply. Marilyn Strathern points out that social phenomena display a curious "fractal" property, in which each scale of analysis exhibits a similar level of complexity[120]. If one were to "zoom in" from a grander consideration of science to a more detailed analysis of its sites of accomplishment, a new terrain would emerge, which would in turn invite scrutiny at new levels. This is something that my informants at AMCG are well aware of, for modelling is always a matter of choosing a resolution. Whatever the scale of your grid, there are always below grid-scale phenomena that are impossible to explicitly resolve.

This may appear to be a flimsy reality, but the reality of social phenomena can hardly be expected to be analogous to the reality of a stone[121]. It takes a sophisticated social theory to handle the complexity of social life without distorting the object of study. The body is a particularly difficult aspect of practice to talk about. As François Sigaut put it: "[t]he knowledge built into a machine can always be retrieved, at least in theory. But we still seem ill-equipped to identify the skills embodied in our own nervous system"[122]. Despite the decades of study dedicated to the phenomenology and sociology of the body, work remains to be done. We must not mistake the fuzziness of the field of practice for a deficiency of the theory, for it is precisely this fuzziness that we endeavour to grasp. The social nowhere exists neatly, and attempts to tidy up its edges, to theorise for example closed and independent cultures, are only destined to failure. To this end, I draw together many different sources of theory: poststructuralist philosophy, phenomenology, philosophy of science, social theory, in my attempt to work out a theory of practice capable of doing justice to the empirical material.

The question of what kind of role the body should have in a theory of practice is one of the most important, but it is important to stress that the many other kinds of materials of the laboratory also play a fundamental role. Embodied action is always action within a world of materials[123]. Things play an active role[124]. Nersessian writes of devices in science as "hubs" around which the different dimensions of practice are co-ordinated[125]. Once we can talk about embodiment, it is a small step to

119 Thrift, *Non-Representational Theory*, 8.

120 Strathern, *Partial Connections*, xxi.

121 But see Ingold, 'Bringing Things to Life', 19 for an admirable attempt to complexify the reality of even a stone.

122 Sigaut, 'Technology', 438; see also Latour, 'How to Talk About the Body?'.

123 Ingold, 'Bringing Things to Life', 19–32.

124 Carusi, 'Computational Biology and the Limits of Shared Vision'.

125 Nersessian, 'How Do Engineering Scientists Think?'.

point out that wherever dispositional milieux are developed, so too is developed a distribution of ever changing materials, modifying them, using them up, combining them, wearing them out, a complex world of interactivity. As with bodies, the relevant fields in which practice occurs will be vague, multiple and overlapping. The material reality of practice is not one of a singular objective world, but rather of assemblages drawn together into a zone of relative consistency.

One of the best ways to start thinking about the depth of conditioning of bodies and things would be to draw on Leibniz's philosophy, albeit read through very contemporary eyes, such as we find in Deleuze's interpretation[126]. Leibniz wants to establish a rationalism, in the sense that he wants to claim that innate forms exist, but he is wary of describing these as a distinct finite set (such as Cartesian Ideas or Kantian categories). For Leibniz, the spirit is infinitely textured, and so is the material world. “Thus it is that ideas and truths are for us innate, as inclinations, dispositions, habits, or natural potentialities, and not as actions although these potentialities are always accompanied by some actions, often insensible, which correspond to them”[127]. Furthermore, Leibniz anticipates the importance of the unconscious, for “acquired habits and the stores of our memory are not always perceived”[128].

When looking at scientific practice, phenomenology provides another angle on the accomplishment of research[129]. Heidegger's analysis of being-in-the-world that he gives us in *Being and Time* provides a way to think about the structuration of a world by concerns and commitments, and thus about the way in which fields of projects are held together along the lines of projects[130]. Heidegger's vivid descriptions of the primacy of the existential phenomena of being-in, being-in-the-world, being-with and thrownness capture the way in which the person is never primarily present to him or herself, but rather is already tangled in a world, is already thrown into commitments, and never exists in the abstract, but rather is only insofar as he or she fundamentally is *there*: Da-sein. A theory of practice must follow this lead, and refuse any decisionist theory of the human in which the person first of all assesses the world, then acts. The actor never “arrives at” a world in which to act, but is immersed in involvements that tangle any intentional moment in a wider web of significance.

The use of Heidegger is here most significant for its ability to insert this dimension of motivation into the discussion of practice, to respond the question that Knorr-Cetina poses: “how can we theorize practice in a way that allows for the engrossment and excitement—the emotional basis—of research work?”[131] Pickering, too, pointed to the role of desire in scientific practice[132]. We could emphasise for example the Heideggerian theme of *care* to talk about the intensive attention my informants devote to their technologies, their digital systems through which they do their work, a

126 Deleuze, *The Fold: Leibniz and the Baroque*.
127 Leibniz, *New Essays Concerning Human Understanding*, 46.
128 Ibid.
129 See, for example, Krieger, *Doing Physics*.
130 Heidegger, *Being and Time*.
131 Knorr-Cetina, ‘Objectual Practice’, 184.
132 Pickering, *The Mangle of Practice*, 1.

care that goes well beyond a mere utilitarian concern for keeping them functioning so that science could get done. So much is personally invested in these things that they are emotionally laden and the work of carefully tending to them is an end in itself. I also use the term *anticipation* to talk about the future-bound concern of work within real projects, projects which may have a plan, yet will rarely be determined by it, which are profoundly open in their possible development, yet retain a very tangible sense of “going somewhere”, an already thrown going somewhere beyond any plan[133]. We can relate this, using Rheinberger, to the effect of the juxtaposition of fields of practice, each of which in its own way partially grasps some aspects of an elusive and constituently vague object, the *epistemic thing* towards which practice is oriented[134]. This will be explored more when we get to chapter 9, 'Stability and Surprise', which seeks to return to the more theoretical themes, and explore further the question of motivation and concern in research practice.

3.6 Skill

We now move on to deal with the question of what a theory of practice might have to say about action, and what conversely it might say for the *accounts* of action that can be produced alongside it.

The analysis of a field of practice is a difficult operation. What we want to capture are precisely those dimensions of what happens that are not easily described, that cannot be reduced to a prior tabulation of the sequence of steps to be carried out or a teleology based on the final result. In both cases, we gain some understanding of the practical activity, but still remain at considerable remove from its actual occurrence. In both cases, objectifications of practice loom large over its actual accomplishment.

Tim Ingold has been a vocal critic of these kinds of approaches in the anthropology of skill. In this area of study a theory of practice is very pertinent because skill transmission has often been regarded as a matter of passing on rules, of codes for conduct, to new learners. Skill is therefore often treated in a dualist manner, with the actions conceptualised separately from their accomplishment. In contrast, Ingold deems it necessary to embrace the constitutive enfolding of mind and body in practical engagement.

> “Merely to witness the finished works, or even the successive steps in their construction, does not suffice to enable novice observers to copy these steps for themselves. My contention is that to explain how they manage to do this requires us to shift our analytic focus from problem-solving, conceived as a purely cognitive operation distinct from the practical implementations of the solutions reached, to the dynamics of practitioners' engagement, in perception and action, with their environments”[135].

In addition to this critique of abstract understandings of action, Ingold writes of the necessity of “reversing a tendency, evident in much of the literature on art and material culture, to read creativity

133 While Heideggerian in inspiration, this use of anticipation departs from his analysis of being-toward-death.
134 Rheinberger, *Epistemic Things*.
135 Ingold, ‘Beyond Art and Technology’, 29.

'backwards', starting from an outcome in the form of a novel object and tracing it, through a sequence of antecedent conditions, to an unprecedented idea in the mind of an agent. This backwards reading is equivalent to what Alfred Gell has called the *abduction of agency*"[136].

Gell's work has been hugely influential in the anthropology of art. For Gell, the central problem is social interaction, and the role of objects in interaction[137]. He is only really interested in the actual creation of things insofar as that process is important for the role they play in social interaction. Ingold's critique arises from his feeling that this perspective obscures creativity, something that he wants to understand on its own terms, as something of interest quite aside from the social role of objects that may or may not be thus created.

It is illuminating to note that the two anthropologists are not so much disagreeing about the nature of practice, but about what questions about practice it is interesting to ask. If we are interested in action as a creative becoming, then it is counterproductive to look at finished objects or rules for the procedure, because of the danger of establishing a retroactive teleology. But if we are interested in the role a created thing plays within a field of practice, as an element conditioning further action, then Gell's approach is very informative. For Gell, the cognitive significance of art objects derives from their indexicality, and from the way that as an index an art object supports complex configurations of abductive reasoning.

"Abduction," Gell explains, "covers the area where semiotic inference (of meanings from signs) merges with *hypothetical inferences* of a non-semiotic (or not conventionally semiotic) kind..."[138] Gell uses the example of the friendliness inferred from a person's smile[139]. This is a precarious inference. It is logically illegitimate. But it is practically efficacious, and abduction enables us to appreciate the complex web of agencies that can be evidenced by created objects, as various producers, recipients, commissioners and subjects are woven together in the play of agency manifest in the existence of the particular object.

The advantage of abduction is that it lets us talk about inferences that are not governed by systems of convention, so they are broader than the kind of meanings that we get from theories of language. Abduction cannot be wholly appreciated from within a rule-bound understanding of meaning. It is inherently open and operates within spaces of inference that have no determining principle. But there is a fundamental incompatibility between this theory and Ingold's. We can never appreciate the indeterminacy of creativity through a concept of abduction because with the latter we always start with a finished object, and read backwards towards its origin. In this direction we inevitably loose something of the potentiality that was manifest in the actual creative process.

136 Ingold, *Being Alive*, 215.
137 Gell, *Art and Agency*.
138 Ibid., 14.
139 Ibid., 15.

3.7 Temporality

The two-fold of Gell and Ingold reflects the asymmetrical temporality of practice[140]. It is important to grasp sites of practice in terms of their temporalities. Time here is not an external metric, a co-ordinate system, within which practices are accomplished, but an intrinsic property of practical fields, which have their distinctive rhythms, cycles of fast and slow, pauses, stoppages, and harmonies with neighbouring systems[141]. Pickering will stress that it is within the unfolding of practice that major features in the landscape of possible research futures emerge. He dislikes the language of constraint and of other Durkheimian theories of science because they posit external forces imposing on practice, as if the latter were to be pressed into their mould. "... let me emphasise that unlike constraint – which seems to me somehow "already there" – resistances *emerge* in the real time of practice..."[142].

The time of research practice is of course regularly objectified in write-ups of method, plans and proposals, and such objectifications, while they stultify the perception of the flow of practice, are important means by which scientists grasp what has happened in the unfolding of their actions. Much research gains its tempo from looking ahead at what is demanded by a representation of the future course of action. As Pickering says, "I find I cannot make sense of the studies that follow without reference to the intentions of scientists, to their goals and plans..."[143] But he goes on to "stress the temporal emergence of plans and goals and their transformability in encounters with material agency"[144]. While plans and goals are extremely important objectifications, they are mutable aspects of research, liable to change according to the unfolding dynamic[145]. They don't predetermine what is going to happen. "There is no algorithm that determines the vectors of cultural extension, which is as much as to say that the goals of scientific practice emerge in the real time of practice"[146]. In these quotes, we grasp the essence of Pickering's departure from "sociology of scientific knowledge" approaches – for these, it is the accounts that are the focus-point, to be explained via an analysis of practice, but in the end they are the primary object of study. Here, instead, it is practice in all its breadth and diversity that is the primary object, within which processes of accounting and the artefacts they produce are only one feature[147].

It is extremely important to grasp the interplay of objectified representations of practice and the transformability of their being put into practice. The explicit goals of research processes do not predetermine the route or eventual destination. This is, in Rheinberger's words "...a movement that is

140 See Taylor, *Philosophical Arguments*, 176.
141 Schatzki, 'Materiality and Social Life', 135; Urry, 'Complexity', 112; On resonance, see Rheinberger, *Epistemic Things*, 225.
142 Pickering, 'Beyond Constraint', 51.
143 Pickering, *The Mangle of Practice*, 17.
144 Ibid., 18.
145 Callon, 'Some Elements of a Sociology of Translation', 207.
146 Pickering, *The Mangle of Practice*, 19–20.
147 Ibid., 12–14.

not goal directed, but nevertheless anything but chaotic for that reason”[148]. While practices are not predestined this is far from a case of “anything goes”. It takes a lot of regulation and close co-ordination to achieve a balance between order and chaos, a question we will return to at the end of the thesis, in chapter 9.

The time of practice was also a major feature of Bourdieu's theory. His critique of functionalism and structuralism was a critique of their external standpoint, from which they would only be giving post-hoc objectifications of what happens. Instead, Bourdieu wants to embrace strategies. He builds on Marx's insight, that “[m]an's reflections on the forms of social life, and consequently also, his scientific analysis of those forms, take a course directly opposite to that of their actual historical development. He begins, post festum, with the results of the process of development ready to hand before him”[149]. Bourdieu writes: “To substitute *strategy* for the *rule* is to reintroduce time, with its rhythm, its orientation, its irreversibility... For the analyst, time no longer counts: not only because – as has often been repeated since Max Weber – arriving *post festum*, he cannot be in any uncertainty as to what may happen, but also because he has the time to totalise, i.e. to overcome the effects of time”[150]. A similar temporality to practical circumstance was recognised by Michel de Certeau, when he wrote that a tactic, operating outside of propriety, “is always on the watch for opportunities that must be seized 'on the wing'”[151]. There is a basic indeterminacy in the unfolding of practice to which it is immensely difficult for any account (including this) to remain true.

The operation of objectification, of accounting for what happens with maps, plans, descriptions and schemes, thus working counter to the temporal unfolding of action itself, is a characteristic feature of Western epistemic practices. But this by no means implies that it is the only possibility, or even the most appropriate. Strathern sums this up succinctly:

> “To describe one's social world as apart from the actions which constitute it, to set up procedures which deal with events as already enacted and closed to further modification, to see adjudication as not itself participating in the events under scrutiny all participate in a common philosophical position within western cosmology. Life is understood in terms of a split between representations (descriptions) of it, and as it really is”[152].

The point of any theory of practice is to open to question the form of description practised by the analyst, to ask what its effects might be, and how it might affect the objects that are rendered within its terms. The point is to set into interaction the forward and the backward, a dynamic emerging from the inherent lack in any retrospective account.

148 Rheinberger, *Epistemic Things*, 183.

149 Marx, *Capital*, 1:46–47.

150 Bourdieu, *Outline of a Theory of Practice*, 9. What is excluded from this quote is the sentence: ‘Science has a time which it not that of practice.’ This is excluded because Bourdieu is using the word ‘science’ to denote the objectifying viewpoint, rather than to denote the empirical domain of scientific practice.

151 De Certeau, *The Practice of Everyday Life*, xix.

152 Strathern, ‘Discovering “Social Control”’, 128.

3.8 Objectification

The critique of objectification is a critique of the social scientist's knowledge. But it is also an account of the real interplay within the fields of practice we are interested in studying, between improvisation and processes of planning and accounting native to them. We can expect that scientists, like social scientists, have trouble perceiving the indeterminacy of their research once it is obscured behind a totalised account. “Epistemic things, let alone their eventual transformation into technical objects and vice versa, usually cannot be anticipated when an experimental arrangement is taking shape. But once a surprising result has emerged, has proved to be more than of an ephemeral character, and has been sufficiently stabilized, it becomes more and more difficult, even for the participants, to avoid the illusion that it is the inevitable product of a logical inquiry or of a teleology of the experimental process”[153]. What Rheinberger describes is the trouble we have with abduction. When given an object to view, abductive reasoning provides the plausible causal explanation for its coming into being, and hence from this perspective, that becoming is filtered through the immediacy of its outcome.

Research in the making is a “groping movement which, from an anachronistic perspective, tends to be seen as a master 'strategy'”[154]. Using Derridean terms, Rheinberger will stress that “[t]here is, as a rule, no unique perspective that could account for the research movement with all its possible turns, no definite direction to its “blind tactics,” its “empirical wanderings””[155]. By giving a practice-centric account of research, we are not looking for a new totalising perspective, but rather a door onto the openness and indeterminacy of projects. We want to understand something of their raising of hope, of striving and searching[156]. No totalising perspective can do these processes justice, but plenty of totalising perspectives are available, as scientists routinely craft narratives of what they did when they did their research, in order to present an account of their method. These idealised accounts strip away the contingency, the strategy, and the tempo of research and provide instead an idealised description of the sequence of actions undertaken. The “tinkering” process is “lost and forgotten” in retrospect[157].

> “This way of conceptualizing the "actual events" in the sequence may seem plausible, natural, and even irresistible. Nevertheless, it is not an accurate historical description, if by "historical description" is meant an account that identifies the significance historical agents "attach to" the events in their life-world at a particular time. Instead, what we might call a "transcendental” vantage point equips the reader with a fore-knowledge of what was determined afterwards; a fore-knowledge that consequently acts as a backdrop for defining what the speakers in the transcript were "really" seeing”[158].

153 Rheinberger, *Epistemic Things*, 74.

154 Rheinberger, ‘From Microsomes to Ribosomes’, 72.

155 Rheinberger, *Epistemic Things*, 184.

156 Bloch, *The Principle of Hope*, 45.

157 Dunbar, ‘How Scientists Think’, 488.

158 Lynch, ‘Allan Franklin’s Transcendental Physics’, 482.

Michael Lynch here aptly summarises the issue, which is a great problem for creating a history, or for a philosophy of scientific process. But for scientists whose own accounts of their research provide such perspectives, it is not a problem. They are not historians. They are not attempting to describe in the methods section of their papers the indeterminacies of their practice. They are providing an objectified description, and it is our job as outsiders not to get carried away with it.

> “The retrospective view of the scientist as a spontaneous historian is not only concealing but in many respects also revealing. It reminds us that an experimental system is full of stories, of which the experimenter at any given moment is trying to tell only one”[159].

By paying attention, therefore, to the dynamics of objectification, and the interplay of accounting practices, we no longer have to view the “scientist's version” of events as a false history, as an opponent, against which science studies must fight in order to establish the legitimacy of a more contingency-centric viewpoint[160]. Geoffrey Bowker has recently made this kind of move, by shifting the terms “from the telos of recall and fold[ing] it into a reading of ways of being in the present”[161]. The important thing is to embrace the asymmetric interplay of practice and account within any present, a fundamental feature of our object, of scientific practice in general.

3.9 Conclusion

Theories of practice help us grasp what is at stake in the accomplishment of research. The next chapter looks to expand the broader theoretical significance of this practice view. By connecting scientific practice with the concept of “phenomenotechnique” we can draw on the history of rationalism, to develop a theory of practice into a theory of practical reason, one which can do justice to the epistemic character of the temporal unfolding of research and its self-articulation, its encounter with itself through its objects and objectifications. With Bachelard, we can embrace that process as rational, while avoiding the positivist trappings of the notion of progress.

159 Rheinberger, *Epistemic Things*, 185.
160 Pickering, *Constructing Quarks*, 3–7.
161 Bowker, *Memory Practices in the Sciences*, 21.

4 Reason in Practice

4.1 Introduction

This chapter argues that a theory of scientific practice can be understood as a certain kind of rationalism. I make the case for the advantages of rescuing the concept of reason from the cutting floor of the science studies toolkit.

By looking at the philosophy of Bachelard, we can appreciate reason as something immanent to skilled practice, reviving the sense of research as a rational enterprise. This proves powerful because it allows us to conceptualise research as something dynamic, with its own integral temporality, one different, perhaps, to those of other fields of skilled practice. This temporality involves a self-articulation, a dialectic of self with self, in which concepts exist in their realisation in systems for the creation of novelties, which fold back upon themselves for transformative effect.

Having written much of his epistemological work during the 1920s and 1930s, Gaston Bachelard is in many ways a lost ancestor of science studies. The philosophy he developed can be read as a philosophy of scientific practice, and thus starkly in contrast with Anglophone philosophies of science conventional at the time, which tended to detach theory from its sites of application according to Reichenbach's famous distinction between the context of discovery and the context of justification. Although he was not an ethnographer, Bachelard was exceptionally well informed of scientific practice, and formulated his philosophical project explicitly in a deep engagement with the radical developments in science that were going on around him (especially quantum theory and relativity). In returning to Bachelard's philosophy of science I am following the lead of several thinkers who have suggested that it is strategic to do science studies under the name of historical epistemology, a school of thought for which Bachelard was a central figure[162].

One of the consequences of bringing this longer-term view to bear on science studies is the possibility of formulating an approach that revives a certain kind of rationalism, a surprise in the

162 Rheinberger, *On Historicizing Epistemology*; Rheinberger, *An Epistemology of the Concrete*; It is interesting to note that Rheinberger’s colleague at the Max Planck Institute for the History of Science, Lorraine Daston, has also proposed the label ‘historical epistemology’ for her historical analyses, but in contrast to Rheinberger, Daston is not intending to refer to the French school also known by that name. See Daston, ‘Historical Epistemology’.

context of the Science Wars, in the wake of which science studies gained something of an anti-rationalist label. Rationalism under Bachelard's influence takes on a new tenor, and is substantially redefined. This redefinition emerges from a reformulation of Kant's critical philosophy within the context of developments in Twentieth Century physics. By situating his approach in terms of the major narratives of rationalism, I claim that the spin that Bachelard gives things avoids the major pitfalls in this history while maintaining itself in the position of heir to critical philosophy, one of the most important intellectual traditions of modernity.

I situate rationalism in terms of its two great dichotomies, in the process delineating my position. Then I move on to more explicitly outlining a Bachelardian science studies, showing how it articulates with more recent scholarship, how it may inform a theory of scientific practice. The aim is to obtain a broad theoretical framework for discussing reason in practice, a stepping-off point from which to grasp the empirical studies of simulation that form the substance of the chapters that follow.

4.2 Internalism and externalism

There are two major narratives about rationalism in tension during the Twentieth Century, one which sees it opposed to constructivism, and the other opposing it to empiricism. To appreciate Bachelard, it is necessary to unpick both.

Rationalism forms a context for contemporary science studies in terms of the internalism/externalism debate that pits sociological claims that social and political factors intrude on scientific activity against positions that maintain that scientific rationality operates by excluding such factors, casting them outside its legitimate domain, and thus proceeding down the single path of truth. In the debate, for example, between Michael Lynch and Allan Franklin, “reason” is treated as a pure domain of logical thought and action that becomes no longer reasonable as soon as extrinsic social or political values come into play[163]. In this light, the social sciences study social phenomena which are by definition outside the proper domain of the rational, so the very enterprise of a sociological study of scientific practice would therefore imply its object's irrationality. There would be no possibility of a social rationalism.

From a position more distanced from the Science Wars, however, Alan Nelson showed these opposed positions to be empirically irreconcilable: no example can ever be sufficient to furnish conclusive proof of the truth of the rationalist position nor of the constructivist position[164]. For any instance of scientific research, Nelson points out, it will always be possible for constructivists to maintain that conclusions were drawn in the way that they were because of contingent social conditions, in other words, if those conditions had been different a different outcome would have emerged. On the other hand the rationalist can always claim that if such influences did have an effect, it would only be short term and will be cancelled out in the long run by science's convergence

163 Lynch, ‘Allan Franklin’s Transcendental Physics’; Franklin, *Experiment Right or Wrong*.
164 Nelson, ‘How Could Scientific Facts Be Socially Constructed?’.

to the truth. As Hacking later put it: “Rationalists, at least retrospectively, can always adduce reasons that satisfy them. Constructivists, with equal ingenuity, can always find to their own satisfaction an openness where the upshot of research is settled by something other than reason”[165].

From the Hacking/Nelson point of view, the best resolution that can be expected is an agreement to disagree. However, both the original debate and Nelson's proposed resolution are premised upon the fact/value distinction. In each case, the dividing line between the internal facts (upon which reason is based) and the external values is regarded as anterior to the issue. The debate can easily be reframed once it is pointed out that what we might (as philosophers or sociologists) be interested in is how this division comes to be established. The two sides from this point of view become mixed. For Bachelard, value is at the heart of reason, for a science works though producing “its own norms of truth”[166]. Furthermore, rationality is a social phenomenon. “Individuals hesitate... but the school – in the sciences – does not. The school... drives research forward”[167].

Bachelard's pun popularised by Latour, “un fait est fait” – roughly “a fact is an artefact” – runs against the idea that a fact is something “out there” waiting to be discovered, already opposed to the values that are the artefacts of our relation to the world[168]. Far from isolating a world of facts and reason from a world of values and irrationality, Bachelard points us towards the cultural nature of reason itself. He did this, however, not in order to undermine the legitimacy of scientific truth, but to understand the nature of the real and radical scientific progress he perceived around him.

The opposition between reason and society (or, equally, reason and culture) must be overcome, and this much has been clear within science studies for quite some time, even without going back to Bachelard. Godin and Gingras argue that “one can transcend such a ruinous opposition... by seeing that argumentation is an essentially social practice inside a scientific field that is the product of previous history (that is, of previous argumentation and experimentation)”[169]. They point out that “there is no reason for any reference to 'logic' and 'argumentation' to be taken as *epistemological* rather than *sociological*”[170]. This argument for the relevance of sociology to logic is not a claim that epistemological ideas of reason should be abandoned in favour of a pure sociology of groups and interests in the vein of the Strong Programme of the Edinburgh School. To do this would be simply to retrench the opposition between reason and society. It means that neither a pure sociology nor a pure asocial epistemology will do, and in this respect Rheinberger takes a similar stance to Godin and Gingras, though he wishes to reclaim the term “epistemic” in the process: “epistemicity is one of the modes by which we humans enter into a particular relationship with the material world around

165 Hacking, *The Social Construction of What?*, 91–92.

166 Tjiattas, ‘Bachelard and Scientific Realism’, 206.

167 Bachelard, quoted in Rheinberger, *An Epistemology of the Concrete*, 32–33.

168 Latour, ‘Force and the Reason of Experiment’, 63.

169 Gingras and Godin, ‘The Experimenters’ Regress: From Skepticism to Argumentation’, 149.

170 Ibid., 150. Note that ‘epistemology’ need not be isolated from sociology. In contrast to Godin and Gingras’ use of the term, Bachelard and Rheinberger both understand epistemology as something already sociological.

us”[171]. Epistemology does not, therefore study something detached from the wider domain of practical goings on of human activities.

Reason is a primary marker of the core epistemic values of any scientific community, a target for a great deal of work aiming to uphold its rigour (“that seems reasonable”). Far from implementing a singular transcendental logic, scientists engage in a variety of strategies of argumentation and experimentation, each with a particular history and tied to material engagements with the environments they construct and maintain around them, what Pickering dubbed “disciplinary agency”[172].

In fact, while he has come into conflict with science studies, Franklin also proposed a more pluralistic conception of reason, outlining a number of different strategies through which scientists give rational weight to their findings and their arguments[173]. Hacking, too, drawing on Crombie, has been influential in looking at the historical emergence of different styles of reasoning “that settle what it is to be true or false in their domain”[174]. Once the apparent antagonism between reason and the social is defused, it becomes clear that each of these strategies will have their own particular histories, something that is especially familiar to those working within the field of computational physics. KA, now one of the most senior scientists at AMCG, had earlier in his career established himself within the field of palaeo-oceanography, and had no experience with computational methods before getting involved with the group. He explained his entry into this new area as an engagement with a radically new way of doing his science, but one which is situated in a history of many similar precedents.

> “I think geologists are always looking for new tools and I remember when the scanning electron microscope was first being rolled out. There were palaentologists using it to look at the structure of shells, and that was going on in the '60s... About a hundred years ago people first started using Uranium series data to date the age of the earth. People are always looking for a new technique that they can import in. There is a lot of history of doing that.” (KA)

New instruments came along periodically and opened up new possibilities. With them come new ways of making and justifying claims, new reasons for such work to be good or bad. For someone interested in prehistoric ocean circulations and tides, the use of computational modelling represented a new set of possibilities, but possibilities inseparable from the argumentative strategies through which they are brought into the academic fold[175]. KA elaborates on the forms of reasoning that were introduced with the new techniques:

> “Traditional geologists work backwards, from rocks, understood to be the result of an experiment that took place millions of years ago – working from the effect to the cause. But with the introduction of computational methods it was possible to start working in the other

171 Rheinberger, ‘Reply to Bloor’, 409.

172 Pickering, *The Mangle of Practice*, 114–115.

173 Franklin, *The Neglect of Experiment*.

174 Hacking, ‘Language, Truth and Reason’, 50.

175 Cf. Weisberg, ‘Who Is a Modeller?’.

> direction, constructing a model in which you can vary the initial conditions to see what will happen." (KA)

Quite aside from their individual strengths, however, it is the *combination* of computational with traditional geological techniques that KA regards as the most productive manner of science. A diversity of elements that can thus be drawn together into a specific research project, and these are mutually supporting. Just as a simulation may support KA in making his claims about ancient oceans, the degree of success of these claims reflects back on the integrity of the many elements and strategies that go into constructing his simulation. Past successes, reliable methodologies, numerical evidence, analytic proofs and community reputation, are all among the motley of strategies mustered around a simulation to back up its validity. A singular logic is ever elusive[176].

Writing of the use of Monte Carlo methods, Galison asks:

> "How should one class this type of... simulation? As experimental theory? Theoretical experiment? Is it a case of induction from data? Deduction from theory? Each such attempt to force the argumentative form back into the older categories strikes me as awkward, a rearguard action unable to capture the novelty of procedure. Once again, I would suggest that the Monte Carlo is best seen as expanding the spectrum of persuasive evidence, a tertium quid"[177].

Strategies are historically contingent and bound up with the communities that foster their legitimacy, and the techniques by which they are implemented. But the objectification of such strategies can never do justice to the processes of their application. As we saw in the previous chapter, it is always going to be possible to map out legitimate techniques for argument and for conduct in the laboratory, but these must be understood in interaction with the tempo of real research, rather than obscuring it. A scientist is expected to put these sanctioned strategies into practice in the performance of research, but the measure of success is as much the particular "practical mastery" of this performance as it is the legitimacy of the strategies that make it up.

Bourdieu talks in the context of gift giving of "strategies exploiting the possibilities offered by manipulation of the tempo of the action – holding back or putting off, maintaining suspense or expectation, or on the other hand, hurrying, hustling, surprising, and stealing a march, not to mention the art of ostentatiously giving time ("devoting one's time to someone") or withholding it ("no time to spare")"[178]. A similar account can be given of the practices of implementing strategies of reasoning – to talk about the social aspect of epistemology is not simply to talk abstractly about norms but about the activities in which those norms are made concrete. Behind the legitimation of rules and strategies, reason can be located in terms of the actual practice of their effective deployment. Rheinberger, drawing on the writings of Claude Bernard and Ludwig Fleck, puts it eloquently:

176 Winsberg, *Science in the Age of Computer Simulation*, 21; Galison, *Image and Logic*, 747.
177 Galison, *Image and Logic*, 747.
178 Bourdieu, *Outline of a Theory of Practice*, 7.

> “As Bernard... has put it, “one must have felt one's way for a long time... have been mistaken thousands and thousands of times, in short, have grown old in the practice of experimentation.” To feel one's way requires *Erfahrenheit* on the part of the experimenter. “Being experienced,” as Fleck uses the expression, is not simply “experience.” Experience enables us to judge a particular piece of work or a particular situation. Being experienced enables us to literally embody the judgement in the process of making new experiences, that is, to think with our body. Experience is an intellectual quality. *Erfahrenheit*, that is, acquired intuition, is a form of life.”[179]

Reason is something developed in scientific practice, not an overarching *a priori* faculty or method, for reason will change as methods change. Bachelard argued that the notion of a unitary method presents a great obstacle to reason, for reason progresses by working *against* itself.

> “For Bachelard a method is static by its essence. It sheds a distorting light. In fact, every single science forms its own method and permanently adapts it to its field, to its object, according to the hurdles it needs to get over in order to impose its rationality”[180].

“Scientific culture”, Bachelard wrote, “must bring about profound modifications of thought”[181], modifications that are expressed historically by the grasp of the past as error. It is not a matter of mechanically applying a fixed reserve of legitimate techniques, but pushing towards the greatest modifications. “For science, truth is nothing other than a historical corrective to a persistent error, and experience is a corrective for common and primary illusions. The intellectual life of science depends dialectically on this differential of knowledge at the frontier of the unknown”[182].

At this point, we need to pause, and address an important critique of Bachelard, which may otherwise obstruct his relevance to contemporary science studies. In their review of French epistemology Bowker and Latour claim that Bachelard's theory relies on a far too strict opposition between science and non-science[183]. Bachelard's rhetorical style is indeed of this nature, constantly phrasing his understanding of scientific practice in terms of what distinguishes it from non-scientific or pre-scientific practice. But if we read the demarcation of science and non-science in terms of a “differential of knowledge at the frontier of the unknown”[184], it is clearly not a difference that can be decided by an outside observer. Rather, it is internally produced, as scientific practice articulates itself against itself. Summing up this view of reason as an intrinsic dynamic of practice, against any objectification of method, Bachelard enigmatically tells us that “[r]eality is never 'what we might believe it to be': it is always what we ought to have thought”[185].

179 Rheinberger, *Epistemic Things*, 77; as the reference to forms of life indicates, very productive parallels can be drawn between this practice-oriented theory and the later work of Wittgenstein. Charles Taylor has analysed the connections between Bourdieu and Wittgenstein to great effect: Taylor, *Philosophical Arguments*, 165–180.

180 Bolduc and Chazal, ‘The Bachelardian Tradition in the Philosophy of Science’, 81.

181 Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 10.

182 Bachelard, *The New Scientific Spirit*, 172.

183 Bowker and Latour, ‘A Booming Discipline Short of Discipline’, 723–724.

184 Bachelard, *The New Scientific Spirit*, 172.

185 Bachelard, *The Formation of the Scientific Mind*, 24.

In other words, it is unnecessary for philosophers and sociologists to constitute an inside/outside boundary between the rational and the irrational; scientists are already at work doing just this. If this is part of their practice, the internalism/externalism debate breaks down, because there is no prior topology capable of defining the difference. Working on and with an evolving milieu of styles of reason, scientists are constantly dividing things (ideas, arguments, data, equipment) into what is legitimate and illegitimate.

The question of boundary maintenance is one which inflames Isabelle Stengers against both externalism and internalism. For to claim that the difference between science and non-science is “socially” or “externally” constructed requires the analyst to disregard scientists' own procedures of demarcation, and indeed to claim that while these procedures are real, they are nevertheless attempting to demarcate something which from the outsider's privileged perspective can be seen to have no ultimate justification.

> “If the response to the question “Is it scientific?” is a construction of scientists, it is not the fruit of an agreement among scientists, deciding among themselves something a detached observer can recognize as always undecidable. The gaze that sees the same, the undecidable, where those he is observing have as their raison d’etre to create difference, is the gaze of power”[186].

Just as Stengers is unimpressed with the perspective of an observer claiming to occupy a position transcending the level of actual research, a theory of practice implies a critique of approaching the analysis according to any *a priori* problematic. For Rouse, the problem with many modern standpoints: positivism, historicism, instrumentalism and constructivism, is that they all take for granted that there exists a problem of legitimacy raised by science as a whole[187]. For him, a focus on practice might offer a way beyond these trends, by looking at research without framing the enquiry in monolithic terms, without assuming that the key issues are already predefined. Legitimacy may come to be an issue at certain moments, but the science/non-science boundary does not frame practice in general.

None of this amounts to a denial that there are many social and political issues peppering the research of a group like AMCG, and these are negotiated as and when they emerge. Climate change and mitigation agendas, oil and nuclear industry agendas, government priorities all feed in to the way in which work is funded, prioritised and publicised. My informants did not carry on with their work in blissful ignorance of intervening factors, but worked hard policing the boundary.

The political priorities that govern research council grants have at times exerted significant influence on the direction and composition of AMCG. The nuclear team, for example, was very significantly pared back by funding cuts in the 1990s and is only now starting to grow again with recent renewed political interest in nuclear energy from programmes such as “Keep the Nuclear Option Open”[188].

186 Stengers, *The Invention of Modern Science*, 73.

187 Rouse, *Engaging Science*, 8, 21–25, 53–68.

188 It remains to be seen what effect the political fallout of the Fukushima disaster of 2011 will have on this kind of research.

Meanwhile, the oceanography and climate side experienced a huge boom in resources. While there is more industrial funding in nuclear engineering than climate science, this often has strings attached. The tolerance of industrial caveats on research varies on an *ad hoc* basis, generally according to whether it impedes what is recognised to be the appropriate freedom for a proper scientific investigation. In one case, for example, of a simulation of a medical isotope reactor, they “can publish on the cooling coil and the fissile solution, but not on the actual reactor that [the client is] making” (AY). Publishing certain details of industrially funded projects would give away their secrets to their competitors, but a complete block goes against the openness that scientists usually demand[189]. Furthermore, working on a project without being able to publish would seriously compromise the career prospects of the researcher in question. But sometimes a balance is struck. KA said:

> “If a company came up and said “we would like you to do X but we don't want you to publish it” then it is almost like that is a pointless bit of work. But if we were not bothered about publishing it then we might take it on. It can be worth the money. Recently for a 30,000 quid [industry project with a publication restriction] we persuaded the department to give it to us without overheads and that employed a post-doc for a two months, during which he was also able to get some other stuff done as well.” (KA)

Other significant issues are deeply engrained in the research at AMCG. For example, a lot of the information about nuclear reactors that is used for making nuclear safety simulations is kept confidential. This research therefore contravenes the widely held value of reproducibility. It is impossible to reproduce results of a simulation of a nuclear reactor if the design of that reactor is not published. But industrial intellectual property stands in the way, as do political interests in controlling international access to these technologies. Nuclear simulationists tend to point out that their work is important, and needs to be done, despite the limitations that these factors impose. There is therefore an internal/external division at stake in science, but one which is produced and reproduced inside the laboratory, rather than being an abstract division between the rational and irrational, accessible only to analyst-spectators looking in from the outside.

4.3 Empiricism and rationalism

In its other great dichotomy, rationalism finds itself opposed to empiricism as reason is opposed to experience. This is very different to the opposition with constructivism, and indeed from this point of view, constructivism can be read as a particular form of rationalism, for the constructed nature of scientific knowledge implies active intervention within its formation, as opposed to the passive reception of data from experience[190].

The opposition to empiricism concerns the origin and foundation of knowledge, and we owe this distinction largely to Kant's legacy, his critical philosophy carving a third path between the

189 Cf. Galison, ‘Computer Simulations and the Trading Zone’, 140.

190 Rheinberger, *Epistemic Things*, 102; see also Knuuttila, ‘Models as Epistemic Artefacts’, 25n.

rationalists (Descartes, Spinoza, Leibniz) and empiricists (Locke, Hume, Berkeley). The debate is often stated in terms of the origin of ideas: Locke's famous “tabula rasa” view of the mind traces every idea back to some originary experience, whereas rationalists, including on this matter Kant himself, maintain that there is some innate structure to the mind, and thus that ideas are possible that do not come from experience, but are “built in”, so to speak. While the empiricist would therefore have to base every legitimate claim to knowledge on its relation to experience, the rationalist would assert that in some cases it is possible to gain knowledge from reason alone, as in Descartes' ontological argument for God's existence according to the inherent necessity of the idea, or transcendental knowledge from the necessity of the form of any possible experience, in Kant's synthetic *a priori*.

While there are certainly problems with grouping philosophers into these two camps, different nuances of each of their positions, and indeed many influences crossing these boundaries, the opposition carried great weight through to more recent times. Foucault, for example, argued for the continuing significance in the middle of the Twentieth Century of a “line that separates a philosophy of experience, of sense and of subject and a philosophy of knowledge, of rationality and of concept”[191]. In other words, the phenomenological and existentialist traditions were to be opposed to the historical epistemology of Bachelard, Canguilhem, Cavaillès and to a great extent, Foucault himself[192]. Far from being an objective fact of philosophical orientation, this division is always invoked for certain polemical purposes, but its polemical power derives nevertheless from the fact that it is effective as a method of organising positions[193].

Like Kant before him, Bachelard's philosophy can be regarded as a new kind of rationalism forged as a third way between empiricism and older rationalisms; the “chief characteristic” of Bachelard's middle way is “a strong union of experiment and reason”[194]. Kant had restricted rationalism to the conditions of possibility of experience and knowledge, thereby opening a space for knowledge of *a priori* rational structures, knowledge that while *a priori* is nevertheless synthetic because it is only obtained through experience[195]. Bachelard rejects the fixity of the *a priori* that we find in Kant but his rationalism similarly proceeds through experience, according to his strict requirement that a scientific concept must be applied. If for Kant, we discover the transcendental concept through experience, for Bachelard, we overturn the concept through experiment. Echoing Kant's dictum that transcendental knowledge is always synthetic, for Bachelard a legitimate scientific concept cannot be unshackled from its application in an experimental set-up. In a famous passage that is worth quoting at length, Bachelard states:

191 Foucault, ‘Introduction’, 8.

192 See Gutting, *Michel Foucault’s Archaeology of Scientific Reason*.

193 Salanskis, ‘Phenomenology and Epistemology: War and Marriage’; see also Dosse, *History of Structuralism, Volume 1*, 3–9.

194 Bachelard, *The Formation of the Scientific Mind*, 69.

195 See, for example, Kant, *Critique of Pure Reason*, 48–49.

> "Conceptualisation [in science] totalises the history of the concept and actualises it. Beyond history and driven forward by history, it gives rise to experiments deforming a historical stage of the concept. What it seeks in experiment is opportunities for *complicating* the concept, for *applying* it despite the concept's resistance, and for realising the conditions of application that reality did not bring together. It is then that we understand that science *realises* its objects without ever just finding them ready-made. Phenomenotechnique *extends* phenomenology. A concept becomes scientific in so far as its becomes a technique, in so far as it is accompanied by a technique that realises."[196]

There is no place for concepts divorced from their being put into practice. While the form of this rationalism is similar to that of Kant, being immanent to the combination of reason and experience, the order is reversed. Rather than innate concepts structuring all possible experience, Bachelard is interested in the deformation of concepts opening new empirical domains. Whereas for Kant, the categories are fixed and immutable, for Bachelard they are under a process of perpetual revision.

> "Science is a discipline of active empiricism, which, rather than rely on whatever clear truths happen to lie ready to hand, actively seek its complex truths by artificial means. Innate truths naturally have no place in science. Reason has to be shaped in the same way as experience."[197]

Bachelard thus will call his "new rationalism" a "non-Kantism, that is to say... a philosophy inspired by Kant which transcends the classic doctrine"[198]. Bachelard undoes the immutability and closedness of the Kantian categories, in favour of an "open-ended philosophy" in which concepts indeed give form to scientific experience, but in so doing are put into a historical process of mutation[199].

This non-Kantism was inspired by the disturbance to Kantian philosophy brought about firstly by the development of non-Euclidean geometries, and secondly by their application in Einstein's theory of relativity. Even the structure of space, which for Kant was a transcendental condition of outer sensibility, and absolutely necessary, turned out to be revisable in light of new experimental findings. For Bachelard the crucial scientific movement is not the elaboration of the knowledge of a closed system, but the overturning of such a system. In his opinion, non-Euclidean geometry "'freed' rationalism by severing its psychological ties to a closed and immutable system of logic"[200]. Mary Tiles expressed it well when she claimed that Bachelard's philosophy is a "natural consequence" of the combination of Kantianism with "a rejection of the givenness and immutability of the structures of rational thought"[201].

Bachelard took a stand against the empiricisms of his day because he did not believe scientific objects are ever simply found in experience. They must be created. He will reject the founding of science on experience in general, while asserting that technically and rationally produced experience

196 Bachelard, *The Formation of the Scientific Mind*, 69–70.

197 Bachelard, *The New Scientific Spirit*, 171.

198 Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 12.

199 Tiles, *Bachelard*, 34; Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 7.

200 Bachelard, *The New Scientific Spirit*, 20.

201 Tiles, *Bachelard*, 17, 204–211.

is something very different, and mustn't be assimilated into an everyday empiricism in which every empirical object is “given” in the same way. “The traditional philosophical notion of a datum or *given* is highly improper to characterise the *result* of the laborious determination of experimental values”[202]. Hacking will later echo this sentiment in his claim that phenomena, in the sense that scientists use the term in their laboratory practice, are not the stuff of everyday experience. They are noteworthy, painstakingly created. They are rare[203].

But again, because of the distance between us and Bachelard, it is important to adapt his thought to contemporary conditions, and even more so to let it respond to the lenses through which he has tended to be read in the intervening decades. Bachelard is commonly associated with Kuhn. Although Kuhn's work postdated much of Bachelard's, the reception of Bachelard in the English speaking world was conditioned by Kuhn's theory of scientific revolutions and consequently Bachelard's thought was seen in this guise[204]. His other route into the Anglophone world was via the philosophy of Louis Althusser, whose Marxist and materialist reading of Bachelard also played up the revolutionary connotations of the notion of the epistemological “break”[205]. Both readings are no doubt productive, but they are particular readings that push Bachelard's thought through specific agendas. Most importantly, both stress the side of Bachelard's theory that emphasises the revolutionary nature of scientific practice. He was indeed living in tumultuous times for physical theory. But does that mean he would be less suited to times of less radical change? Or that he cannot be informative for understanding more everyday scientific practice? It is crucial to note that as much as he works from historical examples of theoretical revolutions, Bachelard's concept of the epistemological break equally applies to the much more mundane practice of being a scientist and learning to be a scientist, where it is necessary to continually work against the obstructions of everyday metaphors and intuitive abstractions. Bachelard was at least as interested in pedagogy as he was in the history of revolutions, and the overcoming of obstacles via a break need not be a moment of great historical significance, but may form the small moments of renegotiation in practice. A great historical moment may well have provided the impetus for Bachelard's break with Kant but it does not therefore follow that his philosophy is inappropriate for times of fewer such moments.

Bachelard's philosophy is a rationalism premised on the mutation of concepts and the application of concepts within experiment, so his rejection of empiricism nevertheless incorporates a fundamental role for experience as experiment. This direction takes rationalism a long way from connotations of the power of pure thought, and takes us to a powerful idea of the rationality of self-articulated empirico-conceptual engagements, the reason of scientific practice.

202 Bachelard, quoted in Tjiattas, ‘Bachelard and Scientific Realism’, 204.

203 Hacking, *Representing and Intervening*, 221–222.

204 Kuhn, *The Structure of Scientific Revolutions*.

205 Althusser, ‘Philosophy and the Spontaneous Philosophy of the Scientists’; Balibar, ‘From Bachelard to Althusser’; Fraser, ‘The Category of Formalization’, xvii.

4.4 Phenomenotechnique

The concept of phenomenotechnique has been hailed as one of Bachelard's most important contributions to philosophy[206]. The term captures the applied nature of his rationalism, a rationalism that asserts that scientific activities have their own inherent but open logics, logics of practice not of thought. Reason cannot be understood as a subjective faculty. For Kant, the categories through which we know the world can be discovered through a transcendental philosophy of our knowledge and experience, discovering the architecture of the transcendental subject. But for Bachelard this architecture becomes worldly and open. We make a move from the subject to worldly practices as the locus of the conditions for the possibility of science. Bachelard will repeat Kantian worries about the excesses of pure reason, about a dangerous tendency when unshackled from experience to depart on a flight of fancy, but for him this is the possibility for any concept deemed immutable, or any concept that does not “incorporate *a concept's conditions of application into the very meaning of the concept*”[207]. “A concept is scientific not on the basis of inductive derivation, but *only if* it is accompanied by an appropriate realization technique”[208]. Bachelard in this sense prefigured the “experimental turn” in the philosophy of science[209].

At the heart of phenomenotechnique is a dialectic between theory and technique, but this is not a Hegelian dialectic. It is an interdependency and a process of mutual revision, rather than formal contradiction. On the other side to his assertion that any scientific concept must be applied, is Bachelard's famous claim that any scientific instrument can therefore be seen as a “reified theorem”[210], that “the instrument of physics is a realized, concretized theory, rational in essence”[211]. The production of phenomena that occurs in experimental science cannot be understood as a special case of ordinary empirical experience. The production here operates at the level of the noumenon, producing a new real. “We can therefore say that mathematical Physics corresponds to a noumenology that is greatly different to a phenomenography within which scientific empiricism claims to encase itself. It is this very noumenology that elucidates a phenomenotechnique through which new phenomena are not simply found but invented, constructed and built from all parts”[212]. We will return to the contrast between a creative practice producing a real with a revelatory practice discovering reality in chapter 9.

One side of his opposition to empiricism is philosophical, and here he is contesting philosophers' views about the basis of scientific knowledge. But on the other side, Bachelard wants to draw out the perils of empiricist tendencies within scientific practice itself. Ordinary experience, he advises, must

206 Castelão-Lawless, ‘Phenomenotechnique in Historical Persepective’, 45.

207 Bachelard, *The Formation of the Scientific Mind*, 69.

208 Tjiattas, ‘Bachelard and Scientific Realism’, 207.

209 E.g. Hacking, *Representing and Intervening*; Shapin and Schaffer, *Leviathan and the Air-Pump*; Galison, *How Experiments End*.

210 Bachelard, quoted in Rheinberger, *An Epistemology of the Concrete*, 31.

211 Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 21.

212 Bachelard, ‘Noumena and Microphysics’, 76.

be cast aside in order to initiate proper scientific practice. Our ways of thinking are heavily conditioned by our everyday lives and these intuitive and metaphorical modes of thought can easily stand in the way of scientific work, which must proceed against them. In chapter 6, 'Images in/of Simulation', we return to this issue, looking at the role of intuitions in scientists' encounters with the results of their research.

At the heart of Bachelard's critique of intuition is the idea of the “epistemological obstacle” and the project of what he calls his “psychoanalysis” of the scientific mind. As Mary McAllester Jones has pointed out, this is to be interpreted not in a Freudian tradition, but rather in a general sense of therapy: “psychoanalysing objective knowledge means ridding it of everything that impedes its progress, whether affective interests or everyday, utilitarian knowledge, so restoring it to health”[213]. Because Bachelard has undone the closure of Kantian categories, forms of thought are historical and cultural, and can require great effort to be overcome. They must be put into experimental practice wherever they risk hardening into obstacles. “It is in the act of cognition that we shall show causes of stagnation and even of regression”[214]. Bachelard is therefore just as opposed to the idea of starting with a blank slate as he is to the idea of immutable categories, a problem in both philosophy and within science: “the scientist thinks he can start from a mind without structure and without knowledge; the philosopher more often posits an established mind, all equipped with categories indispensable to the understanding of reality”[215]. Phenomenotechnique – looking at actual research practice – avoids the pitfalls of both, for it posits the conditioning of research as a continual process of self-articulated becoming.

Bachelard believes in the disunity of science and refuses to provide a general account of method in the Cartesian vein. “[W]e shall ask philosophers to break with the ambition of finding one single, fixed point of view for judging the totality of as science as vast and changing as that of physics. We shall thus end up characterising the philosophy of the sciences as a philosophical pluralism...”[216]. Bachelard's successors would thus forge their own ways in the study of biology and medicine (Canguilhem), mathematics (Cavaillès, Badiou's early work), the social and human sciences (Foucault), and a less well fated attempt to do the same for historical materialism (Althusser). In the context of attempting to understand contemporary practices of simulation, we can regard this new suburb of the “scientific city”[217] to be adding to the existing diversity. Scientific concepts can be realised in many different ways, and simulation opens up another kind of opportunity for realisation, a new kind of phenomenotechnique.

213 Jones, ‘Introduction’, 3.

214 Bachelard, *The Formation of the Scientific Mind*, 24.

215 Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 7–8.

216 Ibid., 10–11.

217 Bachelard, *Le Rationalisme Appliqué*, 133; Cf Rheinberger, *An Epistemology of the Concrete*, 32.

4.5 Reified concepts

There is an important criticism of Bachelard that we must briefly dwell on, for it requires us to extend phenomenotechnique somewhat, or at least to situate scientific practice within a broader context. Galison criticises Bachelard for dubbing instruments “reified theorems”. For Galison, while this appears to hail a philosophy of instrumentation, it risks becoming “a rallying cry against the autonomy of instrumentation and for the all pervasiveness of theory”[218]. Galison is similarly unhappy with the anti-empiricism of philosophers like Kuhn and Feyerabend because it works through asserting the “theory laden” nature of observation. Although the concept of theory-ladenness serves to undermine empiricist notions of pure observations, it also has the effect of overdetermining instruments by theory. Galison raises a crucial point, because while the experimental turn required the displacing of theory-centric philosophy, older arguments such as Bachelard's were not pitted against theory *per se*, but were opposed to empiricism and positivism, a battle that could be fought to a certain extent within the presuppositions of theory-centrism.

The way forward is clearly a middle way, in which Bachelard is brought up to date with these currents in thinking about science. The neutrality of facts is to be denied but not because instruments are *only* reified theorems: they are much more besides. Galison points out that not only do we find different theoretical schools of thought, we also encounter different kinds of experimentalists, and different traditions of instrument makers, in interaction but also with a certain degree of independence[219]. The scientific instrument as reified concept takes on a different tenor when it is recognised that not only is it productive of scientific realities when employed within the laboratory, it has also been connected with distinct cultures of expertise in which it was originally crafted, connected with earlier forms of equipment and moving through various iterations of calibration and modification. Dominique Lecourt made a similar point about Bachelard's thought, noting that his technical theory of science begs the question of the relationship between science and wider histories of industrial production[220], a point which is particularly important in the context of today's globally distributed and interpenetrating technological systems[221]. We can certainly accept that Bachelard leaves certain questions unanswered, and this gives us scope and indeed good reason to expand his theory. But far from refuting phenomenotechnique, this modification to its central pillar serves to make it more sophisticated.

Computational physicists have a number of important relationships with other sites of technical production with other communities. Some of these are industrial, as in the manufacture of computer hardware. In chapter 8 we will encounter new computational devices, such as GPU architectures and accelerator chips, which are becoming increasingly important in supercomputer design, but both of which have a technical history that owes more to the video games industry than it does to scientific

218 Galison, *Image and Logic*, 18.
219 Galison, *Image and Logic*.
220 Lecourt, *Marxism and Epistemology*, 138–139.
221 Barry, ‘Technological Zones’.

computing. On the software side of things, a software system like Fluidity draws on programming languages developed for general purpose applications development, and uses libraries of functions that have been assembled through work in very different fields. The libraries of linear solvers, for example, provide an array of instruments for solving matrices on a computer. The fact that the matrices dealt with among the Fluidity scientists are generated by fluid dynamics problems gives them a particular flavour but many of the techniques that will be deployed from these libraries have been developed elsewhere. Their incorporation into a library has enabled them to travel and be co-opted into a huge range of different software systems. But it is important to note that as much as the scientists at AMCG utilise instruments made elsewhere, they also participate in instrument-making, adding, for example, new solvers to these libraries where they find them lacking.

It is also important to note that the slogan “reified theorem” would be badly misunderstood if it is taken to imply that Bachelard thinks theories exist prior to practical application, and thus serving as a source material for the crafting of instruments. On the contrary, theories for Bachelard exist only insofar as they are reified, and there is no need to assume that a given experimental system can be “translated” into concepts in any straightforward manner. We return to this issue in chapter 7, where the question of whether software systems can be described in publications is addressed. The experimental system is conceptual to the extent that it drives forward the research practice, insofar as it participates in its self-articulated unfolding, and there is no need to assume equivalence to a linguistic structure. Rouse sums up this departure from classic approaches to theory as a key aspect of the recent turn to practice:

> “Scientific theory is better understood in terms of theoretical *practices:* modelling particular situations or domains; articulating, extending, and reconciling those models and their constituent concepts and techniques; and connecting theoretical models to experimental systems. Such a conception of theorising diverges both from the classical sense of *theoria* and more recent analyses of theories as axiomatic or model-theoretic systems”[222].

The concept is not prior (and thus not prior to practice), but exists insofar as it is put into practice, an emergence which will inform and deform its structure towards an open future. In this respect, Bachelard's thought anticipates practice theory, in its generalised critique of the priority of anything (theoretical, methodological or transcendental) deemed to foreclose the field of practice. For Bachelard, as for practice theorists, it is practice itself that is the key domain. To the extent that an *a priori* is meaningful, it is immanent to this dynamic multiplicity.

4.6 Phenomenotechnique in practice

I now take a moment to expand on the concrete lessons of Bachelard's perspective for the understanding of the particular kinds of research that are at stake here. I want to show in brief outline that it allows us to capture the nuances of the research process. My example is a project modelling

222 Rouse, ‘Understanding Scientific Practices’, 444.

tidal barrages in the Severn Estuary. This project involved several scientists at AMCG, but the bulk of the work was being done by one PhD student, HU, a mathematician by background.

Because of the irregular topography of the Severn Estuary, when the tide recedes, what was once a single river splits into a number of isolated stretches of water, eventually to be reconnected at the next high tide. This is a problem for the current version of Fluidity because it cannot easily deal with splitting and joining domains. It is a challenge for most numerical modelling codes, and thus an object of study that is methodologically at the forefront of what is currently possible. What HU is developing, therefore, is a new method called “wetting and drying”, which will enable the accurate simulation of these phenomena. As with all such projects that require developments of novel techniques, research is oriented towards the methods as much as towards the applications that they facilitate.

In the building of computer simulation, the conceptual development of numerical and computational methods is fundamentally tied to their realisation in code, and it is through that realisation that they become objects to be worked upon further. It is through being crafted into a computational system of exploration and manipulation that things become the kind of things that computational scientists study. The scientist does not begin with a pre-formed idea of the final simulation in his head, but rather begins with a very rough plan of a trajectory that will successively move the conceptualisation of the problem and the development of the simulation through practical steps towards progressive realisation. The problem is a problem of method (how to simulate) as much as it is empirical (what to simulate). HU does not know at the beginning what the crucial factors will be within the processes of wetting and drying. His concept of the problem is general, but becomes more specific as he goes through the process of developing the model. Although we can talk about the Severn Estuary in a general sense, the specific Severn Estuary as it emerges through phenomenotechnical realisation is quite distinct. It is a mathematical system, conceptually uncertain and open, teased out in and through a process of investigation.

HU develops the model through a series of stages, testing his new technique for its properties in increasingly complex domains. The new software components he developed use a trick to simulate wetting and drying which involves maintaining the continuity of the domain despite its empirical separations, by allowing an extremely thin film of water to lie over the supposedly “dry” areas during low tide. The working hypothesis is that with certain boundary conditions imposed to prevent spurious flow across this film, the technique will facilitate the study of the dynamics of the estuary without introducing unwanted effects within the simulation results. Not only does HU not yet grasp the tidal estuary as a mathematical system, he also does not yet know how his new technique will interact with the many other techniques that others have implemented within the model: will it disrupt the balance of approximations that is concretised there? Will it bring an unforeseen bug in the code to the fore? Will it prove too sensitive a formulation to provide an apparatus robust enough to explore empirical simulations from a great range of initial conditions?

HU starts with an idealised problem, just some fluid in a 2D domain with a sloping floor. By varying the water level, he can see how the domain behaves as he writes the new wetting and drying module. "At that stage," he says,

> "you are working with your physical intuition. That is all you have. You don't have the ability to validate it against empirical data. You can only compare it against what looks right." (HU)

He is checking visually that the water level gets thin against the surface when the water level drops, to try to see if the code he has written is doing what he thinks it is doing. This is a check that his code embodies what he thinks it does, and that this has the effect that he expects it to. There is at this point no strong assurance of either of these, only an indication that it is time to proceed onwards and try some new things.

> "Once you have something more or less physical you can validate against empirical measurements. But even measurements are not that accurate. The best thing you can have is an analytical solution." (HU)

While this next step can be an attempt to reproduce experimental data from fluid dynamics studies, the problem is that experimental set-up is itself inexact and error prone. Furthermore, even if laboratory scale experiments are often quite simple, they are still a huge step up in complexity from his 2D domain. The next move, therefore, was to use "Thacker's test case", an idealised problem in which the fluid dynamics equations are simplified, and the domain idealised into a very specific 2D parabolic frictionless cup. It turns out that if you have a fluid in the cup with its surface raised up in another carefully worked out parabolic curve, and then let it flow under the force of gravity, the surface height will follow an analytically solvable path. For this highly idealised case, therefore, if HU can set it up as a simulation of Thacker's case that relies on his wetting and drying technique, he has a set of exact answers against which the effects of his method can be compared. He can then vary the resolution of his model to see how the error changes, and meets with success if the error will tend towards zero as he runs higher and higher resolution simulations. At this stage, far from directly investigating the Severn Estuary, as if it was already given prior to investigation, HU is engaged in gradually realising a mathematical system in a computational medium, and the most important aspect of this system, in relation to the equipment already existing in Fluidity's toolbox, is the effect and behaviour of wetting and drying processes.

Matching the analytic results of the Thacker case provides a further indication that the method is working, and that it will continue to work in more elaborate cases, but this point is a long way from making any direct claims. What it does provide is feedback about the properties of the simulation, which HU will tweak to see the effect on these results. As the work continues, HU and his supervisors begin to understand the interaction of the new computational techniques with the type of mathematical system that he is investigating, in the process of which the very idea of what this system is will also be developing. After the analytic solution, he will look to experimental test cases, to try to reproduce results in non-idealised domains, with different geometries and different properties. 3D domains with irregular topographies are next, and all the way it will be possible to tease out of the process more about what the specific effect of wetting and drying will be, alongside the growing knowledge of the specific effect of HU's new technique for implementing wetting and drying in the computer. The eventual modelling of the Severn Estuary must be understood, therefore, in relation to this cumulative trajectory, a trajectory which does not simply build the tools that will be needed, but that is formative of the very object being studied.

We will return to further examples with more detail in the chapters that follow. For now, it is most important to note that not only does Bachelard's philosophy encourage us to look at actual scientific practice, it also indicates that we should be attentive to the dynamics of research processes, the "real time" of practice and its transformative potential. Furthermore, Bachelard points to the differences in orientation that can be found in practice, differences that cannot be subsumed under a single account. He asks, "[d]o you think that, in all his thoughts, the scientist is a realist? Is he a realist when he supposes, is he a realist when he sums up, is he a realist when he schematizes, is he a realist when he makes mistakes? Is he necessarily a realist when he affirms?"[223] In our case, we can point to the very different kinds of orientation found in applications and methods research, as well as the positions encountered in the trajectories of modelling, moving from the mathematical conceptualisation of the new technique, to its implementation in a computationally discretised version of an idealised domain in which mathematics gives a perfect solution, to his modelling of an experimentally artificial domain with its empirical idiosyncrasies, to the modelling of semi-analytic solutions, where a mathematical function is fitted to empirical data, to the modelling of the Severn estuary itself.

When HU models the estuary itself he controls a multiplicity of parameters and variables, from the resolution of the domain, to the rules about when and how much the resolution of the domain will adapt to a new form, to the boundary conditions that control the rising of the tide and the rate of flow in different parts of the channel. As his work moves towards the "real world" case, his own intervention becomes more pervasive, as the control he needs to apply to a more complex simulation grows, even while the progression from the idealisation to experimental artificiality to the measurement of an estuary appears at each step to decrease the level of artifice. The effect of empirical representation gained in this end case conceals under its surface the entire history of its trajectory, carried with it and through which it took shape and turned out the way it did.

4.7 Phenomenotechnique in context

Phenomenotechnique is a way of thinking about the intrinsic rationality of technical practice. We can put this in the context of wider trends in thinking about technology and cognition that became

223 Bachelard, *The Philosophy of No: A Philosophy of the New Scientific Mind*, 35.

prominent later in the Twentieth Century. Phenomenotechnique can be set alongside Bernard Stiegler's philosophy of technical prostheses and Edwin Hutchin's theory of cognition in the wild, both of which situate reason within the world and within our apparatuses of intervention[224]. These kinds of ideas, that “scientific cognition largely relies on non linguistic representations, mental or public, such as images, graphs or simulative models”, are now prevalent in many fields[225]. Reason is a question of things as much as, or more than, it is a question of discourse. Nersessian's cognitive studies of “model-based reasoning” draw these theories further into articulation with the ethnographic tradition in science studies[226].

Looking back even further, Bachelard's philosophy follows a general trend of Twentieth Century thought in his de-subjectification of Kant's categories. Durkheim argued for the validity of sociological arguments (as opposed to psychological explanations) by appealing to the social nature of the Kantian categories. While for Bachelard the non-Newtonian developments in science precipitated his departure from the immutable transcendental, for Durkheim it was the growing evidence from various parts of the world that people from different societies seemed to think through different categories. Concepts, he would posit, are not inherent properties of universal subjectivity, but are of social origin. “Thinking by concepts”, he wrote, “is not merely seeing reality on its most general side, but it is projecting a light upon the sensation which illuminates it, penetrates it and transforms it. Conceiving something is both learning its essential elements better and also locating it in its place; for each civilization has its organized system of concepts which characterizes it”[227]. In his later work, Edmund Husserl embraced a similar notion of the historical and social embeddedness of categories. “Every people, large or small, has its world in which, for that people, everything fits well together, whether in mythical-magical or in European-rational terms, and in which everything can be explained perfectly. Every people has its “logic” and, accordingly, if this logic is explicated in propositions, “its” a priori”[228].

The greatest shift in post-Durkheimian social science has been the integration of this transcendent structure with a conception of embodied practice, something pioneered by Durkheim's nephew Marcel Mauss, and brought closer to maturity by Bourdieu[229]. This brings the transcendental back “down to earth”, within the structures of practice that govern life, and the generative embodied dispositions, “principles of the generation and structuring of practices and representations which can be objectively “regulated” and “regular” without in any way being the product of obedience to rules...”[230]. As Tiles has noted, Bachelard's thought is so difficult to reconcile with analytic

224 Stiegler, *Technics and Time*; Hutchins, *Cognition in the Wild*.

225 Heintz, ‘Why There Should Be a Cognitive Anthropology of Science’, 405; see also Lenoir, ‘Epistemology Historicized’, xvi.

226 Nersessian, ‘How Do Engineering Scientists Think?’.

227 Durkheim, *The Elementary Forms of the Religious Life*, 435.

228 Husserl, *The Crisis of European Sciences*, 373.

229 Mauss, ‘Les Techniques Du Corps’; Bourdieu, *Outline of a Theory of Practice*; Cf. Ortner, ‘Theory in Anthropology Since the Sixties’.

230 Bourdieu, *Outline of a Theory of Practice*, 72.

philosophy of science precisely because his emphasis on actual scientific practice is at odds with a Fregean rationality always abstracted from any particular implementation *in situ*[231]. For Bachelard, as for Bourdieu, it is realisation in practice that counts, not whatever structures that can be abstracted out from those goings on. Emphasis on fixed structures is displaced in favour of emphasis on the sites of their generation and mutation.

As we saw in the previous chapter, much recent social theory has complemented this picture, and brought a new dimension to any contemporary appreciation of phenomenotechnique with deeper engagements with the role of materiality in practice[232]. Looking at phenomenotechnique in computational science means as much understanding the materiality of systems of investigation as it does studying embodiment and the concretisation of concepts. A common core emerges, from which the categories are de-subjectivised, de-absolutised, and opened up. Rationalism is no longer about the structuration of the subject prior to the approach to the world. It is about the sedimentation of intricately textured practical engagements in a world. A science studies that embraces this retains one of the most interesting and productive aspects of the critical heritage.

4.8 Temporality and reason

This last section responds to a final criticism of the Bachelardian position. Bowker and Latour make the accusation that Bachelard's philosophy breaks Bloor's symmetry principle, which he laid down as a fundamental axiom of the Strong Programme[233]. This principle states that sociological explanations should be of the same kind whether the scientists end up making true or false choices: “Having chosen the true option is no less problematic than having chosen the false one”[234]. Bowker and Latour are referring to Bachelard's philosophy of error, of judging the past as error, and this is indeed a far remove from the social explanations of how scientists make their choices, that were favoured by the Edinburgh School. But Bachelard's philosophy is symmetrical in a more important sense, for it regards all present knowledge as the future's error. This is a certain kind of asymmetry here, but it is not the kind of asymmetry of which Bloor was critical. Bachelardian asymmetry reflects the temporality of practice, for which “truth is nothing other than a historical corrective to a persistent error”[235].

While Latour may well disagree with the practice theoretical perspective, and thus reject the fundamental role for temporality that it implies, he cannot rightly portray Bachelard's lack of symmetry as a symptom of a cheap form of argument that attributes causal power to a decision's truth or falsity. Every scientific decision is true in its relation to the past, and false in its relation to the future: “we must put scientific culture on the alert so that it is always ready to move, we must

231 Tiles, *Bachelard*, 25.

232 Ingold, ‘Bringing Things to Life’; Pickering, *The Mangle of Practice*, 6; Miller, *Materiality*.

233 Bowker and Latour, ‘A Booming Discipline Short of Discipline’.

234 Bloor, *Knowledge and Social Imagery*, 177.

235 Bachelard, *The New Scientific Spirit*, 172.

replace closed, static knowledge, with knowledge that is open and dynamic, and dialectise all experimental variables. Reason must in short be given reasons for developing”[236]. This temporality is that of the differential articulation of practice and accounts of it, and it always implies that the past reflections of such a process be viewed in the present light of error.

Despite the potential antagonism between practice and semiotic approaches, a potential reconciliation is suggested by Stengers, where she integrates the Latourian concept of the “factish” with precisely this kind of Bachelardian theme. She writes of moments of transformation in practice, of “the moment when a perspective comes into existence in which what is mixed together can be separated. A new immaculate state of things separates out from its history, becoming, like any factish, capable of explicating the missteps of an outdated past in which it had “not yet” been taken into account”[237]. Latour's rejection of French epistemology, once a considerable barrier to reading Bachelard in science studies, can thus be displaced, opening a new door to reflecting on its rationalist heritage.

4.9 Conclusion

Phenomenotechnique provides us with a basis for understanding scientific practice as something special, as a deployment of reason in skilled settings, in the articulation of scientific work against its own unfolding, its creation of phenomena and setting them into motion. I am interested in the integral dynamics of practice, as a real locus of transformation of the conditions of possibility for further work. But this dynamic is not something to be measured according to some external metric, some scale of advance mapped out through history. It is the temporality of research work itself, which can only be understood in terms of its own logics.

It is hard to overemphasise this point. The term “historical epistemology” itself may be misleading, for it suggests an external viewpoint on practice, charting larger trends. In contrast, this study charts a very small period of time, a period of time containing no huge shifts in the foundation of computational science. But the advantage of this quieter field site is that it allows for greater attention to the kind of work that will never be subject to grand histories of ideas or of revolutions. Reason unfolds, for us as for Bachelard, not in History, as it did for Hegel, but rather in the scientific everyday.

236 Bachelard, *The Formation of the Scientific Mind*, 29.
237 Stengers, *Cosmopolitics I*, 188.

5 Modelling and Representing

5.1 Introduction

We now turn to representation. In a sense, this chapter is the hinge between the theoretical and the empirical parts of the thesis. It is a theoretical coda flagging up the importance of practice for reformulating the concept of representation. But on the other hand, this reformulation turns representation into something a lot more everyday, something to be explored empirically, and thus we start to explore the phenomenotechnical practice of computational physics as a process of modelling, picking up from the example given in the previous chapter, and paving the way for the following chapter on images, in which research in the making takes centre-stage.

One of the themes that has come out of my research has been the claim that modelling is creative. However, the dominance in Bachelard's thought of the idea of the production of phenomena ("phenomenotechnique *extends* phenomenology") makes practices of modelling seem too conservative to count as properly phenomenotechnical. It is too easy to assume that modelling invents nothing new. But I will endeavour to show that practices of modelling are just as productive and creative as are particle accelerators. As Petersen put it, "phenomena are "created", albeit digitally, within the simulation laboratory"[238]. The initial move in showing that modelling is a creative practice with its own integral dynamics is to show that it is in not a matter of creating singular representations of phenomena of interest. In this sense this chapter is removing a key obstacle to our understanding, showing that modelling is dynamic and multiple, that it is creative in its own becoming, and must not be hidden behind the singularity of the products that it sometimes leaves behind.

Any theory of representation offered here must also reconcile itself with the current of post-positivist critiques, which have decisively undermined representation*alism*, that is, any theory of science in which the overall goal and effect of research is conceived in terms of the production of an accurate representation of the world. Tarja Knuuttila has expressed this in her call for a "non-

238 Petersen, *Simulating Nature*, 18.

representationalist theory of representation”[239]. She takes issue with views that consider representation to be the *a priori* central feature of science, and to this extent she sides with anti-representationalist critics, such as Richard Rorty[240]. For too many philosophers, the idea that science represents nature has formed a starting point for inquiry rather than a premise to be critically examined.

The critique of the spectator theory of knowledge has long been a centre-piece of pragmatist philosophy (not just Rorty, but Dewey, Peirce, and others too). But over the last three decades it has been combined with a general turn towards scientific practice. Ian Hacking, for example, aims in *Representing and Intervening* to shift the terms of the realism debate from positions that largely consider realism a matter of the realistic interpretation of statements (particularly where those statements concern unobservable entities), towards a more experimental realism based on manipulation[241]. It is important to note that Hacking's argument, like Knuuttila's, does not imply that we should abandon representation as a concept. In Pickering's terms the key move is a “rebalancing”, in which materiality and practice are allowed their proper space, rather than an outright rejection of representation in general[242]. The point is to de-centre it, to reinsert it within a more practical theory based on intervention and manipulation. Representation is not the only thing going on in science, nor is it the sole criterion for scientific success. This chapter charts the many ways in which representations are deployed, and the many ways they can succeed or fail.

5.2 Theories of theory: syntax/semantics

Recent years have seen a huge surge in interest in models and representation. The usual narrative relates this trend to a longer-term shift in the philosophy of science from “syntactical” views of scientific theory to “semantic” views[243]. One of the major pioneers of this movement was Patrick Suppes, who proposed that there may be advantages to regarding theory in terms of models, rather than in terms of statements. This grew into a general movement contesting the “Received View” that a scientific theory should be thought of as a linguistic entity[244]. Since the days of the logical positivists, the received view had led many to theorise about scientific theories in terms of the logical relationships inside and among these various collections of statements, about their consistency, syntax and axioms, and about the reducibility of one or another collection of statements to observation statements, or to statements in the language of more “fundamental” science, mathematical physics.

239 Knuuttila, ‘Models as Epistemic Artefacts’.
240 Rorty, *Philosophy and the Mirror of Nature*.
241 Hacking, *Representing and Intervening*.
242 Pickering, *The Mangle of Practice*, 7.
243 See, for example, Suppes, ‘The Meaning and Uses of Models’; Giere, *Explaining Science*; van Fraassen, *The Scientific Image*.
244 Suppe, ‘The Search for Philosophic Understanding of Scientific Theories’.

By the 1960s, however, a number of problems with this kind of view had accumulated. Quine had raised important questions about the distinction within these languages between theoretical and observational statements, about translation between theories, and about holism[245]. Kuhn, on the other hand, had further pushed the problem of the relationship between theories by challenging even the idea that a science is compatible with past forms from its own history[246]. In this context, Suppes and colleagues made the move from thinking about theories as linguistic entities to thinking about them as structures through the framework of mathematical model theory. He thus elided many of the problems with the syntactic view. If a scientific theory is considered through model theory as a set theoretical structure, many different kinds of formulations would be permitted of the same theory. This provides a much more abstract notion of what a scientific theory is, because it can no longer be identified with a set of statements found in textbooks or publications. But it presents an exit route from the problems that were mounting up through the philosophy of language[247].

While the semantic view remained highly abstract and formal[248], the general space of contestation of the received view led other philosophers to respond by looking instead at the much more concrete processes involved in the application of theories. The more abstract the view of theories, the greater impetus is generated to question how theories actually get applied to concrete circumstances. The most important writer in this trend was Nancy Cartwright, who famously argued that we are mistaken to think that scientific theory in itself provides a true description of the world, showing that in order for any specific empirical claims to be made it is necessary to go through an intermediary process of modelling, and only through such a process could theoretical "laws" be meaningfully brought into articulation with the world[249]. Her work stimulated a great deal of interest in the processes of applying theories, as something interesting in itself. Along with others, such as Giere, who combined aspects of the semantic view with a more concrete approach to models, Cartwright brought processes of modelling to the centre of attention in philosophy of science.

The simulations made at AMCG are constructed using resources from several different bodies of theoretical literature (branches of mathematics, computer science, physics, oceanography, geology, and others). There is no reason in this context to assume that such different principles can be supposed to form parts of a unified whole. Indeed, as we have seen, studies of scientific practice often stress the benefits of drawing on disunified theories, triangulating practice with resources from very different disciplinary origins[250]. Furthermore, non-scientific principles drawn from the worlds of commercial and open source software development are very important in guiding how the simulation gets built. Finally, it is important not to underestimate the role within scientific software

245 Quine, *Word and Object*; Quine, 'Two Dogmas of Empiricism'.

246 Kuhn, *The Structure of Scientific Revolutions*.

247 For an overview of the problems introduced into the philosophy of science through the philosophy of language, see Zammito, *A Nice Derangement of Epistemes*.

248 French and Ladyman, 'Reinflating the Semantic Approach'.

249 Cartwright, *How the Laws of Physics Lie*.

250 see, for example, Hacking, *Representing and Intervening*, 183; Galison, *Image and Logic*, 781.

development of numerous practical competencies that have been inculcated and transformed over many years of such work, and never formalised or written down, but which exert strong guiding influence on what gets done and how. So when I claim that we should think about simulations in terms of processes of modelling, I am going further than Cartwright. This kind of research can be thought of in terms of the application of theories, but it also has significant autonomy from theoretical practices[251]. Phenomenotechnique does not take objects or problematics ready made, but transforms them within its own domain, within which theories comprise just one kind of resource for modelling among many.

The other major consequence of the semantic turn is that it provoked a heightened interest in representation, even with the effect of displacing propositional conceptions of truth. “Unlike propositions and sentences, terms such as ‘‘true’’ and ‘‘false’’ did not seem suited to dealing with the relationship between models and their target systems. ‘‘Representation’’ seemed to be more appropriate—and flexible”[252]. Representation may appear to raise the question of truth when it is the representational role of statements about the world that is at stake. But when we take a broader view, and look at idealised models, or at physical models, it becomes very hard to capture what is at work within representation with a concept of truth. Many models are effective as models precisely because they are in some respects “false”, because they idealise parts of it (frictionless planes and point masses in physics, for example) or because they physically embody the target in a manner that introduces useful inaccuracies (for example the size of an architect's model of a building allows it to be moved, displayed, and easily examined from many angles)[253].

The concept of representation appears to provide the flexibility needed to capture what is at stake among the diversity of models in science. It also has the advantage of being an ordinary idiom of talk about models. Scientists at AMCG will at times talk in terms of representation, but I have never heard any of them call their model “true”. Truth is a much more natural idiom for talk of judging linguistic claims, and is awkward when brought into the diverse world of models. Shedding the connection between truth and representation is probably the most important move in ridding us of representation*alism*.

5.3 Substantive and deflationary theories of representation

The major dividing line within contemporary theories of representation is between formal and practical theories. Formal approaches tend to consider representation to be a two-term relation between a source and a target. Suarez identifies two main versions, one which understands this relationship as isomorphism, and the other regarding it in terms of similarity[254]. Roughly speaking, Bas van Fraassen is the major representative the isomorphism view in *The Scientific Image*, a key

251 Morrison, ‘Models as Autonomous Agents’.

252 Knuuttila, ‘Modelling and Representing’, 262.

253 Knuuttila, ‘Models, Representation, and Mediation’, 1263.

254 Suárez, ‘Scientific Representation’, 95.

text in the rise of the semantic view, in which he argues for a constructive empiricism[255]. da Costa and French have more recently defended a related view in which representation is to be understood as “partial isomorphism”[256]. The “similarity” conception is best known through Giere's early work[257], though it must be noted that, partly in response to criticisms from philosophers such as Suarez, he has more recently modified his view to make it more in line with the practical enquiry[258].

All these formal approaches to representation share a commitment to the idea that representation is a certain kind of thing, identifiable according to its essence (e.g. isomorphism, similarity, etc.). In contrast, practical theories of representation tend to assert that there is an irreducible contextual element necessary for any adequate theory of representation. For those who hold such views, representation simply cannot be understood if all we pay attention to is a relationship between a source and a target. We must also take into account *something else*, which may be use, purpose, intention, or a more vague conception of the *context* of representation. In each case, representation exists not because of something intrinsic in the relation between a source and a target, but because these terms are situated in a circumstance where they are being put to use as a representation relation. Depending on which version you adopt, this could be because someone is intending for the source to represent the target[259], because the source is being used to represent the target, or simply because the source conventionally represents the target. Giere proposed a four-term theory of representation[260], in which alongside source and target both the agent and his/her purposes have a role, but what is common to all these perspectives is that they regard representation as *at least a three term problem*.

At this point I don't want to commit to any particular version of the practical theory of representation, but I would note that the force that practical positions have today owes a lot to Mauricio Suarez's critique of the major two-term formal theories. Quite aside from his own particular take on the form that the practical theory should take, Suarez launched a wide-ranging attack on the two major dimensions to formal theories: they tend to be reductive and/or substantive in their treatments of representation[261]. They tend to explain representation by reference to something more elementary (reduction) and do so by positing that representation has an essence (substance).

Suarez argues that representation is not the kind of thing that has an essence. And it cannot be reduced to something more basic (such as, for example, isomorphism). He gives several reasons for this. For starters, no reductive theory seems to be able to account for the empirical diversity of things that we are going to want to call representations – in our example, even if some key examples of

255 Van Fraassen, *The Scientific Image*.
256 Da Costa and French, *Science and Partial Truth*.
257 Giere, *Explaining Science*.
258 Giere, ‘How Models Are Used to Represent Reality’.
259 Bailer-Jones, ‘When Scientific Models Represent’.
260 Giere, ‘How Models Are Used to Represent Reality’.
261 See for example Suárez, ‘Scientific Representation’.

representation appear to play on isomorphism in order to do their work, it is difficult to extend this across the many other kinds of representation. Furthermore, what Suarez calls the “logical argument” notes that three features of the things we call representations (non-reflexivity, non-symmetry, and non-transitivity) are usually broken by accounts based on similarity or isomorphism. In other words, it would appear to be an incorrect logical consequence of formal theories of representation that representations represent themselves (since they would be similar or isomorphic to themselves), that the target, if similar or isomorphic to the source, would thus represent the source as much as the source represented the target, and that if you have a chain of representations (A represents B which represents C), the first and last terms would necessarily represent each other[262].

The other arguments Suarez deploys further undermine reduction theories, by showing their principle candidates for explaining representation to be non-necessary and non-sufficient, and finally pointing out that there is a difficulty from such points of view of dealing with “mis-representation” - mistakes in correctly identifying the target from the source. What works in one circumstance as a successful representation, may later be reassessed as a mistake. This brings Suarez to assert that it is necessary to abandon “the aim of a substantive theory to seek universal necessary and sufficient conditions that are met in each and every concrete real instance of scientific representation. Representation,” he says,

> “is not the kind of notion that requires, or admits, such conditions. We can at best aim to describe its most general features—finding necessary conditions will certainly be good enough. Second, it entails seeking no deeper features to representation other than its surface features...[263]”

Suarez does go on to suggest two candidates for these necessary but not sufficient conditions, and thus espouses a particular version of a “deflationary” view of representation: “the representational force of a source is one such irreducible feature; the capacity to allow surrogate reasoning is another”[264]. Both of these are in his view “general features” characteristic of circumstances where representation is found in practice. But I leave them on one side here. Both representational force and inferential capacity are tied to a very philosophical preoccupation with making inferences. Instead I start from the more sociological tradition of looking at practice as a concrete milieu, and look for more practice-based “surface features”.

5.4 From vicarship to realisation

Suarez gives us a powerful amalgamation of critiques of formal theories of representation, and points the way towards a more practical enquiry. But there is a basic problem with his approach. Under the surface he relies on an implicit assumption that when we talk about representation we are

262 Suárez, ‘An Inferential Conception of Scientific Representation’, 768.
263 Ibid., 771.
264 Ibid.

interested in a source *distinct* from a target. This is the case whether or not we also find a fundamental role for the context in the equation.

In the viewpoint that I want to put forward here, representation may involve a source distinct from a target, but this distinction is not something to be taken for granted. It exists as a play of ambiguity, a play between identity and difference. The separation of source and target is at the far end of a spectrum, along which much of what we call representation occurs. This far end, this extremity, has as its model a stereotypical type of representation where the two terms are separate. From this stereotype, we gain the basic formulation of the problem of representation: that the thing to be explained is the bridging of the gap between the two terms. For Rheinberger this situation, called "vicarship", can only be understood as one pole of something broader, "a continuum from vicarship to embodiment to realisation"[265]. We thus gain a much broader perspective on representation, one which refuses to take for granted the separation and separability of source and target, and instead locates this separability in a broader sphere of possibilities.

In the middle region of the spectrum, source and target are not straightforwardly distinguished. Rheinberger illustrates "embodiment" with an example from theatre. In contrast to the sense of vicarship in which representation is "representation of", if "we claim that we have seen the actor Bruno Ganz yesterday evening representing Hamlet, we speak of a representation "as""[266]. Similarly, drawing on studies from political and aesthetic theory, Knuuttila makes this discrimination with the terms "re-presentation" (what I would call embodiment) and "standing for" (vicarship). Using Pitkin's analysis of the etymology and history of the term, Knuuttila notes that re-presentation is closer to the original sense of making present, or of "the making present of an abstraction through or in an object, as when a virtue seems embodied in the image of a certain face"[267]. Knuuttila regards the vicarship sense of the term to be a modern invention. "Representation as "standing for" is typically approached through the metaphors of portrait, map or mirror: what they have in common is that they are all renderings of an "original" in a medium different from it"[268].

In Rheinberger's view, we can also discern a third sense of representation, further along the spectrum. "If... a chemist tells us he or she has produced or represented a particular substance in his or her laboratory, the meaning of "representation of" is gone, and instantiation in the sense of the production of a particular substance has taken over. In this latter case, we deal with the realisation of a thing"[269]. Embodiment, in Rheinberger's version of things, is intermediate between vicarship and realisation. Vicarship is one far end of the spectrum; here source and target are entirely separated, while realisation at the other end consists of their complete coincidence, a circumstance of indistinguishability. Much of what is of interest in representation, however, happens in between,

265 Rheinberger, *Epistemic Things*, 103.
266 Ibid.
267 Pitkin, quoted in Knuuttila, 'Models as Epistemic Artefacts', 21.
268 Ibid., 22.
269 Rheinberger, *Epistemic Things*, 103.

through the intermediate ambiguities of embodiment. The actor both is and is not Hamlet. Hamlet is what is realised in the performance, but is also someone else than the actor. Quoting Mitchell's *Iconology*, Rheinberger notes that "[e]very play is governed by this tension, this "paradoxical trick of consciousness, an ability to see something as 'there' and 'not there' at the same time""[270].

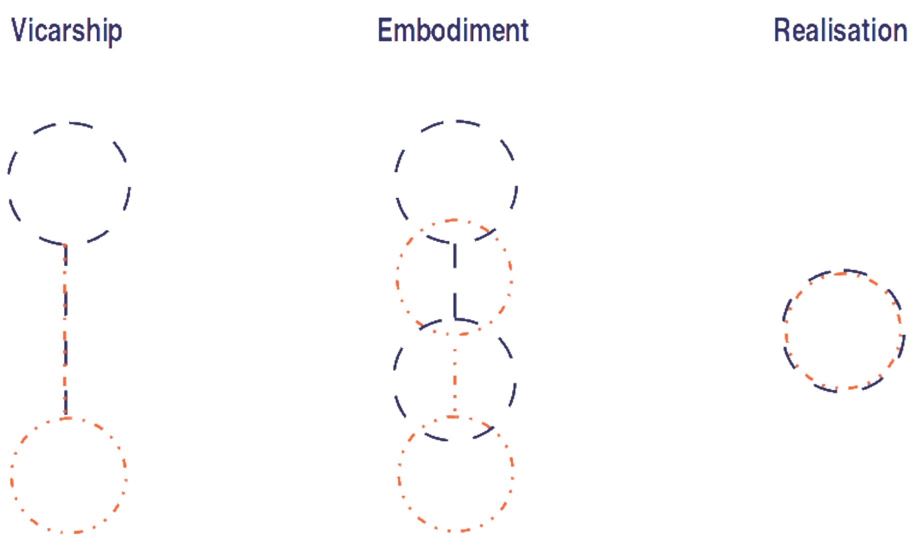


Figure 4: A diagrammatic representation of Rheinberger's spectrum

What figure 4 attempts to put on show is the relatively straightforward situation of vicarship (a binary relationship between two distinct entities) and realisation (coincidence of the two terms in the same entity), when compared with embodiment, which is ambiguous and hard to represent. Embodiment involves the oscillation between coincidence and relationship: the relatedness of terms which exist as separate terms through their being equated; the coincidence of differences, of what is not the same.

If the vicarship-embodiment-realisation continuum is our necessary but not sufficient feature, a theory of representation will have to deal with the full spectrum, and the many ways in which the positions on it move and interact. Mary Morgan provides some excellent examples of these issues in a text where she argues for a spectrum between traditional material experiments and "non-material" experiments such as mathematical models, thought experiments and simulations. Her concerns are largely epistemic, about the manner and grounds of inferences. This aspect I will put to one side, and concentrate instead on the nature of representational relationships that Morgan describes in the course of her argument. In a material experiment, says Morgan, what is produced and manipulated is the same thing as what is being studied. In a non-material experiment, it is a mathematical system

270 Ibid.

that is manipulated, ontologically a very different thing to whatever empirical system is being simulated[271]. We do not need to subscribe to Morgan's view that simulations should be placed at the far end of the vicarship side of the spectrum in order to see that she is dealing with much the same issues as Rheinberger. Having set out these contrasting poles, Morgan shows that many types of scientific practice fall in between, tapping in to the tension between the extremes.

In her example from studies of bone structure, Morgan shows research proceeding through the intermediate zone. In one case, scientists take cow bones and scan them to create a representation on a computer. A computer model will then apply a model of physical forces to test the bones' response to stress and strain. Although the computational medium can only vicariously represent the physics of the bone, to a certain extent the bone structure realised in the bone is transferred to this medium through the scanning techniques[272]. The example resonates with aspects of work in oceanography, because while digital processes are very different from the fluid dynamics they represent, the shape of the domain for the simulation is often generated from satellite scans of coastlines, or ship-borne scans of bathymetry.

Morgan goes on to demonstrate that these intermediary forms expand further when you consider that what is at stake in scientific investigation is often the relationship between individuals and classes. A lab mouse, in Morgan's vocabulary, is a “representative of” mice[273]. It is a mouse. The class as a whole exists insofar as it is realised in individuals such as this one, but on the other hand any one individual (especially a selectively bred lab mouse) has its own empirical peculiarities and thus can only vicariously stand for the species as a whole[274]. Morgan will then push this further to another relationship, in which the mouse is a “representative for” rodents in general, mammals in general, vertebrates, and so on. Much medical research is carried out through such relationships, in which it is posited that in some significant way mouse physiology stands for human physiology. Common aspects of mammal biology are realised in a mouse like this, which may thus be used as a representative of a wider class of organisms[275].

I don't think it is necessary to adopt Morgan's terminology. Whether something is a “representative of” or a “representative for” depends on your view of species and kinds, and this need not be an issue here. What is important is that she flags up the fact that many relations of representation cannot simply be collapsed into the vicarship pole. On the contrary, they “represent the world via mixed modes of representation and representiveness”[276]. Not only does representational practice occur across a spectrum of modes, it also often involves assembling together several different types of representation at the same time.

271 Morgan, ‘Experiments Without Material Intervention’, 218–220.

272 Ibid., 221–226.

273 Ibid., 228.

274 For a discussion of the history of norms of representing classes and individuals see Daston and Galison, *Objectivity*.

275 Callon notes how such intermediate ‘representivity’ can easily become precarious over the course of a project. Callon, ‘Some Elements of a Sociology of Translation’, 214–220.

276 Morgan, ‘Experiments Without Material Intervention’, 233.

Between the extremes of vicarship and realisation is a play of presence and absence. The anti-positivist tradition of critiquing representation, from Heidegger to Foucault and Rorty, is best understood in this context as a critique of pure representation on the far vicarship side of the spectrum. But, recalling Nelson Goodman's contributions, "[u]pon closer inspection," says Rheinberger, "any representation "of" turns out to be always already a representation "as"."[277] Any representation that seems to be a pure vicarship, turns out to play in a much more complex way on its embodiment of the target, and even to some extent on the target's realisation in the source. Pure vicarship is rarely, if ever, attained. This side of things is rarely acknowledged by philosophers such as Suarez, whose analysis, however much triangulated by pragmatic and contextual concerns, however deflationary, nevertheless takes the two terms of the relationship to be relatively unproblematic. Things are never this crisp and clear. The source never fully makes way for the target. And the target is never completely there either. The heart of representation lies in this fundamental ambiguity, which is not an artefact of an insufficient theory of representation, but rather a key mechanism in representation itself.

The combination of a pragmatic deflationism with the kind of poststructuralist-influenced material semiotics that come through in Rheinberger's acccount provides us with a perspective for which representation has no essence, but it does have the necessary condition of being the embodiment of ambiguity in the relation between source and target. To embrace this ambiguity is not to defy Suarez and propose a new theory of unitary essence. There is no claim that all such play of presence and absence amounts to representation. Merely that where there is representation, we are in the space of such play. It is necessary but not sufficient, a "general feature" of representation and in this respect a rival contender to Suarez's two criteria: a directional relationship (force) and a potential epistemic relationship (inference). This play can even be considered anterior to any such criteria, because it is the very condition of difference between the source and target, the very condition for the recognition of a relationship obtaining at all, on top of which it might then be possible to do as Suarez does and identify as its two key properties directionality and inferential capacity.

5.5 What kind of thing might represent?

A deflationary stance displaces substantive questions. It is not appropriate to ask "what is representation?" or "how is representation in general to be explained?". On face value, therefore, deflationism may appear as a form of quietism. But this displacement raises different questions, equally "big". If representation is always irreducibly tied to its specific context, a number of "cultural" questions can be raised.

Representation does not happen in a vacuum, but in specific historical and cultural circumstances. Certain sorts of things are candidates for representing more than others, this candidacy being part of

277 Rheinberger, *Epistemic Things*, 104.

the “social life of things”[278]. Things become the kind of things that open a play of presence and absence in different ways in different contexts. In the Western tradition, photographs are an easily recognisable type of thing, the very recognition of which raises immediately the question of the target represented in the image. In the words of Roland Barthes, it is “as if the Photograph always carries its referent with itself”[279]. But we can imagine a culture with no similar tradition of planar depiction, for which a photograph would not immediately be taken as the kind of thing that would represent, as the kind of thing in which is seen something else, some target. Wittgenstein offered some thoughts along these lines:

> “we *view* the photograph, the picture on our wall, as the very object (the man, landscape, and so on) represented in it... This need not have been so. We could easily imagine people who did not have this attitude to such pictures. Who, for example, would be repelled by photographs, because a face without colour, and even perhaps a face reduced in scale, struck them as inhuman.”[280]

These considerations invite a broad historical view of how certain things become candidates for representing, analogous with the “positivities” discussed by Hacking, a term he takes from Foucault[281]. Hacking unites his reading of Foucault with the concept of “styles of reasoning”, drawn from Crombie[282]. He points out that philosophical discussions (of translation, conceptual schemes, incommensurability, etc.) have tended to regard all statements to be candidates for truth or falsity, and proposes instead that these things are much better appreciated from the point of view of a history of the styles of reasoning that determine what kind of things are candidates for being true or false, and how they might be true or false. Similarly, the historical tradition provides us with a landscape of things that in many different ways are the kind of thing that might represent. The kinds of games we play in setting up representations are extremely specific. A pencil sharpener may represent a football player in an office explanation of the off-side rule. A pillow may become a mountain in a child's game. In both cases, there are certain ways to participate in the scenario. Several philosophers have made the link between these imaginative games and scientists' work in modelling[283].

In Frigg's terminology, drawn from Walton's pretence theory, there are in each case certain “rules of generation”, which “are public and shared by the relevant community”, and which define how things can be drawn into games of representation[284]. He points to the need for philosophers to investigate the sets of rules at work in science, stressing disunity, in that “different disciplines have different rules, and understanding what these rules are will shed light on how modelling in these disciplines

278 Appadurai, *The Social Life of Things*.

279 Quoted in Lynch, ‘Science in the Age of Mechanical Reproduction’, 214.

280 Wittgenstein, *Philosophical Investigations*, 216e.

281 Hacking, ‘Language, Truth and Reason’; Foucault, *The Order of Things (Routledge Classics)*.

282 See also Davidson, ‘Styles of Reasoning, Conceptual History, and the Emergence of Psychiatry’; and Morgan, *The World in the Model*, 13–16 for an application of Crombie’s ideas to economics.

283 See, for example, Toon, ‘The Ontology of Theoretical Modelling’; Frigg, ‘Models and Fiction’.

284 Frigg, ‘Models and Fiction’, 264.

works”[285]. Such rules do not exist in any straightforward manner (see chapter 3), but the possibility of their objectification points to the key importance of historical sedimentations in practices of representing. Over the lifetime of a discipline we would expect such conditions to change, opening new spaces for interplay of elements, spaces that in old and new ways “disrupt the immediacy of presence of a phenomenon by rendering it as a mark”[286], creating new spaces in which representing can occur.

The question of the novelty of computer simulation in science provides a good example. It is common for computational scientists to regard simulation as a new and separate endeavour to experimentation or theorisation[287]. But philosophers argue over whether there is any ontological or epistemological ground for this claim[288]. From the point of view of the broad historical terrain in which different kinds of things become candidates for representation, the issue becomes clearer. Evelyn Fox Keller, for example, points out that when the first computer simulations were being run, they were really not much different from mathematical models[289]. Pioneers of simulation such as Stanislaw Ulam and John Von Neumann had previously been doing similar kinds of calculations by hand, and the computer provided a way to do the same thing mechanically. The entry of simulation into the scientists' repertoire of techniques was smooth. One thing that the growing literature on models and modelling has taught us is that models have long been a central part of scientific work. Early computer simulations were recognisable without much difficulty as the same kind of thing as the numerical systems that scientists were already familiar with. Only minimal effort was required to introduce simulation into the scientific repertoire because it was already recognisable.

This does not, however, mean that the debate over the novelty of simulations is resolved. We can still suppose that if there is a crucial difference, it is to be understood as something that only emerged later. This is Keller's contention, “that what we have now come to see as the epistemological novelty of computer simulation in fact emerged only gradually – not as a consequence of the introduction of any single technique, but as the cumulative effect of an ever-expanding and conspicuously malleable new technology...”[290]. Having been established within the context of a wide historical precedent of modelling in science, simulation's particular constellation of techniques went on to profoundly modify the manner in which this kind of representation could represent. Keller argues that simulation grew away from mathematical modelling by the mid to late 1950s, when a properly experimental dimension emerged as a core part of its practice[291]. More recently, the kinds of simulations typified by artificial life and cellular automata, are in Keller's view, a full departure from

285 Ibid., 267.

286 Rheinberger, *Epistemic Things*, 105.

287 See, for example, Engineering and Physical Sciences Research Council, ‘International Review of Research Using HPC in the UK’, 1.

288 The ongoing debate is nicely captured between Frigg and Reiss and Humphreys. Frigg and Reiss, ‘The Philosophy of Simulation’; Humphreys, ‘The Philosophical Novelty of Computer Simulation Methods’.

289 Keller, ‘Models, Simulation, and “Computer Experiments”’, 203.

290 Ibid., 201.

291 Ibid., 205.

precedents. “In contrast to conventional modelling practices, it might be described as modelling from above”[292]. De Landa will concur with the novelty of these approaches, considering them instances of a new kind of “synthetic reason”[293]. In such endeavours practices of modelling do not involve any strong explicit determination of what it is that is being represented. This is left largely implicit, embodied in the practices themselves. Many of this research concerns the high-order behaviour of complex systems, understood in the abstract, and which may turn out to be applicable to all manner of different areas of biology, ecology, sociology, economics, etc.

Now, Keller acknowledges that in this departure, there is no small dose of controversy. It is not clear what exactly it is that is being represented in many of these simulations, nor how to validate them. The point is that our conventions concerning what kind of thing is recognisable as a representation, and how it might represent, are profoundly modified by the historical course of scientific practice. “So radical an inversion of conventional understandings of the relation between simulation and reality are not yet widespread – either in the physical or the biological sciences (indeed, they have yet to make any noticeable impact on the majority of biologists) – but the very fact that they have become thinkable, and in certain circles even acceptable, is surely worth noting”[294].

5.6 What makes a good representation?

If the first half of the question is “what kind of thing makes a representation?” the other half of the question concerns the virtues that different kinds of representations may possess. Just as we can locate scientific practice in a broad landscape of what kinds of things get taken as representations, these configurations also set up the conditions under which such representations might be evaluated. This is a quasi-moral dimension, epistemic virtues not of character but of the thing[295]. What makes a representation good?

The answer to this question will clearly involve the purpose to which a representation is put, the projects in which it is entwined. But this cannot be the full answer. There are many ways for representations to be good. It is not simply a question of the evaluation of the virtue of a representation on a linear scale from bad to good, but a question of the specific modes in which it can be good or bad. Utility is one, and we can list at least three others: faithfulness, materiality and relations with other representations.

A useful representation is often a good representation, but it invites the further question of what subset of activities the representation might be useful for. Much of the philosophical literature on representation is deeply invested in the *epistemological* utility of different kinds of representation. Matthew Parker, for example, considers the utility of simulations in cases where “deductive

292 Ibid., 209.

293 De Landa, *Philosophy and Simulation: The Emergence of Synthetic Reason*; de Landa, ‘Virtual Environments’.

294 Keller, ‘Models, Simulation, and “Computer Experiments”’, 212.

295 What has become known as ‘virtue epistemology’ tends to focus on character, but we could extend this towards a concept of virtues of/in practice. Cf. Daston and Galison, *Objectivity*.

validation" is impossible. "Where problems are unsolvable, intractable, or just plain hard, we do our best, and computer simulations help a great deal"[296]. The virtue of computer simulations, from this epistemic point of view, is tied to the specifics of the problem. In cases where no exact solution is available, simulations can provide a good informal approximation. Similarly, the other classic case is where experiments are impractical (this was discussed in chapter 2).

We might, however, want to be wary of confining the assessment of utility to the role that a representation plays in making arguments. Many, in fact most, of the simulations run at AMCG are not used to make arguments, but are provisional simulations run in the course of the development of the model. The aim of these may in part be to help ground the epistemic legitimacy of claims eventually made based on future iterations of the code, but in large part they are judged good or bad according to whether they enable the scientists involved to move on to a further stage of model development. They may, for example, be prototypes or proofs of concepts forming the basis for a decision about what features of the simulation to try adding next, or they may help ferret out the cause of a pernicious bug in the code. They may aid the unfolding of the investigation by operating as heuristic tools, cognitive scaffolds for gaining a new perspective on a problem blocking onward travel. Conversely, representations' intricacies may enchant and detain attention, constituting a blockage to the path of the project[297]. The exploration of these issues is an important part of chapter 6.

Representations are often judged according to the perceived faithfulness between source and target. A painting may express what the artist feels needs to be expressed there. It may also show a "good likeness". A simulation may provide a good approximation of the relevant equations in physics. No doubt it is from these virtues that we gain intuitions of correspondence, isomorphism and similarity, intuitions that deeply influence the way that we imagine representation in general. But as Giere will point out, it can only be "*relevant* similarity" that is at stake[298]. In many cases faithfulness is abstracted from any actual isomorphism because the point of the representation is to faithfully represent only a specific sub-set of the features of the target. It needs to be false in a number of respects if it is to be a good representation in others. If a meteorological simulation to predict the weather of next week strove for the greatest faithfulness it would finish its computation many months afterwards[299]! A representation's faithfulness is often to some abstraction of the target, as, for example, when the architect's model is faithful to an idea of the form of the building, conventionally abstracted from its size, but preserving its proportions. What kinds of accuracies and inaccuracies are desirable will depend on the representational tradition and on the particular trajectories of practice involved.

296 Parker, 'Computing the Uncomputable', 462.
297 Gell, 'The Technology of Enchantment'.
298 Giere, 'How Models Are Used to Represent Reality'.
299 Edwards, *A Vast Machine*, 122.

The materiality of a representation is something often neglected in discussions of modelling, in many cases because it is presumed that mathematical and ideal models are immaterial. There has been much debate over what role materiality has to play in simulations, because it is easy to regard them as abstract when they are compared to physical models[300]. One of the major points argued in chapters 7 and 8 is that computational modelling is just as material as anything else. Knuuttila argues that materiality is a key aspect even in the most ideal of cases. “All objects of human culture have both ideal (or virtual, if you like) and material dimensions – even totally fictional ones that are nevertheless materialized as texts and pictures concerning them”[301]. She points us towards the significance of the material “constraints and affordances”, which affect what potentialities are embodied in a particular artefact[302]. Different kinds of representations can be manipulated in entirely different ways, opening up different kinds of possibilities for thought and practice. What and how embodiment may play out in a representation will have a lot to do with its materiality, and given that there is a great deal of scope for unforeseen epistemic opportunities to emerge during the course of the research process, the materials being worked with condition science at a fundamental level.

Other significant material dimensions of the representation are portability and durability. A representation may be good because it can be moved into different contexts, a point emphasised by Latour when he wrote that “the *logistics* of immutable mobiles is what we have to admire and study, not the seemingly miraculous supplement of force gained by scientists thinking hard in their offices”[303]. The logistics of things depends on which ways they are hard to move, easily moved, transitory or durable. Rheinberger echoes this point when he writes that “[i]nscriptions are... not mere abstractions. They are durable and mobile purifications, which in turn are able to retroact on other graphematic articulations – and, what is most important, not only on those from which they have originated”[304]. It is not hard to imagine that part of the power of written texts derives from their easy mobility and their possibilities of being reproduced in many places with relatively little effort.

The medium of simulation has also been afforded new openings by the development of the internet, which allows for the efficient transmission of data and code, on a level that dwarfs what is possible for most other kinds of scientific apparatus. Whole codes can be transferred across the globe, duplicated and shared. On the other hand the medium has its own limitations, confronted on a daily basis by scientists who push the limits of what the technology is able to do. Vast data sets, for example, may be too big to download in a reasonable timetrame from a supercomputer, and too big to store in multiple iterations. As much as computational science involves a free flow of data, it also constitutes a vast machine for its disposal and replacement by an endless tide of further productions[305]. One of the greatest dangers for outsiders misunderstanding the nature of

300 For example Guala, ‘Models, Simulations, and Experiments’; Parker, ‘Does Matter Really Matter?’.

301 Knuuttila, ‘Models, Representation, and Mediation’, 1267.

302 Ibid.

303 Latour, *Science in Action*, 237.

304 Rheinberger, *Epistemic Things*, 106.

305 Edwards, *A Vast Machine*.

computational science is to succumb to the postmodern image of computation as free flow and super speed[306]. Once you work at the limits of computation, you are constantly battling with the sluggishness of machines, the finite capacity of storage devices, and the limited bandwidth of systems of transmission, a messy and tactile world of glitches and capacities. This matters for how good computational representations are for working with, for how they get used, how they get made, and what happens in research carried out with them.

A final manner in which a representation may vary in its candidacy for being good or bad is in its relationship with other representations. In many cases, a representation is good only if it is part of a series. Simulations are good when they are part of a system of investigation, a series of simulations that builds the validity of the model, and that explore its behaviour under different conditions. On the other hand, representations may find other kinds of merit in breaking from a series, representing something, for example, in a distinctive and novel way.

The aim of this discussion has been to show that although a deflationary position may appear not to be able to say much about representation, its silence on the universal level does not amount to silence on the level of more in-depth engagements. There is a lot to be said about how and why particular traditions of representing take the form that they do, how they interact with other ways of representing, and how they change over time, as well as their multifaceted manners of being good or bad, achieving success or failure, and being deemed worthwhile and worthless.

These criteria provide us with a provisional set of tools for evaluating the context of representation, and the next section aims to put them into practice. To do so, however, it is necessary to move from our consideration of representation towards a discussion of processes of modelling. This helps us avoid presuming that representations occur singularly and/or successfully, and instead focusses attention on the practices in which they appear.

5.7 Modelling as process

One of the decisive contributions to the philosophy of models has been Margaret Morrison's theory of models as “autonomous agents”[307]. She argues that in many cases models involve extra-theoretical considerations, that theory often does not provide the means by which to develop an appropriate model of a phenomenon, and in many other cases no sufficient theory exists. Models inhabit an intermediate space between experiment and theory, a “unique and autonomous position... one that is separate from both theory and experiment yet able to intervene in both domains”[308]. This begins to free modelling from presupppositions that it is always only a matter of representing, and is analogous with the claim Hacking made of experiments that they “have a life of their own”[309]. In

306 See, for example Virilio, *The Information Bomb*.

307 Morrison, ‘Models as Autonomous Agents’.

308 Ibid., 64.

309 Hacking, *Representing and Intervening*, xiii.

both cases, the important thing is to open up an equipmental domain as something with intrinsic interest for philosophers, and with its own independent dynamics, rather than regarding experiments or models as simply means to theoretical ends. However, while Morrison pushes us in this direction, and she has been very influential in encouraging a practice-oriented philosophy that pays attention to processes of modelling as interesting in their own right, it is debatable whether she takes us the full way. This is Knuuttila's critique. She points to the fact that because Morrison is primarily interested in epistemological questions of how we learn or make inferences from models, her analysis returns their representational function back to centre stage[310].

A theory of modelling must not be subordinated to epistemological concerns about inferences and arguments[311]. It must foreground the temporality of research practice, research as a *process*[312]. I think Rheinberger puts this well. He says “I am concerned with describing the process of making science as a process in which traces are generated, displaced, and superposed... I contend that if the perspective of a clear-cut dichotomy between theory and reality, between concept and object, is being adopted precociously, this process tends to disappear from sight”[313]. In other words, rather than starting by positing one grand metaphysical relationship between science and the world, we examine the multifarious little mundane relationships by which it is made. What we can expect to see is an explosion of representation, multiplying its forms, and dispersing it throughout each circumstance of research.

Modelling is never started from scratch. It also has no telos. It is, in Pickering's words, “an open-ended process with no determinate destination”[314]. A research project draws on many sources. Each tool has a precedent, each idea a history. A great deal of baggage comprises the circumstances of practice in which research gets carries out. Any simulation, by virtue of being recognisable as a simulation, is already strongly pre-conditioned as a clearly recognisable instance of the kind of thing that represents. The strength of this conditioning is apparent in the fact that even a simulation that has no identifiable target is still regarded as something that represents, albeit with a non-existent or indeterminate target.

Many scientists learn to manipulate simulations by playing around with the software. In the annual training course for Fluidity, participants run example simulations that have been set up for them by the AMCG scientists, and they are then encouraged to make their own modifications to the set-up to create something new. These new simulations may well no longer represent anything specific. The modification may render them useless for their original purpose. But it is not right to simply say that because of this, they are no longer representations. They are representations because part of being a simulation is to participate in that play of presence and absence, in which something else is deemed

310 Knuuttila, ‘Models, Representation, and Mediation’, 1265.

311 Mattingly and Warwick, ‘Projectible Predicates in Analogue and Simulated Systems’ is a good example of the latter.

312 Some philosophers have taken tentative steps in this direction. For example, Lenhard, ‘Computer Simulation’.

313 Rheinberger, *Epistemic Things*, 104.

314 Pickering, *The Mangle of Practice*, 19.

present in the simulation, which, in thus embodying the target, recedes into the background. In these training contexts, however, that “something else”, that target, may never be determined. It may be left blank. With thousands of options and parameters available, and a limitless number of possible simulations to be created, the overwhelming majority have no target easily specified.

Training, it might be thought, is a special case. It is yet another case of that awkward fact that most scientists are also by virtue of their institutional affiliation, lecturers and teachers, an aspect of their world purified from most philosophical accounts of science. But this is wrong. A good simulation involves a good simulationist, a scientist who has learned to work with the equipment, someone who can get it working, often in ways that no amount of theoretical knowledge will replace.

> “In most cases you have to adjust a number of parameters and it won't be obvious from the theory what they should be. At that point there is a user bias in there. Even if you give two different people same experiment, same model, they will get different results. They have different ideas about how well the model performs in different ways.” (TX)

For some of these criteria there will be well established conventions about how things should be set up. But such conventions only apply to a sub-set of the total set-up and the question will always remain whether your simulation is different enough from previous simulations to merit a rethinking of convention.

> A good example of this is mesh adaptivity. Mesh adaptivity is going to give you a benefit [in impact physics] because you need high resolution around the shock wave. This is a wave that moves very fast through the material, does work on the target, starts it in motion. It is this shock wave that you are interested in and you want to model it accurately. So you set up your mesh refinement to give you detail where the shock wave is – but you might also want to focus resolution around a particular layer, around a particular boundary in the material – so you have that well resolved as well. But when you do this you don't just switch it on; you have to give it some logical rules for where it puts resolution and how much resolution it puts. One user might want 10 levels of mesh refinement and also a very smooth gradual movement from the small to the very large. But another user might come along and say we only need 5 and we can do it quite abruptly. You are using the same technology but you are using it in slightly different ways. One person might say that you need to model the impactor with 40 cells or else you are not really modelling anything, whereas someone else might say that in their experience with their model it is only really necessary to use 10 cells.” (TX)

Your study of impact adaptivity in your model may well give you reason to think you need at least 10 levels of mesh refinement. But perhaps my model is significantly different enough to yours that it can achieve the same results with less. Even if both studies use the same modelling framework, their different choices in the model set-up may still make space for such particularities. Modelling begins always within a nexus of such formal and informal principles: well documented ideas about what is appropriate in which cases, as well as undocumented tacit knowledge about how to get the thing up and running, doing what you want, in a way that is facilitating the research. In this respect all work involves the cultivation of skill in the laboratory, and training is no separate domain. Even in terms of epistemic concerns about the reliability of knowledge gained through a simulation, considering that very few critics would ever spare the time to examine the full option-set let alone read the code, the participation of individuals regarded as skilled is no small factor in strategies of justification.

The central process in modelling, however, is the cumulative path for their construction that we discussed in sections 2.10, 'A multiplicity of simulations' and 4.6, 'Phenomenotechnique in practice'. The functionality of the model under development and the practical competencies of the modellers in manipulating this model move through a series of positions, modelling very different things, from intuitive and analytically tractable idealisations, through different kinds of more complex experimental and empirical systems. Considering that these creative processes are nested into wider process of development, validation and manipulation that characterise the broader context of the modelling framework, there is very little reason to suppose that we would understand modelling by looking at it only once it has come to a halt. While individual projects may finish, their existence as endeavours of modelling must be understood in terms of their becoming.

Research projects carry with them the whole weight of their past. While the trajectory of construction may move from a mathematical model of an analytical solution to a model of a well-studied experiment, the results of these previous stages become concretised in the apparatus as part of a testing system. When a scientist moves on to model something new, it is important to be assured that changes made in doing this have not undone earlier successes that built the foundation for the project. So as a test incorporated into the automated build and test suite, the earlier result will be run every time modifications are made to the code, ensuring that confidence from past success can still hold. This is just the tip of the iceberg, because for Fluidity over 14,000 tests are run every day, automatically flagging up unforeseen problems. This has been a big focus for the group's methodological approach. When a model is under active development, it is never enough to cite validations and verifications that have been made in the past, because these have been made with respect to a different code. Past verification and validation is accreted in the present system of research and is thus carried forwards with current research projects, applied over and over again to every new iteration of the code.

All of these dimensions speak of the need to see models in terms of the wider processes of modelling of which they are part. There is little sense of speaking of a model in the singular. Modelling processes are the practical, phenomenotechnical becoming of research itself.

5.8 Representation in simulation

Taking representation on Rheinberger's continuum requires viewing it as a play of differences. Sometimes these openings are strategic, and other times they are less amenable to manipulation. I deal here with two categories. Firstly, there are the conventional ways in which representation becomes an issue in modelling. In these cases, the simulation is already posited as a representation, and the question is more how to manage and manipulate what is represented and how. In the next section, we will encounter cases where shifts in the wider technical milieux start rendering a thing problematic in new ways, something for which representation was not previously an issue, unsticking it from the pole of realisation and opening a new field of play which starts becoming an issue in research.

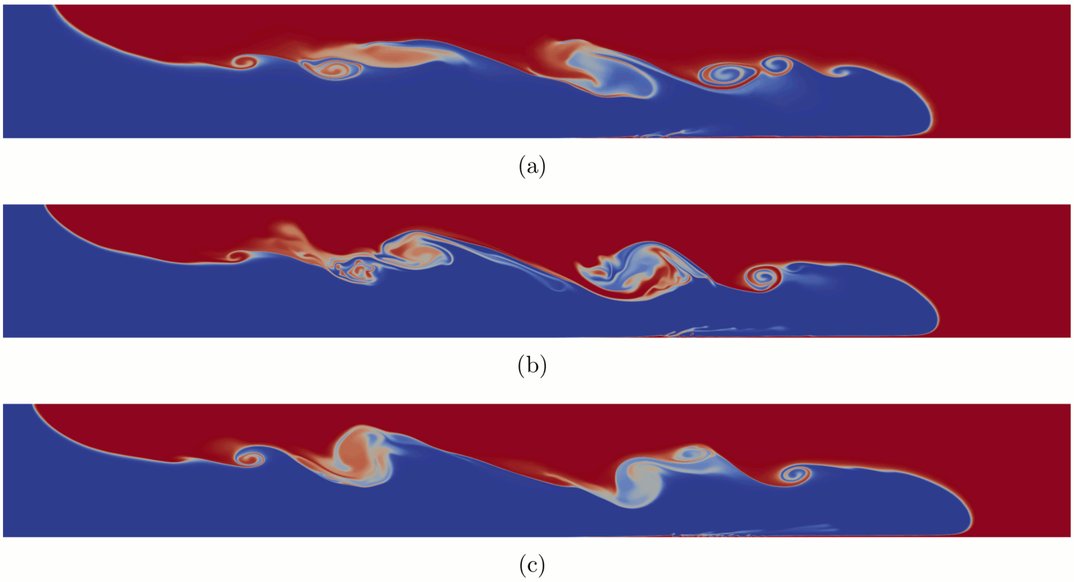


Figure 5: Three snapshots of turbulence in a simulation of the lock exchange

The foremost representational opening for a simulation renders the simulation itself a mark of something else. What is often neglected is that this relationship goes in at least two directions. Firstly, the simulation is a representation of an empirical or ideal system. In the case of the simulations of the lock-exchange that we will encounter in chapter 6, and which is shown in figure 5, this system is easily identified as an experimental apparatus for studying the mixing of two bodies of fluid of different density. On the other hand, however, the simulation is always also a representation of the solution to a set of equations. One of the reasons that simulation is such a prevalent technique in the study of fluid dynamics is that for the majority of complex fluids problems the Navier-Stokes equations are intractable. It can be argued in turn whether these equations themselves represent real fluids, ideal fluids, or whether they require modifications even for that[315]. And as continuum equations, they are at odds with theories of molecular mechanics, not to mention quantum mechanics, but at AMCG scientists are concerned with macroscopic phenomena, for which the Navier-Stokes equations are generally taken as good enough, and their own representational duplicity is left unexamined.

In a sense, we can think of the lock-exchange problem as a triangular set of representational relationships, in which the problem can be specified as an intractable problem in continuous

315 Bloor, ‘Sichtbarmachung, Common Sense and Construction in Fluid Mechanics’; Batterman, ‘Idealization and Modeling’.

mathematics, a tractable approximation in simulation, and an experimental set up. Each represents the others in certain respects. The Navier-Stokes equations become an issue when working with a model of the lock-exchange. With experimental data standing for the correct solution, the process of modelling can tease out details of the internal dynamics of the system on a level impossible experimentally. The empirical problem of fluid flow in this experimental set-up can thus also be at issue, with both the simulation and the continuous equations serving to represent it. Rather than regarding these different configurations as determining different fields of study, it is closer to actual scientific practice to regard as fundamental the process of their interplay. "The process of modelling," says Rheinberger, "is one of shuttling back and forth between different spaces of representation. Scientific objects come into existence by comparing, displacing, marginalizing, hybridizing, and grafting different representations with, from, against, and upon each other"[316].

Quite how representational relationships are opened up depends on convention and circumstance. One frequently cited adage from continental philosophy is that there is a slippage of signified to signifier[317]. In other words, wherever we try to grasp the target of a representation, the target reveals itself as just another mark, standing for and embodying another target, for which we only have vicarious contact. In SS's investigations of the lock-exchange, for example, experimental data is used to validate the simulation, holding the experiment steady as the target of the simulation, which lets us see it in new light. But the project wasn't just about the lock-exchange. It had general applicability to oceans problems involving gravity currents, such as the flow of dense saltier water out of the Mediterranean basin, through the Straits of Gibraltar, into the Atlantic. The simulation may represent the experiment, but the experiment is a representative of a much wider class of fluids problems. Recalling the terminology above, the experiment embodies gravity currents in general, because a particular gravity current is realised in its set-up. But it also can only vicariously stand for gravity currents, because most oceanographic gravity currents exist a far larger scales, and involve much more complex topography.

These relationships are entwined in the research. Research proceeds not by isolating one such relationship and concentrating on that, but finding ways to play with many such relations' combinations. And these relationships are not confined in their effects to one field of practice. They may traverse many. Indeed, it may be that the processes of modelling carried out by simulationists establish the embodiment of gravity currents in general in the lock exchange experiment to a degree that is not possible solely via the practice of the experimentalists.

316 Rheinberger, *Epistemic Things*, 108.

317 See, for example, Lacan, 'The Instance of the Letter', 419.

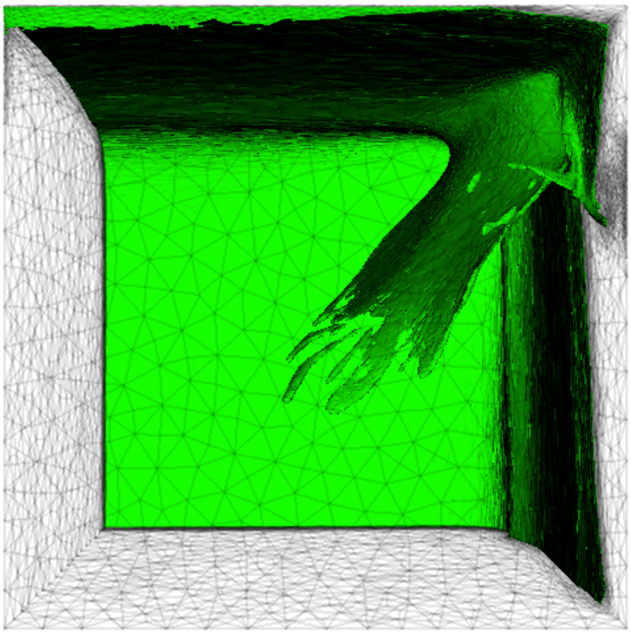


Figure 6: A visualisation of a time step in a water column collapse simulation

These relationships play out not just in private, but across the whole breadth of private and public situations in which research is proposed, undertaken and defended. In 2010, at an AMCG presentation of current work on the water collapse simulation, another commonly studied "experimental scale" problem, a visiting academic queries an anomaly in the amount of mixing found at different time intervals when the simulation is compared to the experimental data. "I can see you have a good fit," he said, "but can you explain why you don't manage to pick out this peak in mixing that is there in the experimental data at the very first time steps? It seems to be missing in your simulation." This discrepancy was visible in the graph projected at the front of the room. To this QY, the doctoral student who was presenting, responded: "Well, we don't know for certain what is going on there, but we think that is an effect of the drag caused by removing the barrier." The water collapse starts, in experimental form, with the removal of a barrier that is holding a body of water in one corner of a larger container. This barrier is designed such that the effect of its removal on the flow is minimal, but that effect can never be zero. If the simulation was really attempting to represent the experiment, surely they would have included an extra forcing factor to simulate the barrier removal? Confronted with this discrepancy, the relationship slips away. Does the simulation really represent this experiment? Simulation and experiment were two terms in one relation, but in this new situation they are revealed as two different representations of another, hitherto hidden target, an idealised water collapse. Both experiment and simulation embody this idealised system in their own distinctive ways, prone to their own characteristic virtues and defects. The idealised system exists insofar as it is so realised, and is more fully existent the more vessels are assembled around it. The simulationists, when put under pressure like this, demonstrate that they are not slavishly representing an experiment after all, but trying to represent that thing that the experiment itself, in its own way, is representing, putting themselves and the experimentalists on level pegging. This circumstance forced the unpacking of what is meant by the "water collapse", a term unproblematic in routine discourse. But it is not just a question of the meaning of terms. It is a question of *what is there* in these research practices.

Similar kinds of issues are raised when handling data from experiments, which for many simulationists is the closest thing they ever get to experimental research. "Experiments always have a certain error in them," says HU. "The best thing you can have [for comparing your simulation results with] is an analytical solution." How can experiments have errors in them? They are real, so from what can they possibly deviate? This is the same kind of issue that Hacking raised when he asserted that "phenomena" are rare in experimental science[318]. Not everything counts just because it is real; what matters is defined by a finely balanced relation to scientific endeavours.

Experiments therefore cannot serve modelling practices as origins, as points of anchorage in the real. They are themselves participants in an already complex and dynamic network of relationships, and serve as representations engaged in a commerce with those of the simulationists. The experiment as experiment embodies something significant, which both is and is not present in it.

While the simulation is already rendered a mark of certain kinds of mathematical, empirical and ideal problems, it important to note that it is also an artefact studied in its own right, and this opens up further layers of representation. When research is conducted under the orientation that I called a "science of method", particular modelling practices come to embody wider technical dilemmas. Fluidity is a fluids model and a finite element model. It involves techniques for parallelisation, mesh adaptivity, and other kinds of mathematical systems. In these respects, there are many levels on which a simulation can stand as a representative of a variety of wider classes of simulations, on which a technique can be representative of other kinds of techniques, and on which a modelling framework breaks out of its own specificity and embodies something wider. To swap to Rheinberger's terminology once again, these are expansive movements of embodiment, in which what is occurring in the research is more than its particularity; something bigger is afoot.

One recent project revolved around the classic problem of flow past a cylinder. Fluid is forced through a channel and as it moves past an object (the cylinder), turbulent effects are observed. Again, this is a classic type of experiment that can also be taken to represent wider fluid phenomena, as well as an idealised theoretical model. But in this case, another opening was manipulated, as the scientists discovered a key dependency between the model set up and the effects observed. Varying the degree of resolution around the boundary layer, the layer of fluid at the very edge of the cylinder, is shown to produce qualitatively different effects in the turbulent flow (how it oscillates and the stage at

318 Hacking, *Representing and Intervening*, 227.

which it breaks into streams and eddies). The hypothesis that was presented in one public seminar was that given the importance of this kind of these kinds of flows in ocean currents, if this simulation is representative of other simulations, we might doubt the results of models of ocean flow wherever they fail to give a certain degree of resolution around the boundary layer over the topography. The questions from the audience revolved around the issue of whether this simulation really can be taken as a representation of other kinds of simulation. The issue was not how this simulation represents the empirical data against which it had been validated (which came from laboratory experiments), but rather whether there were other factors that made this simulation too specific to be used as an embodiment of more general quandaries of modelling.

These are all very well established manners of representing that are routinely combined within the laboratory. But the next section concerns aspects of the modelling process that end up opening issues of representation in unforeseen ways. This is an important point because it captures the sense in which representation is constructed, but without necessarily involving any specific intentional achievement. We cannot explain how these things come to be an issue by reference to the intentions of the scientists; we need to look at the wider milieu of their practice.

5.9 New kinds of representation

New forms of representation can appear where a previously unproblematic realisation comes unstuck, travelling into the territory of embodiment. One notable example is in the relation between output data and the code that generated it. For the majority of cases, this is not at issue. The data produced by a simulation is simply identified as the causal outcome of running the code, the code amounting to a complex arithmetic equation, and the data produced being its answer. The two can be identified because the code directly creates its answer. The code is realised in its output. It is the same every time, as the logic gates of the processors mechanically crank out the outcome as inevitably and precisely as apodictic logic.

For the most part, this circumstance is maintained, and the data as a direct realisation of the program forms the unproblematic basis for research. It is at the far end of the spectrum, an extreme at which there is no sense of vicarship opening up a play of presence and absence. But certain trends in computing have started to break down this relationship, to create a subtle shift in the other direction, from which we might doubt it would be possible to return.

Errors can creep in at all stages of the computational process, from interference among components to read/write errors when accessing memory. There are many techniques for identifying and minimising errors, and in the vast majority of cases there will be no difference made to any aspects of the simulation that are of interest. But as simulations get larger and more complex, running larger and larger sets of calculations on larger supercomputers, even the smallest probability of hardware error starts to become significant, and at a certain point tiny glitches can be assumed to be the norm.

In most cases, they will make no difference, as when a perfect statue is cast from a mould but under microscopic examination of the surface tiny bubbles can be seen. However, nonlinear systems can be very sensitive to the slightest of perturbations, and such difference may well count for something bigger. “Very small differences in the hardware or software design can have large consequences for [computing nonlinear] systems. One can think here of the use of different floating point precisions (i.e., the number of significant digits used by computer chips) or very small changes in the order of arithmetical operations (which would only produce no differences in outcome if the precision were infinite)”[319].

However minor these concerns, however much they may be countered by techniques to minimise their impact, they begin an inexorable displacement of the relationship between code and its output data along the spectrum toward the middle zone of embodiment. No longer realising its output in a straightforward manner, the code becomes subject to the material complexities of its machinic environment.

Supercomputers are massively parallel, and in the future only going to get more so, and thus we can posit that there is a certain class of simulation that will only ever be run in parallel – it would take too long for any serial computation. Once you start to scale up parallel operations, the materiality of the medium further intervenes in disrupting the relationship between code and data. Digital computers store only a finite number of digits for the variables they compute. This means that they often round off the calculations they perform. In normal arithmetic, if a computer stores five decimal places, the answer to 0.43954 multiplied by 0.68392, which would be 0.300610197, will be stored simply as 0.30061. This rounding is a fundamental feature of computational finitude and means that the order of operands makes a difference[320]. $(A*B)*C \neq A*(B*C)$. This would be all well and good if a supercomputer always performs its operations in the same order. If you ran the same code twice you would get the same output. However, the physics of the machine matters, and this was something WS was keen to point out. Differences in temperature distribution across the array of parallel processors can result in slight discrepancies in the order in which the results of sub-processes are combined. Even without errors, it thus becomes impossible to identify a unique and definitive output from a simulation. The relationship between the data generated and the code is open for negotiation, and a new space of representation is up for grabs.

Other materialities are also contributing to this trend. Large simulations can easily generate terabytes of data, quantities that place enormous stresses on systems for transporting, storing and viewing data. For most such simulations, only a minimal subset of the output is written to disk as the simulation proceeds, and of this only a fraction is ever downloaded from the supercomputer to the desktop workstation, and the manipulation of the data there is subject to further constraints according to its hardware and software environment. There are many systems in place to assist these

319 Petersen, *Simulating Nature*, 34.

320 See Edwards, *A Vast Machine*, 175–176.

processes, often involving writing the code such that it performs a first stage of number crunching as it goes, to minimise the data to be stored. For a many of these big simulations, therefore, no single "output" ever exists. It is already mediated by all sorts of other interventions.

To be clear, I do not mean to imply that computational science is approaching a crisis. Competence in computational science already entails competence in the maintenance and manipulation of many relations of realisation, embodiment, and vicarship. It is significant, however, because this is an example where technical change has an effect within the distributions of representations. It is still very close to realisation. It is still very easy to treat the output as an analytic realisation of what is already contained in the code. But once displaced even slightly, it is hard to imagine that this relationship can ever fully be put back in its box, now its representational play has begun.

5.10 Conclusion

Modelling cannot be understood as simply a process of creating a representation of a target. And the epistemology of simulation should not be confined to studies of how knowledge is gained from such a singular relationship. Representation is best understood in terms of a play of differences, and modelling as a multidimensional process that folds such ambiguities together.

Representation has thus taken us deep into the heart of practices of computer modelling, but for the next chapters we need to pay closer attention to the materialities and temporalities characteristic of this research practice. This chapter has shown the benefits of applying semiotic insights, but within a practice theoretical framework. Representation, as I have theorised it here, is no master concept through which we are to understand modelling. The concept of practice takes over from representation in this regard. Images, texts and software systems are representations but they are, in this frame, always more than representations. To grasp modelling as practice we need to see its unfolding among these many materials. To this end, we turn next to look at images, and then in the following chapters, to look at texts and at software.

6 Images in/of Simulation

6.1 Introduction

Over recent decades, the role of images in science, engineering and design has received a lot of attention[321]. Images are among the most visible aspects of scientific material culture, and are widely used across many fields of science. In this chapter, images serve as a way in, as a handle on the research process by which we might grasp its dynamics and rhythms. They are shown to be key handles on this process for the scientists, too, who make and use them. It thus explores how we might take images beyond an analysis of representations, and embed them in an account of practice. In the course of this chapter we also gain a first insight into a key feature of computational science practice, which will be important in the coming chapters: error. In what ways is error manifest? How do things go wrong and how do they come to be grasped as such?

The first part of this chapter clears the ground by looking at images as objectifications of research. Several of my informants gave scathing criticisms of “pretty pictures” in computational physics, and this initial section traces their comments through to the basic tension that we identified in chapter 3 between the temporality of research in practice and that of accounts or objectifications of it.

The second part traces another path. It instead looks at the role of images within the becoming of research. I use them as a methodological handle with which to grasp the dynamics of scientists' investigations, the textures of everyday life within the laboratory. This side of images shows that they are important not just for depicting things. They are tools which aid the ongoing flow of research, tools for the scientist and through which an external observer may get a glimpse of these processes. It is a viewpoint from which we can grasp the roles of temporality, of rhythm and habit within the evolution of concrete research projects.

321 Lynch and Woolgar, *Representation in Scientific Practice*; Daston and Galison, *Objectivity*; Henderson, ‘The Political Career of a Prototype’; Pauwels, *Visual Cultures of Science*.

6.2 Trouble with images

Fluid dynamics research produces some wonderful images of swirling vortices, something of which many of my informants were openly proud. Occasional emails circulate through the group's communication channels calling for the collection of members' latest and best images for use on the website or on posters showing off the group's work. Such images abound in the fluid dynamics literature, adorning websites and textbooks and play no small role in defining the public perceptions of what this field is.

However, trouble arises where images form objectifications of research, and this is a particular problem in computational physics because research itself is somewhat hard to see. Its visible outputs can easily come to stand for the research itself. The problem is not only that they fail to capture the dynamics of research, but that because they are externalisations, material things with their own independence, they are hard to control.

“Images lie,” said HU; “it is much better to work with numbers”. KA commented that “there is a substantial percentage of scientists, maybe even 10 percent, who will see a pretty picture and just want to use it, without even knowing how it is validated. I think that is just disturbing”. More strong language from QY: “I think pretty pictures are an utter waste of time”, and QS went as far as to claim that “images tell you nothing”. The trouble in all of these cases was that beautiful images are compelling for aesthetic reasons, giving the effect of correctness, where caution would be better advised. They are a kind of pure rhetoric in visual form, usurping the painstaking systems of research with their aesthetic force.

This trope – “representation gone bad” – is hardly something new. It has long been part of our Western heritage[322]. From Plato's expulsion of the artists from the Republic to the history of smashed idols, representations embody an entrenched antagonism between the authentic and the derivative.

In charting the recent history of simulation methods across the sciences, Sherry Turkle shows that computer-generated images emerged as a key locus for anxieties about the legitimacy of these new ways of conducting research[323]. In the 1980s, when simulations were spreading through many scientific fields, Turkle notes that the professors at MIT “feared that even skeptical scientists would be vulnerable to the allure of a beautiful picture, that students would be drawn from the grittiness of the real to the smoothness of the virtual”[324]. “Physics faculty” she says, “were concerned that students who understood the theoretical difference between representation lost that clarity when faced with compelling screen graphics”[325]. These kinds of issues are still relevant today but for my informants, they raise less the question of the merits of simulation in general; raising instead the question of the proper role for images in their practice.

322 Latour and Weibel, *Iconoclash*.
323 Turkle, *Simulation and Its Discontents*, 5, 31, 76–81.
324 Ibid., 5.
325 Ibid., 31.

Digital images in computational science are created with visualisation software, explorations of the vast possibilities for scientists seeing what they have done when making and running simulations. The interactivity of these systems for the generation of images is a major element of research practice[326]. But this work, choosing colours, variables to display, contrasts to highlight, viewing angles, is just the start. Any thus created image can be manipulated and explored, and while easily discarded, it can also be rendered mobile, exported, saved, now embodied in a standard format data-file. It can travel and multiply, across many screens, projectors and print media. It can facilitate decisions elsewhere, catalyse discussions, act as a heuristic, and be embedded in new sites of narrative, contrast, and series.

Images are among the many inscriptions that form the material culture in which research is carried out. And as material entities, their importance is to be judged

> “not by what they depict, but by how they work. Immutable mobiles [a term from Latour] fix transient events (make them durable), and in doing so, allow them to be moved in space and time (make them available in many places). That is their power.”[327]

Rheinberger points us towards the many images of the laboratory, how they connect with the many other inscriptions (text, data, code) found across its domain, and directs us towards a study of what they do rather than what they represent.

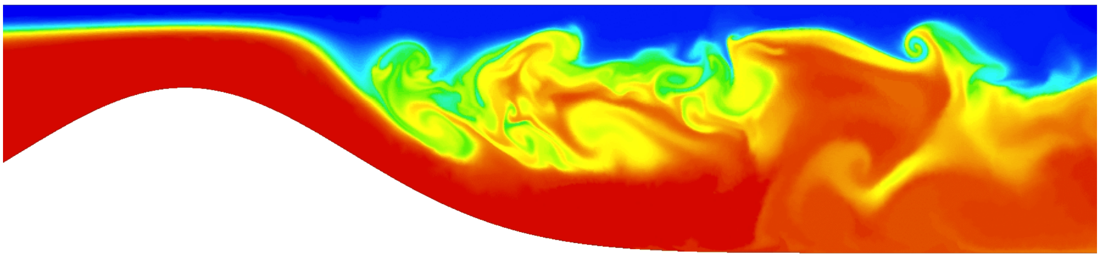


Figure 7: Visualisation of data generated by a simulation of the generation of internal waves, and their breaking, in stratified flow over a bump

To talk of representation is to talk of a shifting assembly of related traces: “material signs, entities of signification”[328]. The image in figure 7 is one such trace. Recall from chapter 5 the multiplicity of relations involved in modelling. Figure 7 represents internal waves in stratified flow and emerges from a project investigating the physical processes at work in “real world” sea loch systems. But it also represents the data-set that is embodied in its contours and colours (blue standing for fresh water, red for salty; yellow and green in-between). Quite aside from any empirical target, the image is a way to display this data set, an attempt to render visible what is embodied there. Furthermore, any such generated traces are indexical of the software that produced them and the “model” it

326 Monteiro, ‘Reconfiguring Evidence’.

327 Rheinberger, *Epistemic Things*, 106.

328 Ibid., 111; cf. Derrida, *Of Grammatology*.

embodies. Not confined to a singular motive of studying a sea loch, the research project is also equally concerned with validating the larger software framework, demonstrating that to some degree it should be judged capable of simulating turbulence in stratified flows, a key stepping stone towards simulating many other fluids phenomena. The code itself is further connected with other bodies of software and theory. Rheinberger's approach stresses the dynamics of interaction and substitution among a this wide array of relations, and he is therefore “not concerned... with the relation between theory and reality, between concept and object as such”, instead he is “concerned with describing the process of making science as a process in which traces are generated, displaced, and superposed”[329]. This point of view stands in stark contrast to the general tendency in the philosophy of representation to isolate a single relation and ask of it alone what it does and how it does it.

What we are thus interested in is a wide field of play, of many images, and many other traces, connected in many ways and involved in a dynamic process of articulation. We need to appreciate images as cognitively significant things. Following Bachelard and Heidegger, we should regard cognition as a worldly practice, not as something hidden within the enclosure of a subject: “the perceiving of what is known is not a process of returning with one's booty to the 'cabinet' of consciousness after one has gone out and grasped it; even in perceiving, retaining, and preserving, the Dasein which knows remains outside, and it does so as Dasein”[330]. Research as practice occurs among things. Scientific visual culture is an artefactual stratum that embodies something of the cognitive circuits that comprise the scientists' systems of exploration. They are not just technical but “mnemotechnical” in the terms of the philosopher Bernard Stiegler, systems of externalised memory[331].

Both Rheinberger and Stiegler draw inspiration from the writings of Jacques Derrida, for whom the kind of ambivalence expressed by my informants would be a general symptom consequent of all processes of working with and through externalisations. This ambivalence he traced to the “very condition of systematicity or seriality in general”[332]. As external artefacts, inscriptions can be “generated, displaced and superposed,” material manipulations which facilitate the grasp of their complex inter-relatedness, their “systematicity or seriality in general”, and this grasp is possible insofar as they are external. But they also harbour a fundamental ambivalence because they thereby gain an independence, a potential unruliness, threatening thought. Stiegler thus offers a normative stance. Technical systems materialise knowledge and thus render it workable for all sorts of new kinds of “distributed cognition”[333], but they equally always harbour the potentiality to “short-circuit” thought, blocking access to certain parts of the network[334].

329 Rheinberger, *Epistemic Things*, 105.
330 Heidegger, *Being and Time*, 89.
331 Stiegler, *Technics and Time 2*.
332 Derrida, *Dissemination*, 106.
333 Hutchins, *Cognition in the Wild*.
334 Stiegler, *For a New Critique of Political Economy*, 35.

In the opinions of my informants, the image poses an obstacle, a) when it convinces through aesthetic compulsion; and b) when it is therefore taken “out of context.” The proper context is one in which the intrinsic limitations of the simulation that generated the image are known. The danger of the pretty picture is that its self-sufficient existence, its coherence and integrity, come to short-circuit the broader field of the research which made it[335]. What represents, and how it does so, we are told by a wave of contemporary philosophers, depends on the context[336]. But the thing about inscriptions is that they can travel. Images travel beyond the site of their generation, and along the way there is the possibility of sloughing off accompanying contextual markers, with a variety of legitimacy-bolstering or undermining effects[337]. These markers are put down in written form in order that they may entwine the travelling image in a mnemotechnical circuit, enveloping it with traces of its origin. In publication, it is epistemically virtuous to mark the narrative with caveats, largely consequences of the idealisations through which the simulation was made. “Some models,” says KA, “simply can't answer the question you want to ask”. But whether or not you can tell if a given model can answer your question depends on what has accompanied it on its travels. Obviously given the Sisyphean nature of attempting to “give” the whole context (where could you possibly draw the line?), the question is one of judging the merits of the fragments that remain.

Science, like many human activities, occurs among technical systems. But any externalisation has its independence, what often seems an animistic “life of its own”. It cannot be fully controlled, even by those with the strongest claims to authorship. The problem for images, as expressed by my informants, is a problem of control which relates to the materiality of the systems they create, manipulate, and eventually render partially independent. Images are accompanied by a web of other inscriptions that embody the much less tangible relations between the image and its site of production. It is the relation between the image and these other inscriptions that constitutes the risk of the short-circuit, and the point at which any configuration stands to be judged as good or bad. This relation is too often eclipsed in the second-order accounts of social scientists and philosophers by another relation, its more assertive neighbour, that between the simulation and the reality it simulates, which under the name of validation is the focus for a huge amount of concern and philosophical labour.

There is no reason to expect that this kind of problem will be confined to computational science. Anxieties about representation in many different spheres have crystallised around images[338]. But the novelty of computational methods and the disturbance they pose to established forms of science raises questions about legitimacy with particular force, and images in this field find themselves the outlet for much wider concerns for those whose agency and identity are bound up in these pursuits.

335 Turkle, *Simulation and Its Discontents*, 76–81.

336 Suárez, ‘Scientific Representation’; van Fraassen, *Scientific Representation*; Giere, *Scientific Perspectivism*.

337 Lahsen, ‘Seductive Simulations?’.

338 Maurer, ‘Does Money Matter? Abstraction and Substitution in Alternative Financial Forms’.

The robustness of the travelling image stands in contrast to the fragility of its relations to the traces of its context, its ties to its origins in a particular time and place, a particular research project, a model with a history and specific limits. Despite all the scientists' best efforts, this contrast cannot be overcome. The textual accompaniment is always at risk of being insufficiently explained or just written badly or wrongly. “You can't write it all down in your paper,” says GY, “and even if you do, you might get it wrong. There are examples where someone has written a paper and has got something wrong in their description of what they did, and when you do and look at the original source code, they actually did it right”. There is always a risk of being skimmed over, skipped, ignored, misinterpreted, mistranslated, lost or forgotten. Or maybe the image is just too compelling.

In these scientists' complaints, therefore, the key issue is not the image as such. Nor is it the context for visual representation. It is the material-cognitive interrelations between visual inscriptions and the markers of their context, as these wind their way through the networks of scientific culture. Such worries about images are more than a local expression of the troubles with representation. For we should not accept the assumption that the natural state of culture is stasis and coherence, but open our eyes to human agency among the exigencies of symbolic decomposition[339]. These scientific troubles would then instantiate in a microcosm the feats and frustrations of assembling culture, working against the broadest tendencies of things to fall apart.

6.3 Beyond representation

So, if images are so troublesome why carry on creating them?

We have seen that as externalisations, as artefacts, images present threats to scientists' control over the materials of their work. Pretty pictures direct our attention toward this endemic concern, this problematisation of materiality and control that is woven across the laboratory. But if we now turn towards research in the making, images take us in another direction, towards the role of these objectifications of research within its accomplishment, in which they do not obstruct its flow, but rather facilitate it.

In a general seminar-style discussion on the theme of images with about half the group, I displayed a brochure for a fluids modelling code, replete with colourful graphics of turbulent mixing, of velocity arrows flowing through jet engines, of intricate machinery modelled in detail[340]. “The trouble with this,” said QY, to general assent, “is that it makes it look like the simulation is already finished, already complete.” What is erased by these inscriptions is the process, the always incomplete becoming of their work. Whatever confidence scientists feel about their work, there is an aspect in which this sentiment is deferred towards future validations of the model, future proofs about the method, and future studies of the system. Even when inscribed in publication, research is never

339 Wagner, *The Invention of Culture*.

340 Major international companies such as Ansys offer similar simulation services to the work done at AMCG, but on a commercial scale, and with a visibly greater attention to the efficacy of marketing materials (see for example http://www.ansys.com/).

wholly present, but is always part of an ongoing becoming. What is not captured in marketing images is the fact that, in Rheinberger's Derridean terms a model is “an entity that draws its effectiveness from its own absence... [A] model is a model only in the perspective and by virtue of an imaginary reality at which it fails to arrive”[341]. These marketing images are the ones most widely disseminated beyond the laboratory but this is an expulsion from which they cannot return. While they do represent the simulation and its object, in so doing they eclipse its temporality, its coming into being.

As discussed in chapter 3, the exploratory process in itself, animated by dull feelings, vague suspicions, moods and hunches, wrapped in a culture of artefacts and procedures, is destined to be erased in the end analysis. No retrospective objectification can do justice to the indeterminacy of its genesis. Like the creation of marketing materials, the “write-up” would be a process of erasure as much as it is one of inscription. In both cases, the indeterminacy of the exploration is analytically out of focus where the “result” of research looms large, for wherever there is a result, there is a tendency to see the process of its formation only under the condition of its eventual outcome, to collapse indeterminacy into the determination of what will emerge from it[342]. As Rheinberger puts it:

> “An experimental system can readily be compared to a labyrinth, whose walls, in the course of being erected, in one and the same movement, blind and guide the experimenter. In the step-by-step construction of a labyrinth, the existing walls limit and orient the direction of the walls to be added. A labyrinth that deserves the name is not planned and thus cannot be conquered by following a plan. It forces us to move around by means and by virtue of checking out, of groping, of tâtonnement”[343].

There are many images in research that do not retain attention within their own capacities as images, in the manner of the pretty pictures deployed for marketing purposes, but rather pull it through them, beyond them towards the emerging future of the ongoing project. Here any representational capacities are implicit; “the Image” does not appear as a source of problems, because images are thoroughly embedded in technical milieux of code, equations, statistics, equipment and procedures, drawn together around a temporally unfolding research project, a background from which neither images nor any specifically visual dimension are routinely extricated or isolated.

There are very good reasons to be wary about equating visualisation with the production of visual sensation, of shapes or colours. Martin Heidegger captures this, though writing of the ears: “Much closer to us than all sensations are the things themselves. We hear the door shut in the house and never hear acoustical sensations or even mere sounds”[344]. What is significant in moments of investigation is what is seen in the image, moments in which the image itself disappears, in which what is significant is less “what is seen in the image”, than simply “what is seen”, what is

341 Rheinberger, *Epistemic Things*, 110.

342 Cf. Simondon, ‘The Position of the Problem of Ontogenesis’; Ingold, ‘Beyond Art and Technology’.

343 Rheinberger, *Epistemic Things*, 74.

344 Heidegger, ‘The Origin of the Work of Art’, 25.

encountered. Mediation is not the issue and thus not the message. Much closer to the scientist than all images is the problem at hand, the simulation, the data, a bug in the code, or its potential solution. In the time of their work, present representation is less of an issue than future realisation.

6.4 Discontinuity and process

Social scientific theory has over recent years become increasingly concerned with process and becoming, and in a sense my analysis is indeed intended to take us in this direction. Pickering captures this trend in his assertion that “we live in the thick of things, in a symmetric, decentred process of the becoming of the human and the non-human”[345]. But why have we not always seen things in this way? Why is dualism so entrenched in Western culture? It would be easy to claim that modern scientific viewpoints are responsible, and indeed Pickering will continue: “But this is veiled from us by a particular tactic of dualist detachment and domination that is backed up and intensified... by science as our certified way of knowing”[346]. To be clear, however, it is not science itself that is responsible, as if scientific research was dualist and detached, but rather a metaphysical position that stereotypes science, that requires a “zoomed out” view that brackets off the improvisations and processes through which research itself is carried out, for the insight of science studies has been that if we look in at the detail of practical research we see that in fact “science is itself caught up in the flow of becoming...”[347]. What is needed is not a rejection of science, but the performance of a figure-ground reversal[348]: “a gestalt switch between the margins and the hegemonic centre of gravity would be a way of putting dualist ontology and its associated projects in their place”[349].

Emphasis on flows and processes is often achieved through the construction of a contrast with a world of ontic beings and static structures, for example between on the one hand the inscription of the marketing image which presents the model as finalised and on the other hand the research process that is always deferred, always reaching out towards an open future. In the context of talking about anthropological narrative, but in words that could equally apply to the navigation of research, Strathern points to a potential naivety here. The danger is that these flows and processes are imagined as smooth, a smoothness which is an artefact of the contrast drawn for rhetorical purposes, rather than a feature of the process itself.

> “Ideas and arguments are often regarded as “flowing”... The time it might take to travel, as the reader moves through the text, gives a kind of experiential unity to the exercise. Yet this unity or sense of flow or movement is at the same time made up of jumps over gaps, juxtapositions, leaps – unpredictable, irregular. So, continuous as the process of narration

345 Pickering, ‘New Ontologies’, 8.
346 Ibid.
347 Ibid.
348 Wagner, ‘Figure-Ground Reversal’.
349 Pickering, ‘The Politics of Theory’, 205.

> might seem, the closer we inspect monographs, paragraphs, sentences, the more aware we are of internal discontinuities”[350].

Scientists at AMCG inhabit a world of the interplay of the continuous and discontinuous, between the continuity of the Navier-Stokes equations for an ideal fluid, and the discontinuity of the molecular dynamics that the ideal fluid overwrites for macro-scale phenomena, between these continuous equations and the discrete mathematics of the digital computer and between the continuities and discontinuities of the numerical approximation and the necessity of modelling continuities and discontinuities within the simulated system. Each is implicated in the other. It is impossible to idealise continuity as representing some more authentic essence of process. Discontinuity is not the antithesis of process, but is endemic within it, folded many times over.

Thus we might gain first a glimpse of images in the depths of the research process, of their relation to a differential of movement, manifest at moments of unpredictable leaps, of irregular jumps, all those crucial moments where the ongoing path of investigation is textured by discontinuity, gaps where what comes next does not automatically follow, not by prior plan or internal momentum, but requires some additional force. Images are fodder for habit and improvisation, the material to be grasped in order to push onwards. Somewhat like stepping stones, points of hardness at odds with the ongoing movement, without such material artefacts that movement would not be able to push onwards towards its future. They are points at which what follows unfolds. As Bourdieu was well aware, even when the question of “what comes next” (in a rite, a procedure, an application of scientific method) is as rule bound as it can be, the whole success of the enterprise may nevertheless rest on a thoroughly tacit sense of timing and of rhythm, a sensitivity to the moment for which rules are no substitute[351].

6.5 Images and intuition

The danger is that images are, to paraphrase Levi-Strauss, so “good to think with” that they are hard to think past, and hence Bachelard would claim that scientists must always be wary of them. In his view, it is necessary that scientists disrupt their imaginative faculties, that they find ways to kick the feet out from under intuitive ways of thinking that have been cultivated across the many arenas of daily life. Only then can they break past epistemological obstacles[352]. Past experience has provided us with images through which we make sense of the world, some of which yield false certainties so entrenched that only great efforts will see them shifted. Thus Bachelard would assert that “[a] science that accepts images is, more than any other, a victim of metaphor”[353].

Substance is one of Bachelard's examples. He writes of the way in which images of interiority and inwardness exert an influence on how scientific phenomena have been treated. These imaged forms

350 Strathern, *Partial Connections*, xxiii.

351 Bourdieu, *Outline of a Theory of Practice*, 7.

352 Bachelard, *The Formation of the Scientific Mind*, 28.

353 Ibid., 47.

of thought are more than ways of speaking: “there is in fact more here than just description by a word: there is explanation by a thought. You think as you see and you think what you see: a speck of dust sticks to an electrified surface and therefore electricity is a glue, a very sticky glue”[354]. Images may be useful, but they are also dangerous. They are sources of ambivalence, “pharmacological” in Derridean terms[355].

The imagination in Bachelard's view wanders among tropes laid down over a lifetime, especially during childhood, expressed particularly vividly in poetry and in dreams. These intuitions are in constant interaction with new experiences. Patterns in fluid dynamics images such as those in figure 7 recall the ways in which we dwell among fluids, in which we bathe and consume them, the ways we paddle, pour, stir, soak, and splash. Certain kinds of images are favoured, as are certain scales, which are conditioned by the kinds of bodies and environments we inhabit. As Myers has shown, far from being disembodied, computational science involves a great deal of “body-work”; scientists engage with phenomena through a whole range of bodily and gestural entanglements[356]. So while Bachelard sees a pitfall in enlisting intuition into the scientific fold, we need to supplement his viewpoint with the other side of the coin, for at times in the development of simulations, there are few better resources capable of giving scientists such a handle on their work[357].

The course of developing a model proceeds through a time in which it is still uncertain whether a simulation is behaving correctly, or what the correct behaviour might be, and, if it does appear to be behaving itself, whether it is doing so for all, some, or none, of the right reasons. Prior to the emergence of the simulation as it will be spoken about in publication and presentation, is the simulation as it is in its process of being made, when existing capabilities of the code are tested for their ability to handle the problem in question, when new capabilities are being added in order to supplement any deficiencies thus identified, and when the many hundreds of virtual dials and knobs within the model set-up and parametrisation are being tweaked and refined. This is a time before the simulation is yet ready for the formalised assessments of validation. But at this point many important questions will have to be decided, a whole landscape of possibilities to be navigated, for which precedents and rules of thumb provide only partial guidance, especially when the project also involves adding new functionality into the framework. What kind of basis functions will you use? What kind of mesh? At what resolution? Will it be adaptive and if so how? What kind of time stepping (implicit, explicit, or somewhere in between?)? What kind of boundary conditions and forcing factors? What kind of parametrisations? And do they work in combination for the kinds of procedures that will be key to this project?

During this period of model composition, the scientist is constantly modifying the model and attempting to understand the results of this modification. Experience here is not an exercise in data-

354 Ibid., 109.

355 Derrida, *Dissemination*.

356 Myers, ‘Animating Mechanism’.

357 Monteiro, ‘Reconfiguring Evidence’.

collection, but of reflection, of interacting with one's own effects among what happens. The data-sets produced by the majority of simulations run at AMCG are simply too vast to be directly understood. Errors in the emerging simulation can appear in three main ways. Some will cause a complete crash of the program. Others will produce wildly inaccurate results. And then there are errors of a range of subtleties, some of which can only be picked up later on by verification and validation. Errors can creep in at all stages, arising from the specific set up of the simulation, from changes made by others elsewhere in the code, or from shifts in the wider technical milieu (external libraries, compilers and operating systems, for example, being constantly updated and modified).

In these early phases of development, simulations usually evolve through a series of wildly inaccurate results, with the scientist trying to understand these, tame them, eliminate any problems. While otherwise hard to grasp, error is often starkly obvious when it is manifest in an image, and in many moments, even while a large simulation is in the process of being run (which may take a long time, even on a supercomputer), it is common practice to download the simulation's data, its work in progress, and visualise to “keep an eye on it”, hoping to catch any error as it occurs, if it occurs.

> “The point is not to run it for a week and then check at the end of the week and find it was wrong after day one, so that is why I keep on checking.” (IW)

Looking at a visualisation, scientists can often tell that something is going wrong, because problems are manifest in the image such that it intuitively “looks wrong”. Indeed it is claimed that non-scientists would have the same response if erratic behaviour in the model output leads to a visualisation that simply doesn't resemble everyday experiences of fluids.

> “If the water moves through the solid object rather than around it, I know there is a problem, because that is just not physical.” (AB)

If the image, often in this context a moving image, evokes intuitions of fluids this serves as a rough, provisional indication that the project remains on the right lines. This informal kind of check is thus ubiquitous in the laboratory routine. QY put it like this:

> “Would you expect that if you bent a piece of wood into a certain position, and then bent it further, it snaps? If it keeps bending and bending and didn't snap you would say that is unphysical. I guess you could say the same of fluid dynamics. There are just some things that the system just isn't supposed to do.” (QY)

On several fronts, fluids exceed codified knowledge of what they do. Turbulence, one of the central phenomena of fluid dynamics, implies for modellers that there are dynamics going on at scales smaller than those they explicitly model. CQ put it enigmatically: “I think one of the best ways to put it is that turbulence is that thing that you can't ever model”. There are always phenomena that escape the model, and thus no absolute or “mechanical” means for recognising a fluid to serve as a yardstick against the fallibility of intuition. We should therefore be justified in offering a light critique of Bachelard's damning view of the role of intuition in science. In some circumstances, it is integral to the unfolding of the phenomenotechnical milieu.

In other cases, however, errors are less apparent. They do not show themselves so easily, and the scientist relies on long experience in working with this kind of simulation, experience which conditions the space of practice opened by the visualisation. Work with images is often a time for inclinations to linger longer on a certain stage of development, to double check some settings, to run an extra diagnostic or to look at a different visualisation of the data, movements through the process of investigation that open further spaces from which it may become obvious that there is a problem lurking somewhere within the system, or which may set up a confidence sufficient to move on towards the next steps. But rather than ever actually affirming that the simulation is correct, the best these intuitive checks can offer is a provisional double negative: it is "not wrong". It is not even that nothing is wrong, but that nothing appears wrong, in the initial check. "Just [by] opening it up and looking at the colours in Paraview, you can't tell it is correct, you can just say if it is wrong" (QS). It is not infrequent for something that "looks right" will later on turn out to have subtler problems that were just not the kind of thing that would show up in the image.

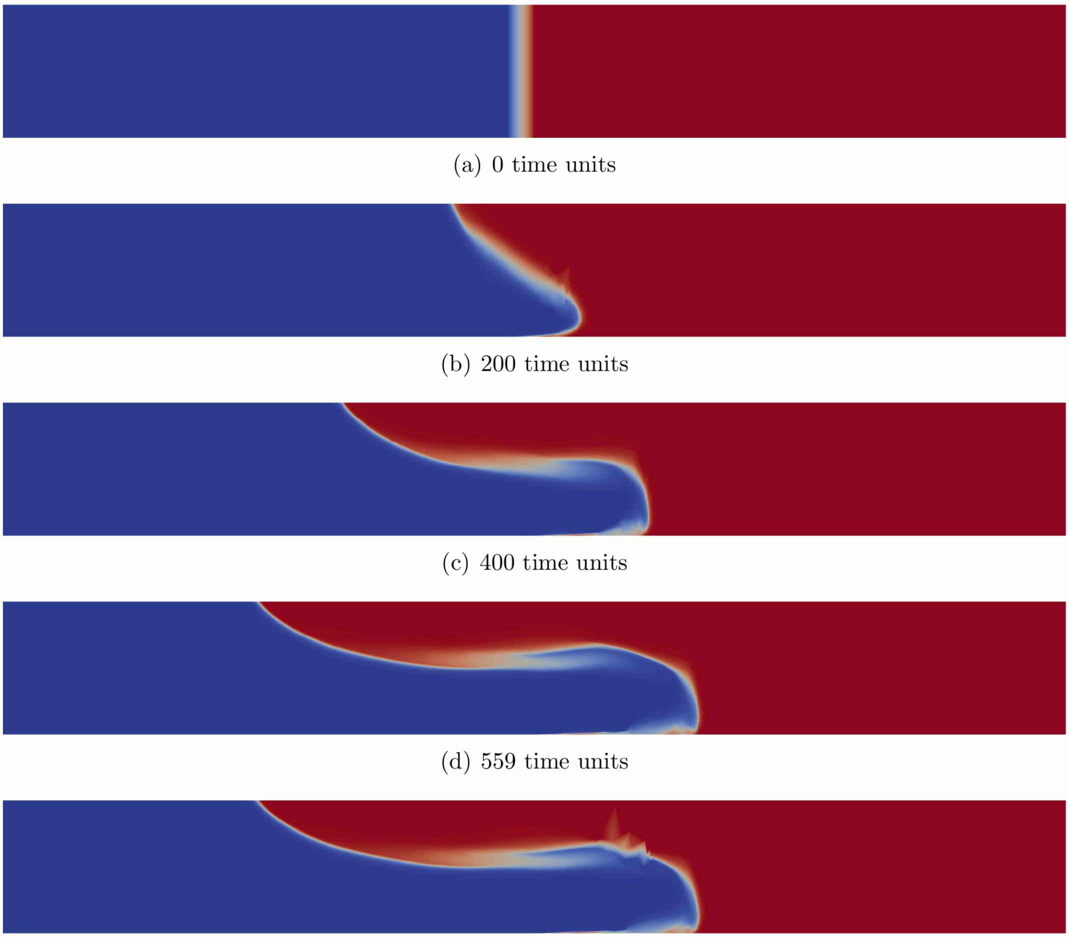
> "Looking at a visualisation, you can eyeball it and see if there is something obviously wrong, but if it looks roughly like what you expected then you would do your diagnostics and find out exactly how much it looks like what you expected." (QY)

An image "looking right" provides little ground for any kind of formal epistemic confidence, but it does exert an influence on the scientists' course of investigation by indicating that things are stable and "not wrong" enough at this point to move on, to take another step, to go and introduce further elements of physics, simulate a system of greater complexity, or to attempt to formalise the data outputs according to statistical measures, through coding up diagnostics, a mathematical transformation of the data-set that often in the end reduces the need or inclination to visualise it at all. The image is no substitute for the diagnostic measures that distil the data into numbers that can support some kind of positive conclusions, a much more formal grounding of confidence which may reveal issues invisible in the image. But these kind of formal measures are not usually available until later on in the development process.

> "The initial visualisation will tell you if it is wrong, if the velocity is flying off the sides or whatever, but it won't tell you if it is correct. It can never tell you if it is absolutely correct and I suspect that is why I plot these graphs now. When I first was developing this [simulation] about twenty months ago when I came here I probably did open the file and look at it and say "its wrong" or "its right" but it has been looking right for 20 months now, but it has actually been wrong. So looking at it wouldn't have told me anything which I guess is why I have moved into distilling it down into a graph." (QS)

6.6 Navigating error in the lock exchange

Error is encountered in the image but the real importance of visualisation is only apparent when we note that this encounter can go much further. It can draw attention beyond the mere fact of error, towards its underlying cause and towards the future of its eventual resolution.



(a) 0 time units

(b) 200 time units

(c) 400 time units

(d) 559 time units

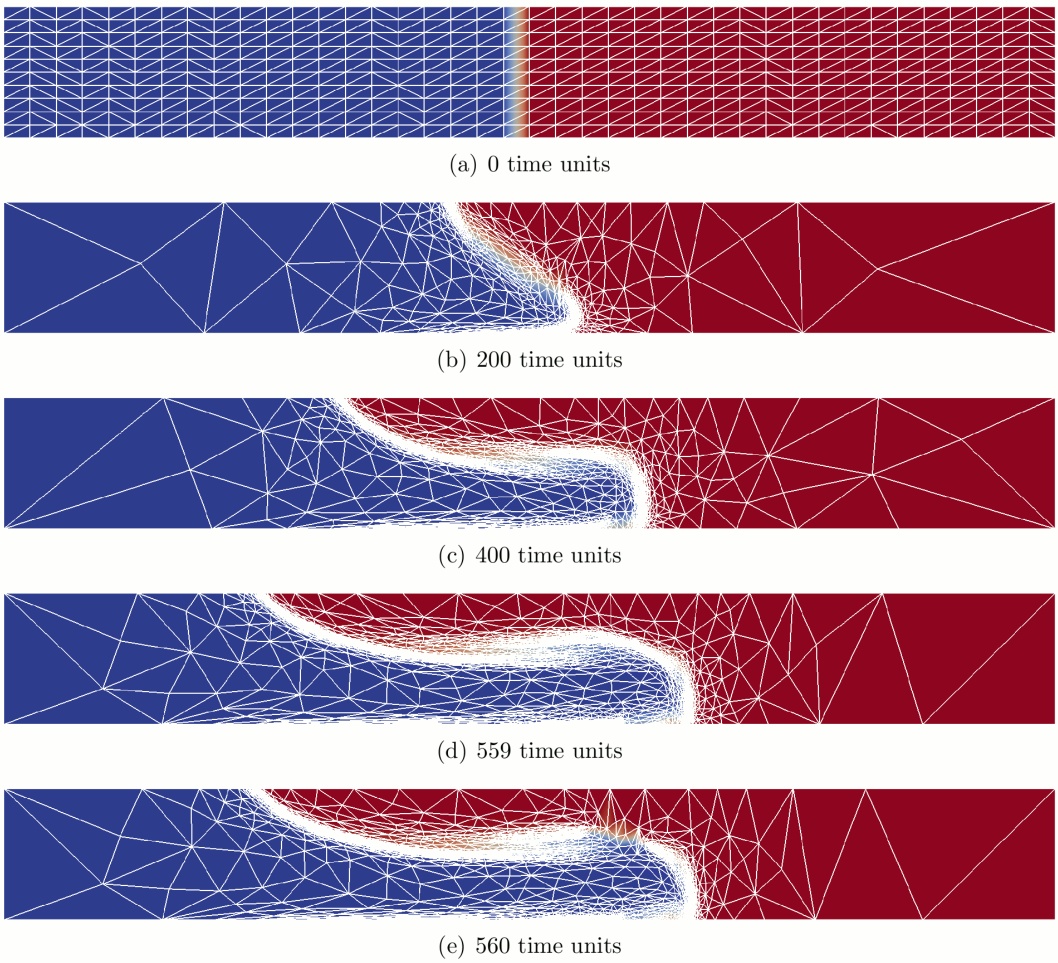
(e) 560 time units

Figure 8: A cross-section of the lock exchange simulation, at different time-steps.

The images in figure 8 show data from a simulation of the lock exchange, a well studied experiment where a tank is filled, with the aid of a barrier in the middle, half with hot fluid and half with cold fluid. The barrier is removed, and the mixing that occurs is a paradigmatic example of “gravity currents”, currents driven by the differing density of the two bodies of water. In this case, something has gone wrong.

The simulation that is visualised in figure 8 is a “short test”, a small simulation that is automatically run, along with thousands of other tests, by the automated testing server every time anyone changes anything within Fluidity's code. The purpose of these tests is to let scientists know when changes have affected key results, so that they can be confident that they will be able to reproduce a result they got earlier. This system, coupled with the code repository, introduces an element of stasis, of reproducibility, into a field defined by perpetual technical evolution. In this case, the example was set up with a diagnostic that gives a statistical measure of the amount of mixing between the two bodies of water. An alarm was triggered when the test suddenly started exceeding the expected range of values (expected from the experimental and computational precedent), alerting SS, the “owner” of the simulation, to the existence of a problem.

Having been thus alerted, SS knew because she designed the test case that the symptom was excessive mixing, but with a massively complex code such as Fluidity you are guaranteed no simple way to diagnose the cause. SS immediately visualised a cross-section of the simulation, plotting temperature. The result is a succession of images that she can click through or play as an animation, to see the data at each time step. A selection of these snapshots is displayed in figure 8. All time-dependent simulations must approximate the continuum of time that underlies the Navier-Stokes equations by a series of discrete steps. SS's attention was drawn to peculiar jumps in behaviour between consecutive time-steps, such as the sudden spike in mixing visible between 559 and 560 time units (d and e in the diagram). Individually, both 559 and 560 show intuitively feasible results. It is the difference between them that stood out. The abrupt change on the boundary should not occur, for there is no process simulated between the time-steps that could explain it.



(a) 0 time units

(b) 200 time units

(c) 400 time units

(d) 559 time units

(e) 560 time units

Figure 9: As in Figure 8, but with mesh overlaid

Having spotted this, SS went on to visualise the simulation in another way, to create different images through which she can hope to push onwards towards a more specific idea of what was going on. She chose the same visualisation set-up, but with the addition of an overlay of the mesh, to check how it was adapting between these time steps. The mesh is the discretisation of space, which like time must be divided into a finite number of discrete units in order for it to be dealt with on a computer. In figure 9 you can see the mesh evolving from a regular starting distribution towards a distribution that concentrates resolution on the boundary, where the smaller scale dynamics are occurring. Looking at figure 9, and seeing that the abrupt change in mixing corresponds to an abrupt change in the resolution of the mesh, was a good indication that the problem was something to do with the adaptivity algorithms.

That it is a local problem, affecting only a sub-set of elements (you can clearly see a localised section of the boundary markedly relaxing its resolution in figure 9) raised a strong suspicion that the problem resides at a deep level: in the parallel processing of the simulation. These finite element simulations are designed to run on multiple processors by dividing up the elements within the domain (the triangles in the mesh) so that each processor handles a small sub-set. It seemed likely therefore that something was causing just one processor to behave strangely. This points SS towards further pathways, checking that this was not a hardware malfunction by running a repetition of the test, and adding weight to her suspicion by trying it in serial (all on one single processor).

That the investigation heads in this direction takes the issue further from SS's own domain of expertise, and having exhausted her own ideas for what would be causing the processor to go astray she took the issue to a weekly meeting, which is usually attended by several scientists who are specialists in parallel processing. Having the problem externalised in these visualisations had facilitated the process of investigation that pointed towards it being a parallel processing issue, and then later these images can be brought into the communal forum to catalyse a dialogue with others, enabling the SS to marshal assistance to help find a way through the technical system towards the source of the problem, a common tactic in a group with a strong communal dynamic. Images tip the pathway towards new courses of action, new tests, new forms of scrutiny, new suspicions. But they also bring individuals together.

> “It is fairly common to go to others to see if they recognise what is going wrong. Not just in visualisation, but if the compiler fails and someone comes to me and I recognise the error message and know how to fix it... same thing with visualisation. If it goes wrong and the velocities are pointing downwards I have seen a similar thing: 'I saw this... I did this... it should work'” (QS)

It is not pure goodwill that SS relies on for help. There are bugs in all complex software, some of which can be very difficult to analyse and fix. It is in everyone's best interests to fix bugs when they appear. The bug's manifestation in a failed test such as this could provide a sufficient handle with which to grab the opportunity to iron out a problem. If ignored it would just lie dormant within the code only to cause problems in the future. It is not just a single scientists' problem, but a problem for everyone who is invested in this modelling framework.

> “All the libraries that we depend on, so PETSc and VTK for example, they all have bugs. We can't fix them either except when we hit one and we fix it and report it back. Compilers have bugs, operating systems have bugs. There is no way we can get rid of them all. We often hit compiler bugs. It is just part of computer modelling as a whole. You are going to have bugs and you are going to hit them. You just have to make sure that there aren't enough bugs that it derails everything, which is why we have test suites. Most research centres by the way do not have test suites.” (QS)

In many cases, an automatic alert like this will lead to the problem being traced to the most recent change to the code, the change that triggered this particular round of testing. This can then be searched for errors and fixed or just “rolled back” and removed. But this case was slightly more unusual, and the change in question was not itself the source of the problem. This change had modified the operation of the code in a subtle way that had allowed a deeper more insidious bug to manifest itself. Some bugs, of course, are simple: a bracket too few or too many in a newly added file, or a problem with how the model has been configured. But others can be pernicious.

> “Sometimes when I see certain problems that I have never seen before someone will say 'That seems like this is going wrong' then I will have a look and find out. 'Oh yes that's what is going on'. If someone is really stumped on a problem they will ask and hopefully someone will have seen something like that before. But occasionally you get a problem which no-one has seen before, so you just have to battle through it.” (IM)

6.7 Becoming a scientist

The research process does many things. It produces knowledge and generates complex software systems. Yet further to the production of knowledge and equipment, the repetition of seeing and working upon problems, bugs, causes and solutions also forms a basis of scientific habituation, a milieu of sensitivities across the community. When we look at the meandering course of investigation, it is impossible to separate the becoming of the process of research from the process of becoming a scientist. Recalling Ingold's critique of skill transmission discussed in chapter 3, the skills of scientific practice are not transmitted through the communication of codes of conducts. There is an art to science that requires habituation. “Since an art cannot be precisely defined,” says Polanyi, “it can be transmitted only by examples of the practice which embodies it”[358]. To go further, we should extend this point, and say it is not even a question of examples. It is a question of becoming a co-participant in a field of practice, a co-constituent, to give oneself over to its contours. Over countless iterations of problems and solutions that have characterised his or her work, what the scientist sees (in the image) is subtly and strongly conditioned.

358 Polanyi, *Science, Faith and Society*, 15.

Don Ihde talks about what he calls the "multistability" of images, the plurality of what can be seen in an image, and he uses the famous image of the "duck-rabbit" to make this point[359].

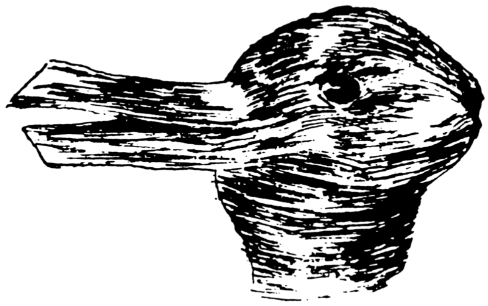


Figure 10: The duck-rabbit

The duck and the rabbit are both possibilities here, but with respect to scientific visualisation, it would help to extend this image, and to imagine one where the duck is more stable than the rabbit, one where only certain people would, through long years of encounters, be habitually inclined to see the rabbit, and one where the rabbit even at its clearest to even the most rabbit-inclined observer is still a slightly tenuous and vague suggestion, a fleeting suggestion like a shape recognised in a passing cloud. Alac and Hutchins explore this habituation through a distinction between seeing and looking, writing of a "trajectory that took the novice from *looking* at the image to *seeing* the structures in it "[360]. But the precariousness of recognition, its instability, is equally important[361]. Training does not just provide new kinds of clarity, but opens new windows for new kinds of ambiguity to be brought to bear.

Where habit is at stake, it is apt to think back to the quote of Rheinberger given above, where he notes that the walls of the labyrinth "blind and guide" the experimenter. The blindness is the lack of clear foresight of what is to come, the basis of the improvised decision, following a vague suspicion. But the same thing that blinds also guides, regulates and determines. This interplay between the regularity and difference, between the rule and the act, is commented on by both Bourdieu and Deleuze, who must be recognised, for all their differences, as two of the most influential commentators on habit. It is in precisely this vein that Bourdieu characterises the habitus as the "durably installed principle of regulated improvisations"[362]. Improvisation is regulated according to principles "which can be objectively "regulated" and "regular" without in any way being the product of obedience to rules, objectively adapted to their goals without presupposing a conscious aiming at ends or an express mastery of the operations necessary to attain them and, being all this, collectively orchestrated without being the product of the orchestrating action of a conductor"[363].

359 Rosenberger, 'Quick-Freezing Philosophy', 68; Ihde, *Expanding Hermeneutics*; cf. Wittgenstein, *Philosophical Investigations*, 204.

360 Alac and Hutchins, 'I See What You Are Saying', 659.

361 Kruger, 'Visualizing Uncertainty'.

362 Bourdieu, *Outline of a Theory of Practice*, 78.

363 Ibid., 72.

Deleuze points to the novelty that emerges from and resides within the repetition of the same. “Habit draws something new from repetition – namely, difference... In essence, habit is contraction. Language testifies to this in allowing us to speak of “contracting” a habit, and in allowing the verb “to contract” only in conjunction with a complement capable of constituting a habitude”[364]. Over many instances of generating, navigating and working with and through a great many different images, working through them in the same manner and mode, according to the same kinds of code, the same kinds of physics, and the same kinds of problems, sensitivities are established. Through these, research is regular, yet this regularity is what opens up the space for irregularity, for improvisation, following barely grounded suspicions to investigate further, suspicions which may well later confirm themselves, a confirmation that only emerges through having already had to act, to follow a suspicion when it was no more than a vague feeling. Those that come to nothing constitute departure points for further exploration while those that are confirmed shed their vagueness, and establish a retroactive teleology that collapses the discontinuity of the decision into the unity of the continuing process. The form of the scientific everyday is not stamped upon it by some *a priori* definition of the correct scientific method, but grows instead out of the difference and repetition of research practice.

6.8 The fate of the image

In the above the problem (the error) is seen through the image. The image is not the problem. It is not that the image does not represent, but the question of their representation risks establishing a discord between the questions of the analyst and those of the subject. Images of the lock exchange may well pose questions of representation when at a later point the research is written up for publication. There, the multiplicity of their targets will be unpicked. Do they represent the data? If so, how well? Or are they instead to be representing the simulation? Or the target of the simulation? Or the simulation's representation of the target? Which image will be chosen out of the countless possibilities that can be generated from the same data-set?

But the images exhibited above in figures 8 and 9 will never make it this far. They will never be subject to such questions and will never find their way into publication. They are generated from a simulation which has gone awry. They are part of the greater stock of scientific visual culture, all but the tip of the iceberg, never disseminated, never anchoring truth claims: disposable images created in the between times, finding their place in the midst of the repetitions of daily life, never in the final analysis, where they are written out of wider concerns with eventual results. At such points within the becoming of research, even the most deceptive image would be valuable if it helps SS find her orientation, for at issue is what will bring the process forward, and establish for it a future. Confident or tentative, research takes its steps forwards according to such materials, catalysts for its progression.

364 Deleuze, *Difference and Repetition*, 94.

In the end, the image can never be left alone. It gets accompanied by other kinds of narrative and inscription. While SS works to keep her simulations functional, combating problems such as the one described above, she writes her research up into publications, and here visualisations take a back seat. Here the big question is the representative capacity of the simulation, not of its images, and it will be measured by diagnostic variables, statistical measures of its fit with experimental results, with other scientists' simulations, and with different set-ups of this simulation modelling a range of analytic and empirical targets. Much more than visualisations, these diagnostics play the central role in the strategies through which claims about the lock exchange and about simulating both might be justified. For this reason, the image draws the eye back into the incomplete process of research, rather than anchoring it to its results, its eventual outcome, which is characterised by an general desire for mathematisation.

> “You want hard numbers... Turbulence is a good one because it will produce some very pretty animations of flows oscillating and you have the wave region behind something and it is all circulating and it looks amazing, but it only means something if you start analysing it statistically, finding out what it does on average and finding out what the scales of motion are there...”(QY)

6.9 Conclusion

Winsberg notes that simulation does not fit squarely within the concerns of traditional philosophy of science[365]. He attributes this in part to the fact that simulation is usually based on theory that is already known, so it would be easy to assume it would not pose any problems outside those that are already tackled in the philosophy of scientific theory. In other words, if a simulation is built out of theoretical knowledge, it can only discover what is already there. Theoretical systems, such as the theory of fluid dynamics embodied in the Navier-Stokes equations, are in many cases analytically intractable, so that many of their results are inaccessible to direct solution. Simulation would therefore be a means at getting at these results by means of clever approximations and computational power. Winsberg sets out to debunk this myth by showing the extra-theoretical and inter-theoretical resources that simulation-building requires[366]. While simulation does explore theoretical systems, it also steps outside their scope, by drawing on computational methods and syntheses of conceptually incompatible theories in order to achieve its results. This is an extremely pertinent critique, opening the question of simulation to new kinds of questions. But as I have argued here, we can effect a further displacement, in which the frame is shifted from the simulation to practices of simulating, regimes of activity in which such artefacts are realised. Such regimes take wandering paths forged within concrete cultures assembled around them, generating, working through and discarding artefacts such as images as they go.

365 Winsberg, *Science in the Age of Computer Simulation*, 3.
366 Ibid., 26, 73.

Artefacts such as software, simulation data, and images of such data, have been wholly constructed by humans and would thus be entirely within the realm of culture, radically distinct from the natural realm where the unknown, mysterious and contingent reside[367]. But as I have argued, things are not so simple. Software may have been written by human hand, but this does not imply that it is graspable in any straightforward manner, especially not when it is of a high level of complexity. Large software systems exhibit an unruliness that commentators on software engineering long emphasised, and with which programmers have struggled since the first operating systems[368].

Scientific software is an intricate labyrinth, one whose construction and navigation are accomplished by one and the same movement[369]. Working upon it in everyday research practice is a matter of enlisting techniques such as visualisation through which scientists try to understand what it is that they have done when they make a simulation. Research is not simply turned outwards towards the domain of nature, but holds itself in the picture too, in a process of phenomenotechnical self-articulation that follows clear paths as well as negotiating through abrupt discontinuities, where uncertainty is rife and vague suspicions the only guide. Through the image scientists encounter the effects they have brought about, reflect on them, act upon them, an inherent reflexivity in which this kind of research finds its footing. In the space of decision toward the onward path of investigation, what is encountered (in the image) also exerts its own kinds of influences on how and what may be realised in that immanent future, while traversing a field of practice itself defined by a stratigraphy of such tracings, its cultivation the intrinsic historicity of research sites. In Kathleen Stewart's words, a study of these phenomena requires new kinds of scholarly attention:

> “An attention to the matterings, the complex emergent worlds, happening in everyday life. The rhythms of living that are addictive or shifting. The kinds of agency that might or might not add up to something with some kind of intensity or duration. The enigmas and oblique events and background noises that might be barely sensed and yet are compelling”[370].

The image, I said, is among these rhythms a point at which what follows unfolds, emerging out of a reflexive entanglement of the process upon itself, a knot sufficient to grasp the future so to bring it about, and to realise the project of which it was born.

367 Cf. Strathern, ‘Artefacts of History’.

368 Brooks, Jr., ‘No Silver Bullet’.

369 Rheinberger, *Epistemic Things*, 74.

370 Stewart, ‘Atmospheric Attunements’, 445.

7 Code and Writing

7.1 Introduction

We now turn from images to code, to analyse this characteristic element of computational science practice. Computational science has much in common with other kinds of research, but by focussing on code we can embark on a journey towards its heart. The way in is through writing.

In my account of modelling, the site of scientific practice can be described as a material and bodily multiplicity. Investigation works upon and with these materials. Such an account is intuitively suited to systems of experiment, where a plethora of equipment and techniques are very visible, on the lab bench, as it were. The challenge is to show that such an account is equally at home with computational science, in which much of the key equipment is “virtual”, and in which many of the practices are undertaken from the visibly quiet position of sitting at a computer. This chapter aims to tackle the issue head-on by inquiring into the nature of software.

How are we to understand software? What kind of theoretical, philosophical or social scientific approach would be appropriate to its particular mode of existence? These questions are at the forefront of the “software studies” movement, which seeks to establish an interdisciplinary framework through which software can be made a central object of cultural critique[371]. So far, however, software studies has tended to foreground art and media[372], and the particular question of *scientific* software remains to be raised. We can expect that an encounter with the philosophy and sociology of science would be a productive one for this young interdisciplinary field. This chapter concentrates on the diversity of texts in the laboratory. The chapter following this, 'Workability and Habitability', is its companion piece, and builds upon the foundation established here, to turn to look at the materiality of software environments.

After outlining the diversity of texts of the laboratory in section 7.3, and the performative effects of writing in section 7.4, I argue that software in contemporary computational science is of a size and complexity that it cannot be accounted for in the paper medium, that it becomes practically

371 See, for example, Manovich, *The Language of New Media*; Fuller, *Software Studies*.
372 Berry, *The Philosophy of Software*.

irreducible to the public texts of its conventional objectification. I read the move towards open source in scientific software as a move beyond paper, towards a new medium for scientific dissemination, one which profoundly transforms knowledge towards a more practice-centric conception, transmitting not discourse about practice, but the software itself, the material substrate for this kind of research. First, though, we return to a theme from early chapters, to a new rationalism in which inscriptions are thought in terms of an *a priori* brought down to earth.

7.2 Concepts and inscriptions

The great advantage of the term “inscriptions” popularised by science studies has been its levelling power. If we talk about inscriptions, we are not approaching a domain already organised into what is pre-judged to be important and what is not. It is also a useful concept because it defies the intuitive separation between the ideal and the material, to help find the rational in the practical and the material in the intellectual.

> “This mysterious thinking process that seemed to float like an inaccessible ghost over social studies of science at last has flesh and bones and can be thoroughly examined. The mistake before was to oppose heavy matter (or 'large-scale' infrastructures like in the first 'materialist' studies of science) to spiritual, cognitive or thinking processes instead of focussing on the most ubiquitous and lightest of all materials: the written one”[373].

In this vein, Rheinberger has recently proposed that we open up our understanding of writing beyond the formal and “begin to observe and investigate in its epistemic positivity the “economy of the scribble” in the lab”[374]. He is interested in the use of temporary and informal writing surfaces for taking notes, jotting down ideas, doodling and sketching, within the practices of research. To a certain extent these forms of writing can reconcile the finalised and formalised publications that eventually become the officially sanctioned medium for telling the story of research, with the material systems through which that research was accomplished. “They lie *between* the materialities of experimental systems and the conceptual constructs that leave the immediate laboratory context behind in the guise of sanctioned research reports”[375].

Software would sit alongside these scribbles as forms of writing crucial to laboratory practice. Both have been sidelined by an overt emphasis on the public discourse of science. Like scribbles, software can be brought into the open by the levelling effect of a generalised analysis of inscriptions.

The concept of inscriptions, however, carries a risk of which we must be wary, a risk of rendering everything banal, of flattening everything into a flat economy of traces, which can be described but which struggles to support any wider argument or explanation, the danger that science studies collapses into a naïve semiotic empiricism. This was what Collins and Yearley famously dubbed

373 Latour, ‘Give Me a Laboratory and I Will Raise the World’, 162.
374 Rheinberger, *An Epistemology of the Concrete*, 244.
375 Ibid., 245.

“epistemological chicken”. They claimed that “[i]t is no good just talking about inscriptions and immutable mobiles; I won't learn from a No Smoking sign why some people obey it while many others ignore it”[376]. In my analysis, inscriptions are not free roaming materials, but rather situated within the material and bodily multiplicities that make up fields of practice. By placing them in practice, we are able to associate them with the Bachelardian displacement we encountered in chapter 4, 'Reason in Practice' from fixed transcendental *a priori*s to the dynamic *a priority* of technical systems. It is this displacement that helps us grasp inscriptions in their relation to practice.

Ethnographic science studies achieved its greatest results by treating scientific concepts not as ideal entities but as material things, as real inscriptions involved in systems of circulation through the laboratory and beyond. This involved a refusal of the *a priori* nature of analytic epistemology. While epistemologists speak of the conditions for knowledge, before and above the actual work of science, science studies scholars prefer to “follow” the entities that come to be called “facts” and “knowledge” during their actual processes of production and dissemination[377]. However, this distinction between *a priori* philosophy and *a posteriori* science studies fails to capture the full importance of material systems of scientific investigation. Rather than abandoning the *a priori* entirely, we can instead locate it within the technical cultures of research. The *a priori* becomes something else, something existing in the historical unfolding of practice. Science involves the worldly re-production of the conditions of its possibility, in its work with and upon the apparatuses of its practice. Software becomes a key part of these apparatuses, a system of technical objects, which in Rheinberger's words are “the frozen product of former epistemic activity, or historical *aprioris*, to use the language of Edmund Husserl”[378].

Bernard Stiegler expresses this transformation when he points to the disturbance of Kantian created by our recognition of the *a priority* immanent to technical externalisations, these fundamental techno-historical conditions of fields of practice.

> “*A priori* synthetic judgement would be supported by an *'a priori'* prosthetic synthesis -- an *'a priori'* which nevertheless has to remain in inverted commas because, upon closer inspection, the *a priori* of synthetic judgement of consciousness takes place after the event [*après-coup*], after a prosthetic synthesis, and thus *a posteriori* (empirically, it pre-cedes this consciousness in time as the possibility of its already-there). But at the same time it also partakes in the *a priori* of the synthesis of judgement that it only makes possible -- in a somewhat mythical, performative and foundational *après-coup* -- and which, *in being a precondition for any possible experience based on recognition, is 'transcendental', even though it only exists under the* a posteriori *conditions imposed by the history of technical inventions*.”[379]

376 Collins and Yearley, ‘Epistemological Chicken’, 318.

377 Pickering, ‘From Science as Knowledge to Science as Practice’, 2.

378 Rheinberger, ‘Reply to Bloor’, 409; cf. Thrift, *Non-Representational Theory*, 10.

379 Stiegler, ‘Our Ailing Educational Institutions’, np; see also MacKenzie, *Mechanizing Proof*, 334 for a discussion of the relevance of the concept of prosthetics for talking about computers in science.

Bachelard's philosophy provides a good ally, not just because he too wished to bring *a priori* categories “down to earth”, but because he asserted that scientific concepts never simply exist in abstraction from experimental practice. They must always be put to work. “A concept becomes scientific in so far as it becomes a technique, in so far as it is accompanied by a technique that realises”[380]. In studying software within scientific research we are studying concepts that are put into practice: concepts realised in code. We are not passively following a web of inscriptions, as it mutates and evolves. We are studying forms of writing and practices of writing as aspects of technical systems that set up and set forth a world in which scientific work gets done, and in which the very question of what it means to accomplish actions as a practising scientist is given its primary reality. In other words, while a theory of inscriptions is a key ally, there is no reason to suppose that we need be drawn into a banal empiricism. We can have a strong theory of technical-practical systems as the very site of reason, one significant substrate of which is made up of the interleaving of many kinds of texts.

7.3 The many texts of the laboratory

Before zooming in on the question of software in particular, I want to situate it within the sheer diversity of texts that circulate within the research group. We will discover that software is intimately linked to many of these. It is already anticipated throughout the technical culture. Any appreciation of software must begin from these arrays of texts, because it is only among these that the question of precisely what is to count as software, and as relevant to its operationalisation in practice, can be asked.

We can start with scientific publications. Those produced by the group as well as those produced elsewhere circulate around the research group, on paper or in electronic formats. There are research papers in journals, PhD theses, and textbooks for reference. Many potential publications created in the group spend considerable time in draft form. All have multiple authors and are discussed and commented upon individually and in groups. Other texts also abound, such as conference papers, and texts for lectures. After having been displayed at conferences, posters showing the results of particular research projects are often put up on the walls of the corridors and of the lecture and seminar rooms throughout the college.

The comments we find on the texts of draft papers are most often made digitally, but sometimes simply scrawled in the margins of a print-out. Similarly, many scientists take notes on publications that they read, on loose paper, on computer, or on post-it notes peppering the scientific textbooks that are found throughout the offices. Most scientists would take notes in meetings on pads of A4 paper, which most often turn out to be mixtures of “to do” lists and sketchy graphs, pictures and formulae. These are “messy” pads, and it was rare to find anyone keeping a coherent “journal” of research. When explaining things to each other, they often used the whiteboards which are ubiquitous

380 Bachelard, *The Formation of the Scientific Mind*, 70.

throughout the offices. Explanations in these informal settings usually rely on the use of such externalisations because it is helpful, even when talking about the most well known mathematical formulae, to be able to point at particular terms within the expression, and to draw graphs to illustrate the point. The A4 pads of paper that are carried around to meetings are commonly swapped around at such moments and whoever is attempting to explain the concept will draw the illustration on the other person's pad, or else pieces of paper are torn off and exchanged.

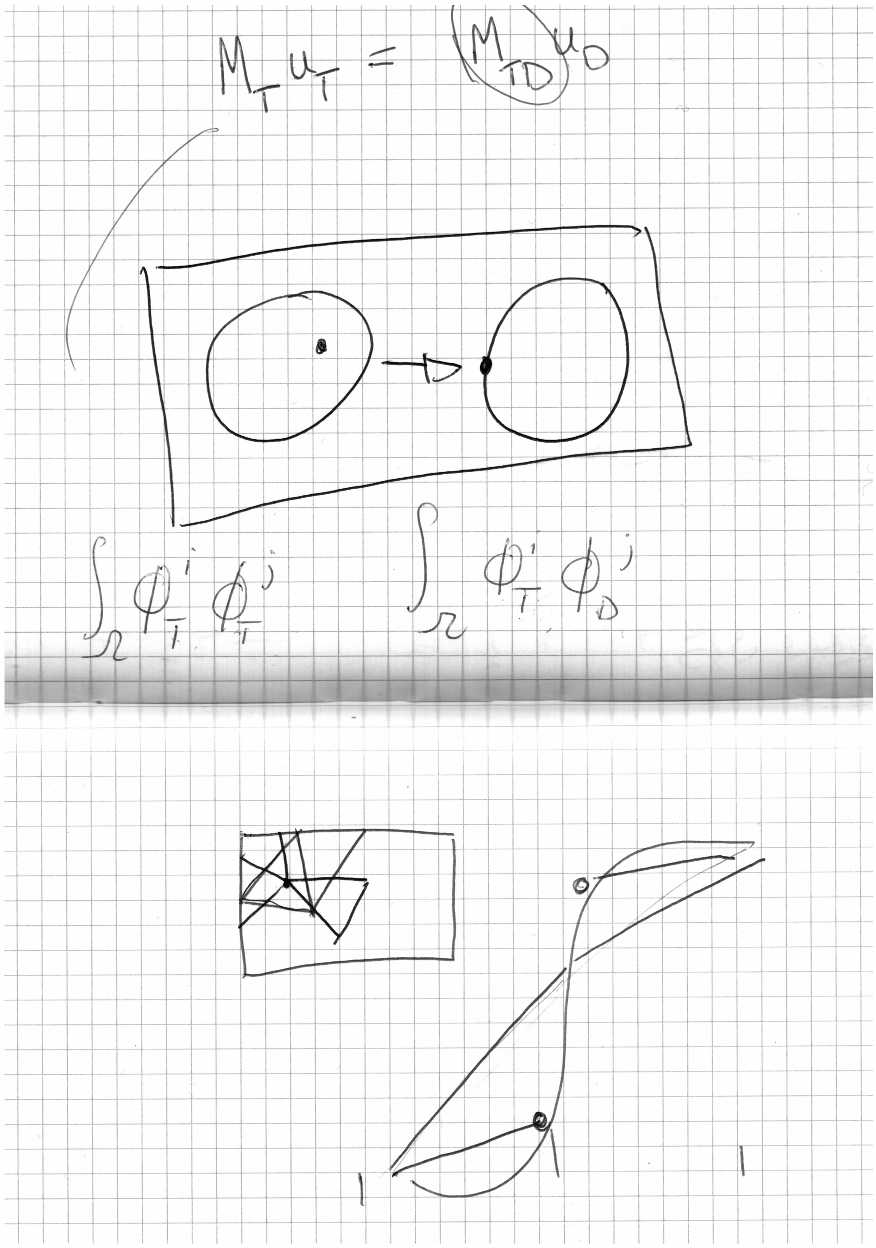


Figure 11: Typical scribbles, sketched in order to facilitate explanation of the problem of the "supermesh" (top) and of discontinuous elements (bottom)

Drawings and writings such as those in figure 11 emerge alongside laboratory talk, and I don't find it helpful to isolate this kind of discourse from the gestures that accompany it. The chatter of the laboratory is never purified in the way that publications are, through their rigid conventions of composition and format. While it is easy to point out that marks such as these have a durability that is lacking in the case of speech, these scribbles would rarely last more than a day or so. Those written on paper are discarded at a similar rate to that at which new ones are generated, while those on the whiteboard get confused, mixed in with other sketches and writings, and eventually erased whenever someone desires a clean slate with which to explain something new.

Members of AMCG email each other privately where it concerns private meetings and documents, but most digital communication happens in the more public media of email lists and the group's chat channel. Email lists are used to inform everyone of upcoming events and of technical changes (e.g. server downtime, upgrades to operating systems and so on), to ask for help, as well as to respond to questions. They are also used to distribute details of changes to source code and comments produced by the review board system. General technical support is also provided on this channel, especially for outside users who have downloaded the group's open source systems. It is common to see queries that were originally sent to a single individual being bounced into the list, either because responses will be quicker when the question is received by the whole community, or because many of these queries and their answers are potentially interesting for the wider readership.

The group runs a chat channel that most scientists log onto in the background throughout their day. This medium is used more informally than email, for co-ordinating lunches and coffee breaks as much as for meetings, as well as for locating members of staff ("Is X person in the office yet? Can I come and see you about Y issue some time this afternoon?"). But they are also used if a scientist wants to get a quick response to a plea for help on a technical question. As opposed to the email list support, this kind of support is generally between group members. Chat has the advantage of providing quick responses, although sometimes the format makes the issue or solution hard to follow (sometimes several conversations are on the go at once). The automated testing system is directly hooked in to the chat, so conversations are occasionally interrupted by a message about a failed test, which sometimes provokes comment.

The following is a discussion in the chat of a problem in a small section of Python code:

```
CE 11:01
how do I fix this:
x(:,1) = x0 + dt*(/0.,sin(x0(2)) /) + sqrt(2*k)*dW(:,1) 11:01
1 11:01
Error: Element in REAL(4) array constructor at (1) is REAL(8) 11:01
all my variables are declared real*8 11:01

WS 11:02
A. don't use real*8, it's nonstandard

CE 11:02
```

what should I use instead?

(this is in some Fortran compiled with f2py) 11:02

WS 11:03

The cleanest way to do this is to declare a parameter, say D for double or P for precision

you then define the parameter either directly to 4 or 8 relying on the convention that kind types are the byte count 11:03

or you use the selected_real_kind intrinsic to do it the official way 11:03

then you can declare your variables real(P) 11:04

CE 11:04

you mean

WS 11:04

and in those cases where precision really counts, such as in array constructors, you can append _D to the literal

CE 11:04

integer, parameter :: P = selected_real_kind

? 11:04

WS 11:05

eg 0._D

for a double precision 0. 11:05

CE 11:05

oh, it's the 0. that's the problem is it>

WS 11:05

almost. selected_real_kind is a funciton

CE 11:05

?

AX has disconnected (Remote host closed the connection) 11:05

WS 11:05

the direct problem you're having is that that's a single precision 0.

since it comes first in the array constructor, that sets the kind of the array 11:06

CE 11:06

I see

WS 11:06

then you put sin(x0(2)) in and that's a real(8)

```
which it doesn't like 11:06

CE 11:06

how come we don't have to do all this stuff in Fluidity?

WS 11:06

because fluidity cheats

fluidity does everything in default precision and relies on compiler extensions to change the
default 11:06

there are a few places in fluidity where this matters. Generally because of interfaces to
external libraries. 11:07

CE 11:07

ok

so, I'm probably declaring P in the wrong place 11:07

I have a file containing subroutines which gets compiled by f2py 11:08

and I did 11:08

subroutine shearflow(x,x0,dW,dt,k,n) integer, parameter :: P = selected_real_kind() real(P),
intent(out), dimension(2,n) :: x 11:08

... 11:08

and got: 11:08

real(P), intent(in), dimension(2,n) :: dW 11:08

1 11:08

Error: Parameter 'p' at (1) has not been declared or is a variable, which does not reduce to
a constant expression 11:08

AX has joined the room 11:08

WS 11:08

actually, you almost certainly have an error above that

CE 11:08

does P need to be declared outside of the subroutine declaration?

oh, you are right 11:09

WS 11:09

didn't your mother tell you to always fix the first error first

? 11:09
```

You can see here how quick a response it is possible to get. Provided willing respondents are logged in to the channel, help is forthcoming in real time. As opposed to email lists, the chat has a “live” feel, the interactivity of the medium helping WS and CE work through their mutual misunderstandings, towards some kind of resolution. Fragments of code here find themselves transferred into new electronic formats, generating a consequent cascade of other kinds of text.

Another example of more informal comments, interspersed with a failing test, from slightly later on the same day:

```
CE 12:21

lunchtime?


YN 12:23

aye


amcg-data 12:23

running failed [82 seconds],: ; No blamelist; http://buildbot-
ocean.ese.ic.ac.uk:8080/builders/test-sphere-3D-drag-Re1-cx1/builds/651


WS 12:23

Right-ho. Anything is better than reading the random grammar of this MSci lit review


CE 12:24

a lit review on random grammar would be interesting

I'll see you outside the SCR imminently 12:24

QH has joined the room 12:24


QH 12:25

hello all from my shiny new laptop!


amcg-data 12:25

Hello, QH!


CE 12:25

yay!


SS 12:25

GY: could you put the output from lock_exchange_3d_parallel on Buildbot on /scratch for me to
look at please?

CE trundles down to SCR 12:26


TT 12:27

does anyone know of a way to get the college email archiving system to play nice with
anything other than outlook?

QH keeps forgetting that the mouse is attached to the other laptop 12:29


HP 12:30

did QN not mantis an issue at some point where the intel build would crash in the mainfl_fc()
call?

oh no, that was gfortran 12:31

QH laments the loss of her apple key and tries to get used to ctrl-alt instead 12:33
```

> KU has joined the room 12:35
>
> KU 12:35
>
> have people gone for lunch?
>
> KU is at an openmp training course in aero 12:35
>
> HU 12:35
>
> WS has gone, I think 5mins ago?
>
> KU 12:36
>
> ok
>
> thanks 12:36
>
> KD has joined the room 12:49
>
> amcg-data 12:55
>
> running failed [62 seconds],: ; No blamelist; http://buildbot-ocean.ese.ic.ac.uk:8080/builders/test-circular-duct-from-rest-cx1/builds/1057
>
> amcg-data 13:05
>
> running failed [362 seconds],: ; Blamelist: KD; http://buildbot-ocean.ese.ic.ac.uk:8080/builders/test-water-collapse-2d-valgrind-cx1/builds/1316
>
> KU has disconnected (Quit: leaving) 13:22

The majority of AMCG members work in the same buildings, but there are several floors, offices and sections. Some are based in different buildings on the same site, while others are based in different cities. The chat gives a constant sense of bustle, a connection to the being-together of the community, which could easily be downplayed if the relative quiet of the working environment was the only thing ethnographically observed.

In addition to the automatically generated notices of failed tests are the manually submitted, monitored and assigned bug reports, which describe problems with a view to keeping track of their various manifestations and their seriousness in order to help the appropriate person find a way to sort them out.

> **Meshes of 2D topology in 3D space are not correctly partitioned by fldecomp.**
>
> Fluidity Bugs
>
> Bug #929690
>
> Reported by AB on 2012-02-09
>
> Bug Description
>
> When supplying topologically 2D meshes on a geoid, to be extruded in the vertical direction within Fluidity, the following error-message appears:

*** FLUIDITY ERROR ***

Source location: (Quadrature.F90, 282)

Error message: make_quadrature: 3 is an unsupported vertex count.

The fldecomp help suggests using the '-s' option, to indicate that the input mesh is a spherical shell mesh. However it does not fix the problem. To date, the problem is circumvented by using a mesh in the triangle format and that use an awk script to edit the .node file:

```
sed '1s/\<3\>/2/' < mesh.node > modified_mesh.node

mv modified_mesh.node mesh.node
```

Thus when using meshes on a sphere, one can only use triangle-format meshes, along with the fix above. Binary Gmsh meshes cannot be used.

Related branches

lp:~fluidity-core/fluidity/gmsh-on-sphere

Ready for review for merging into lp:fluidity

QS: Pending requested 2012-03-26

Diff: 2132 lines (+909/-758) 17 files modified

AB on 2012-02-09

Changed in fluidity:

assignee: nobody → AB

QS wrote on 2012-02-23: #1

AB: can you upload an example FLML, mesh file and makefile (or script) that shows this, please? I can then mark it as confirmed.

Changed in fluidity:

importance: Undecided → Low

QS wrote on 2012-02-23: #2

Discussion in the dev meeting was to try flredecomp instead of fldecomp. Awaiting results of this test.

QS on 2012-02-28

Changed in fluidity:

status: New → Confirmed

importance: Low → Medium

HU wrote on 2012-02-28: #3

Related to this: I found the gmsh reader does not handle the spherical+extrusion case correctly even in serial. So this might not (only) be a fldecomp issue. To reproduce, see the new sample_netcdf_test/ on the lp:~fluidity-core/fluidity/fluidity-initialisation-from_netcdf branch and switch the input to gmsh.

```
AB wrote on 2012-02-29:         #4

    sps_6G4.1_local_bin_gmsh.tar (100.0 KiB, application/x-tar)

I attach a test case, the mesh is a rectangular area on a sphere, extruded in the vertical
direction within fluidity. The makefiles in the top-most directory and the mesh directory
will generate the mesh, decompose it and run fluidity. The third line of the Makefile in the
top most directory must be changed, and the 4th line in mesh/Makefile will have to be changed
if one needs to use the makefile in the mesh directory separately (make input in the top-most
directory should use the makefile in mesh/, with all the right variables). Then 'make
NPROCS=x' should run the case. There is also a run_gdb target, so 'make run_gdb NPROCS=x'
will run fluidity in parallel under gdb on x processors.

My apologies for the svn files (including hidden svn files)


AB wrote on 2012-02-29:         #5

Following discussion in the last dev meeting flredecomp has been tried as well, but it fails
with a zoltan error, I will be posting a separate bug report with more details and a test
case.


HU wrote on 2012-03-21:         #6

The issue is not so much with fldecomp, which seems to be doing the right thing with -s
option, but with the gmsh reader in general. This also doesn't work in serial, i.e. the gmsh
reader used by fluidity doesn't handle a 2d spherical input mesh (mesh%dim=2 but 3 coordinate
dimensions).

This is fixed in the gmsh-on-sphere branch. All it needs is a test which I'm working on.
(currently there are no short tests on the sphere at all!)

Changed in fluidity:

assignee:      AB → HU
```

The discussions of bugs that we find alongside bug reports are generally more formal than those on the chat channel. There is a sense of creating an archive, of actions, questions and ideas, brought together using this software system (the group used to use “Mantis”, but transferred to Launchpad's integrated bug reporting system in 2011). Given that many bugs will not be solved immediately, bug reporting creates a database of known issues, to which new information can be added, priorities set and responsibilities assigned.

The software itself is accompanied with several layers of text. New commits are always commented on, something quite strongly policed[381]. It is considered very bad manners to make a change to the code without a note saying what the change is and why it has been done. These are distributed to the lists with appropriate metadata automatically attached (e.g. version number, time and date, file name). The review board system that was set up in early 2011 also enabled comments to be made by other users. When a commit was made, it was queued up for an appropriate reviewer to check the quality of the code, and many of the comments made at this point refer to the style of the software writing (e.g. naming practices for variables) as well as for more substantive issues such as the management of memory.

A typical example of comments on a commit:

```
Author: QS

Date: Mon Nov 29 10:19:51 2010
```

381 A “commit” is a change made to the code base.

> New Revision: 15049
>
> URL: http://amcg.ese.ic.ac.uk/cgi-bin/viewvc.cgi?view=rev&root=fluidity&revision=15049
>
> Log:
>
> Adding new option to adaptivity to limit node change, rather than just maximum increase as with the existing options. No tests as yet. Doesn't make the slightest difference to the spikiness during adapts in my tests, but hey, someone might find it useful. Also adding a limit_metric call for the vertical adaptivity. The limit applies to each vertical column. Current limits only applied to the surface mesh
>
> Modified:
>
> trunk/assemble/Adapt_State.F90
>
> trunk/error_measures/Assemble_metric.F90
>
> trunk/error_measures/Limit_metric.F90
>
> trunk/schemas/adaptivity_options.rnc
>
> trunk/schemas/adaptivity_options.rng

Another typical example, using the comments on commits as a point of reference for general coding standards:

> Author: XC
>
> Date: Fri Dec 3 13:06:04 2010
>
> New Revision: 15099
>
> URL: http://amcg.ese.ic.ac.uk/cgi-bin/viewvc.cgi?view=rev&root=fluidity&revision=15099
>
> Log:
>
> A little tidying up of Momentum_CG.F90 as some coding sins were being committed making it extremely hard to add new functionality.
>
> Some basic coding standards for the future (these apply beyond Momentum_CG):
>
> - Arguments to subroutines should not be declared as optional if all instances of calls to that subroutine have that argument. In the case of Momentum_CG there were several subroutines that are only called from one place in the entirity of fluidity and yet they had optional arguments declared for no reason. This meant that adding new arguments to the subroutine was difficult and I think possibly led to people adding extra superfluous optional arguments, which leads to the next point...
>
> - If optional arguments exist in a subroutine call and a new required argument has to be added, it should be added before the optional arguments as a required argument. It should not be added at the end of the argument list and declared as optional (as it is required).
>
> - If an optional argument is really required (i.e. there is more than one call to that subroutine, one with the argument, one without) then a check should be made that it is present before using it.
>
> - New arguments to subroutines should be declared at the start of the subroutine - not interspersed amongst the local variable declarations. Furthermore they should be added in the same order they appear in the subroutine call.
>
> - Within the assembly routines a convention has arisen that logicals and parameters turning on and off options within the assembly loops can be declared as private module variables, which can be accessed from any subroutine within the module. This reduces the problem of having to pass every parameter through every subroutine in the element loops. However it should only be done for logicals and parameters that do not change during assembly. Furthermore they should not generally be made public.
>
> I'm anticipating that intel debugging on cx1 might have a problem with this commit. I will keep an eye on the relevent test cases and update as necessary.

```
Modified:
   trunk/assemble/Momentum_CG.F90
```

In addition to the comments attached to the commit and to the review, the code itself is full of comments which describe what the code is doing (or supposed to be doing) at any one time. When part of the software is in the process of being written, and is not yet ready, these comments often include many “notes to self” to help the programmer keep track of what he or she is thinking while writing. When the code is working properly, and ready for general use, comments are edited so they address a general reader, rather than being specific to their own particular thought processes.

Comments within the code are marked as such at the start of the line (in the case of Fortran, by two exclamation marks). This instructs the compiler to ignore that line. It is intended solely for the reader of the code. For example:

```
  subroutine construct_advection_diffusion_dg(big_m, rhs, field_name,&
   & state, mass, diffusion_m, diffusion_rhs, semidiscrete, velocity_name)
!!< Construct the advection_diffusion equation for discontinuous elements in
!!< acceleration form.
!!<
!!< If mass is provided then the mass matrix is not added into big_m or
!!< rhs. It is instead returned as mass. This may be useful for testing
!!< or for solving equations otherwise than in acceleration form.
!!<
!!< If diffusion_m and diffusion_rhs are provided then the diffustion
!!< terms are placed here instead of in big_m and rhs
!!<
!!< If semidiscrete is present and true then the semidiscrete matrices
!!< are formed. This is accomplished by locally setting theta to 1.0
!!< and only inserting boundary conditions in the right hand side.
!!< Setting semidiscrete to 1 probably only makes sense if a separate
!!< mass matrix is also provided.

!! Main advection_diffusion matrix.
type(csr_matrix), intent(inout) :: big_m
!! Right hand side vector.
type(scalar_field), intent(inout) :: rhs

!! Name of the field to be advected.
character(len=*), intent(in) :: field_name
!! Collection of fields defining system state.
type(state_type), intent(inout) :: state
!! Optional separate mass matrix.
type(csr_matrix), intent(inout), optional :: mass
```

```
!! Optional separate diffusion matrix
type(csr_matrix), intent(inout), optional :: diffusion_m
!! Corresponding right hand side vector
type(scalar_field), intent(inout), optional :: diffusion_rhs
!! Optional velocity name
character(len = *), intent(in), optional :: velocity_name
```

At the top of this excerpt, you can see a general explanation of how the code works, and below it is a line-by-line labelling providing more fine-grained description. While it would be possible, in theory, for new developers to work all this out by reading the code, without reading any comments, these reading processes are greatly accelerated by the additional layers of text. The need for new developers to start work quickly, for scientists with little coding experience to learn to program, and for the code to be constantly modified while being kept consistent, has thus resulted in the profusion of commentary throughout its structure.

A final layer of text that surrounds the code is the documentation. In a sense, all the comments are documentation, but the central point of reference is the user guide, which is kept up to date alongside the software itself. The rule of thumb is that if anyone changes the code, they must ensure that the relevant changes are made to the user's guide. Actually, this is often let slip, and the annual or biannual official releases of the code are times for a comprehensive policing of any parts of the user's guide that might not be up to date.

The code itself is far from being a homogeneous text. The majority is currently written in Fortran 90, a language commonly used by mathematicians and engineers. It is organised into a structure of documents and folders. However, many of the functions that it uses are found in external libraries, libraries for parallelisation or for solvers, for example. Some of these are written by members of AMCG but many are written by outsiders with nothing to do with the group. This interfacing or overlapping of software systems with each other is ubiquitous. If a new solver method is needed for a particular kind of oceans problem, for example, there is no point in writing this method into the Fluidity structure. Instead, you write this method into the general library of solvers and have Fluidity use it through its interface with the library. This keeps things organised but also enables other programmers to use that method in other software projects.

Fluidity is modular, written according to the principles of object-oriented programming. It is also modular at a higher level, with its components designed to be as general purpose as possible. When writing a new input system for specifying options and parameters, the AMCG scientists decided not to make a system specifically for Fluidity, but rather to make a completely separate programme, that can itself be configured to manage any input set whatsoever. The idea is that it can be used by other scientific models too, something that will have the advantage of widening the community of users who have an investment in keeping it maintained and up to date. This system, called Diamond, therefore has its own documentation, its own source code, its own comments. It also requires further texts, because it requires a “schema” to tell it what the model is that it is to be used for, and what parameters and options that model has. Diamond has a front end, called “Spud”, which displays all these options in a visually intuitive interface, with a tree structure. The schema specifies dependencies so that Spud can indicate if an option is not set which should be set, and has a comprehensive system of commenting so that every parameter is displayed with a short section of text explaining what that parameter is needed for and any extra information that would be needed by a user. For a user who is not involved in writing new functionality, Spud will provide the main interface for working with Fluidity, accompanied by the command line for executing the code, and visualisation and graph-generating suites for interpreting the output data files.

One of the most significant differences is between “compiled” code and “interpreted” code. The bulk of Fluidity source code is compiled code, and is therefore compiled into an executable binary file before being run. In contrast, interpreted code is effectively compiled at run-time. Some parts of Fluidity are interpreted, and are written in the Python language, a general purpose programming language used in many different fields of software development. The majority of Fluidity is compiled in order to achieve maximum efficiency. It also works well for the normal user work flow. When a binary is obtained, it can be used to generate many simulations with different initial conditions. It is compiled, so the core functionality cannot be changed without recompiling, but all the options can still be modified through Spud/Diamond. If, on the other hand, the project involves making changes to a more fundamental feature of how Fluidity works, it would be necessary to regularly re-compile a new binary in order to see the results. Exactly what kind of software is dealt with on a daily basis, and how it is encountered, will thus depend on the kind of research that is being done, and on what stage the project is currently at. QH commented in an interview that:

> “at the moment, most of my time is taken up with running simulations. A year ago, though, it was very different. Then I was developing the software to create these simulations.”

Different kinds of code are nested together. Fluidity runs sections of Python within it. Python is most commonly used within the options system to set a parameter or boundary condition as a mathematical function rather than as a constant. If, for example, you want a flow through a channel to vary in its velocity according to some quadratic oscillation, you can write that function in Python, and feed it (through Spud/Diamond) into Fluidity, which will calculate its value for each time step. Interpreted codes are generally called “high level”, as opposed to “low level” compiled codes like Fortran. In choosing between the two for a given aspect of a model, the trade-off is between the speed of programming and the speed of the program. When I queried this balance at a training event, WS explained that where it is a critical part of the simulation, Fortran is used, but where it is a non-critical part, and where it would be handy to be able to quickly and effectively read and modify the script on a regular basis, Python becomes the obvious choice. Further language diversity will be discussed towards the end of the next chapter, which outlines a current project to add another high level language for the formulation of the basic equations.

It should be quite clear, therefore, that 'Fluidity' cannot be treated as a single text. It is many, some of which are written in different languages, and some of which are to a certain extent separate programs or libraries written and maintained by different communities. Software systems are more like networks with mobile boundaries rather than bounded Cartesian objects. Their identity is topological and dynamic rather than geometric.

The next section will investigate the practical dimension of code, to ask whether, among texts, code is something special. The clearest manner in which code might be special is the sense in which we speak of it “running”.

7.4 Putting text to work

One of the major theoretical shifts in Twentieth Century philosophy of language saw the displacement of the assumption that the important thing about speech or text is that it says something, to allow space for thinking about the many aspects in which speech or text *does* something. From Austin's *How To Do Things With Words*, to Foucault and Butler's theoretical texts, *performativity* became a hot topic[382]. Whether you are interested in the discursive creation of subjectivities or in declarations and naming practices, it is no longer possible to take for granted that language is simply about communicating meanings. Speaking, writing, or manipulating texts in other ways, are all real actions that change the circumstance in which they are performed. This paradigm would seem to be strongly embodied in software, which is surely the epitome of text for which its ability to do things, to be transformed into real processes of computation, is never hidden behind its semantics, behind what the meaning of what it says[383].

I want to argue, however, that this very division, between saying and doing, is not very helpful in understanding what happens when software is put to work. I will argue that it is impossible to regard code only in terms of the physical processes of computation, but that to thus displace its inherent operationality is not to revert back to a passive theory of language. The best way to capture the nature of code is to regard both its operationality and its interpretability as different ways in which it is approached and handled in practice[384]. As an object, code is reducible to neither aspect, but features in practice in both ways, depending on the circumstance.

Code “does things”, but only insofar as it is embedded in a concrete field of practice, insofar as it is already enfolded in other goings on, in projects and endeavours that have many non-digital aspects to them as well. The “doing” in question in the performativity of code cannot therefore be simply that of the brute reality of it computation, but must extend to the human aspects of its being part of situations of research. To claim this is not to claim that it is “socially constructed”, but rather to point

382 Austin, *How to Do Things with Words*; Butler, *Excitable Speech*.
383 Cf. MacKenzie, *Cutting Code*, 90.
384 Ibid., 5 makes a similar point.

to the fact that its real effects as part of a computational system exist as part of concrete human activities.

Computational scientists at AMCG confront the twin possibilities for approaching their code on a daily basis, in the need to balance formal and empirical measures of their simulations. We encountered this in section 2.12. Mathematical results about discretisations and methods treat the code as a text, as a text in formal language which can be interpreted, analysed, and subjected to mathematical techniques of proof. On the other hand are the empirical investigations of the simulations they produce, their behaviour under different initial conditions, and the relationship of sets of output data with validation data. These investigations are grounded in the software's actual running. They treat it as a real electronic process which can be subjected to empirical techniques of manipulation and measurement. There is no shared ultimate ground upon which formal and empirical treatments get combined, no essence to code that unites them. To see their commonality we have to look outwards towards the research practice in which they are found, and this takes us into a sphere of multiple techniques, materials, bodies, competencies and actions.

This duality of approaches to code in practice is reflected in long traditions of computational science that flag up either side as the “real” essence of software. I would prefer to shift emphasis and take these disagreements as a sign of the real elusiveness of any such essence. Tony Hoare, for example, claimed that “[c]omputer programming is an exact science, in that all the properties of a program and all the consequences of executing it can, in principle, be found out from the text of the program itself by means of purely deductive reasoning”[385]. If this were the case, then it would be possible to initiate a pure study of software which would discover what it does from what it means, from the text alone. It wouldn't really be necessary to run the software; the whole context of practice could be disregarded. But no matter how many formal proofs are deployed in the verification of software, and no matter how influential they are, they cannot erase the possibility of pointing out that the computer itself is an empirical entity subject to all the limitations and exigencies that characterise our dealings with other such things. Its electronic occurrences may be amenable to formalisation, but such formalisation requires a mathematical treatment that is itself an empirical practice, and does not apodictically follow from these events. James Fetzer states the converse point of view:

> “These limitations arise from the character of computers as complex causal systems whose behaviour, in principle, can only be known with the uncertainty that attends empirical knowledge as opposed to the certainty that attends specific kinds of mathematical demonstrations. For when the domain of entities that is thereby described consists of purely abstract entities, conclusive absolute verifications are possible; but when the domains of entities that is thereby described consists of non-abstract physical entities... only inconclusive relative verifications are possible.”[386]

385 Quoted in Franklin, ‘The Formal Sciences Discover the Philosophers’ Stone’, 527.
386 Quoted in ibid. NB: The verification Fetzer mentions is the computer scientists’ sense. See section 2.12.

While it is easy to treat computation as logically determinate, this determinacy is the property of an abstract representation (a very effective one) of the computation, and can never be applied to the actual computation without empirical caveats.

There is another issue that further entangles the two treatments of software. When software is run on a computer, electronic events ensue, and produce output data as a result. This relationship can easily appear to be a one-way determination. The software determines the computation. The software, and formal treatments of it, would be prior to the empirical occurrence and the traces it leaves on disk. But in practice, even after the running of the programme, the software continues to play an ineliminable role by conditioning the interpretation of that data. It is only by bringing that software into articulation with its computational output that the latter can become meaningful for research.

There is a forward-backward asymmetry that requires the text and the process to be combined at every point. Whether you are given a computational process, its data output, or even a ready-compiled, ready to run binary file, none of these things can be understood in their formal structure without access to the source code that generated them. This is the only human readable text involved, and it is only in its association with these materials that it confers intelligibility upon them. As opposed to the interactive software of desktop interfaces, in which meaning is created “on the fly”, within the running of programs, computational science requires interpretation of computational processes in terms of the mathematics embodied in their actual running. For this it is always necessary to already know what code has been used. Reverse engineering the binary file by looking at the computational process can produce a massive number of different possible binaries, each of which in theory would generate that computation. Reverse engineering the source code by looking at the binary would similarly yield a huge number of possible texts, the vast majority of which will be completely unintelligible. This property is relied on by commercial software vendors to discourage customers or competitors from reverse-engineering their products. Finally, most simulations do not write to disk anything like enough data to reconstruct the computational process that created it. For each step to be recorded it would be necessary to make modifications to the source code and set-up, and this then raises the question of whether it would still be the same computation. Just as the code cannot be fully understood without being run, the running of the code cannot be meaningful without a relation being maintained to the texts of its origin.

We thus can understand computer modelling in terms of a two-fold structure. The code can be run, but once it is run, the transformation undergone in its execution gives the computation itself an independence from its origin. It is severed from its origin. There is no straightforward way of identifying a computation such as a simulation with the source code. We thus find two very different sets of practices at work together: empirical treatments in which the simulation is treated like an experimenter's apparatus, and formal treatments in which the code is analysed as a text. To understand this twofold it is necessary to refuse the temptation to look for the essence of computational science in code or in computation. We can only grasp their relation if we concentrate on another frame: that of research practice itself. From this point of view, we grasp formal and empirical treatments as activities, and we also start to see the composite whole, including bodies and materials, things which usually seem very foreign to software. It is this broader frame that gives us some grasp of the significance of software in practice.

7.5 Software in practice

Once we start to look at the systems of investigation through which scientists work on and with software, we start to see that there is more to code than its empirical execution. Dexter et al., for example, point to the need to understand the ways in which processes of working with software draw on the most complex imaginative, embodied and metaphorical resources.

> “Code may appear to some to be among the *most* 'linguaform and propositional' modes of contemporary human expression and, thus, completely unsuitable for attaching completely different forms of meaning. But... the development of modern programming depends absolutely on a complex scaffolding of metaphor and non-propositional meaning drawn from the roots of embodied human experience.”[387]

Software is a question of practice more than it is a question of language. The most important thing is not what *software* “does”, because in approaching the issue in that way we make an untenable isolation between the computation and the environment in which it is put to work. The most important thing is what *the practice of working with software “*does”. We have seen, in previous chapters, how scientific objects emerge in this sphere, in the iterative processes of writing and revising software, in exploring its consequences, testing it and opening it out to new possibilities.

We are talking about texts as real things, immersed in systems of practice. What is important is not what is “said” about a physical system by representing it in a simulation. What is important is the effect of folding it into a practice of computer modelling, aligning it within the activities of building and manipulating that are developed within this frame, as possibilities of exploration. We are interested in manipulation, not denotation. As Rheinberger puts it:

> “The whole thrust of my argument lies in the assumption that the primary way of symbol-making in the realm of scientific activity is itself a material process and not linguistic, that the epistemic semiosis is one of traces that we relate to invisible entities, and not between names and things. Handling a virus *as* a gene, that is, on the model of a gene, can take for instance the experimental form of trying to mutate the building blocks of its nucleic acids... Handling the virus as a chemical molecule may take the form of trying to crystallize it. All these investments in turn, and as a rule, may lead to changes in what a virus is understood to be. The concept is nested into the exploration of the epistemic thing.”[388]

KU expressed a similar sentiment common among the scientists at AMCG when he commented that:

387 Dexter et al., ‘On the Embodied Aesthetics of Code’, 16.

388 Rheinberger, ‘Reply to Bloor’, 408.

> "you think you know something when you learn it. And then when you go to teach it you find out you didn't really understand it. But then when you go to code it, that's when you really realise that you didn't understand it. When you can code it up and it works that is when you really know what you are doing."

The practice of writing software is not a matter of putting down in a new language a set of concepts already known in another language. It is a materially and bodily mediated process which creates new understandings. This was the core principle of phenomenotechnique, that concepts in science do not properly exist outside of their putting into practice. This field of practice is a space for creative and transformative processes. In chapter 8 we will look at the work involved in creating and maintaining an environment for computational science which provides a crucial quotient of freedom, which we can think of as a space for playing around, challenging preconceptions, and creating new things. At this point, it is sufficient only to note that there is nothing straightforward about working with software. It is unruly and bug-prone. Its freedom is that of skilled craft, not of an idealised space of pure thought. Scientific ideas, insofar as they are put into software practices, require practical competencies as much as does any other laboratory practice.

> "In order to program, you have to understand something so well that you can explain it to something as stonily stupid as a computer. While there is some painful truth in this, programming is also the result of a live process of engagement between thinking with and working on materials and the problem space that emerges. Intelligence arises out of interaction and the interaction of computational and networked digital media with other forms of life conjugate new forms of intelligence and new requirements for intelligence to unfold."[389]

To appreciate this "live process of engagement", we have to regard software not just as the thing being made, but as also forming the environment in which the practice takes place. Work with software is work on an edifice, a generative structure embodying, in KU's words, "terabytes of implicit knowledge". Fluidity is the concretisation of countless past projects, investigations and explorations undertaken by members of the group. It is the material embodiment of their research. In this sense it *is* their research, as much as it *stands for* their research. As such it inspires strong feelings of duty, care, and respect, towards both the group and the code, which underlie the concept of "best practice". Whatever the content of these stipulations of best practice, they gain their force and efficacy from embodied attachments to software as the site in which research gets done, structures of care, built through the past and formative of whatever successes are to emerge in the future.

We can therefore appreciate that scientific software is always more than simply operationalised theoretical concepts, mathematical theory turned into a new productive machine. It is more than a "reified theory" (see chapter 4), or it is a reified theory that teaches us that we never engaged properly with theory until putting it into practice, and that once it is there it can never be extracted

389 Fuller, 'The Stuff of Software', 10.

and purified from this material backdrop, this irreducible scaffold of computational science research. As Winsberg puts it, “these models are best viewed not as mere solutions to theoretical equations; they are rich, physical constructs that mediate between our theories and the world”[390].

I have argued that there is no sense in regarding software simply as a system of language, that it must be seen in the terms of the systems of practice in which it is embedded. In the next, final section, I want to take this further, and explore the potential of software to displace natural language and “public sphere” discourse from our conception of scientific research. My thesis is that if we understand the relationship of software to language within the context of its use in research, we will no longer be able to unthinkingly give epistemic priority to the kind of propositional knowledge that has dominated our understanding of epistemology for several centuries.

7.6 Epistemology beyond the paper medium

One of the highest stakes of science studies is the displacement of the kind of propositional knowledge that is embodied in official paper publications, in favour of a more diverse account of science, in which tacit knowledge, materiality and practice play leading roles[391]. In a sense, this is a displacement of the concept of knowledge itself, but it can also be thought in terms of an extension of the concept of knowledge beyond what can be discursively formulated, the idea, as Polanyi said, that “we can know more than we can tell”[392].

Philosophy of science and epistemology have traditionally been dominated by views of science in which what counts are theories, bodies of explicit propositions, which are backed up by evidence and argument, and which in some sense represent the world. An appreciation of the role of software in computational science can help break down this image of science in favour of a much more pluralistic one, because on a number of fronts it becomes impossible to reduce software to ordinary discourse. Software itself is the output of research. It is not just a means to an end of creating statements about the world. Research is directly creating things, what Daston and Galison theorised as a recent trend in science from representation towards an “engineering-style *presentational* approach to the real”[393]. It is about creating stuff, about creating productive research systems as much as it is ever about representing the world in discourse.

Writing code is not a matter of conceptualising a process in natural language and then translating it into software. To the extent that there is a language of thought, for scientists who write software every day, this language is the programming language they habitually use, as much as it is the natural language (English) with which they explain their science. Many of my interviewees reported that they routinely “think in code” and that after their daily routine it is not uncommon to dream in

390 Winsberg, *Science in the Age of Computer Simulation*, 28.
391 See, for example Pickering, ‘From Science as Knowledge to Science as Practice’.
392 Polanyi, *The Tacit Dimension*, 4.
393 Daston and Galison, *Objectivity*, 395.

code. Scott Rosenberg entitled his study of a Silicon Valley software project “Dreaming in Code”, drawing on a quote from the well-known programmer Jaron Lanier:

> “To be effective at any large software project, you have to become so committed to it. You have to incorporate so much of it into your brain. I used to dream in code at night when I was in the middle of some big project”[394].

Natural language is not an underlying medium for science, but a source of awkwardness and trouble. Real difficulties are encountered when trying to describe research in natural language and in reading others' descriptions of software. QS recalled the trouble he had when implementing another scientist's parameterisation method:

> “The paper I based most of the turbulence parameterisation on has several major errors in the formulas. They have written out the maths and it is incorrect. It has to be incorrect. But the model results were correct. They gave the right answers. This is an easy mistake to make but it means that when you are reading papers and trying to reimplement the methods that they describe, the first thing you have to do is to check for errors.” (QS)

The transition between the software and the kind of propositional description found in journals is a difficult one. The peer reviewing process does not reliably identify these problems, which emerge during the reincorporation of these ideas into modelling frameworks. Moving from working in code to working in mathematical or natural language descriptions, and vice versa, is difficult. “The problem is that the more detail you go into, the more chance of error between what you have written down and what you actually ran” (QS).

Over the years, as Fluidity gained multiple new methods, functions and capabilities, it was accompanied by a growing trove of publications reporting their successes and exploring their potentials for further expansion. The challenge has always been for this piecemeal assemblage of descriptions to comprehensively relate to the software system. “In a publication,” says TX, “you would not describe your whole code, but would only describe the new part of it. You might, I suppose, reiterate the most important bits, but never the whole thing.” You can, of course, refer back to past publications, but for TX “there will be bits of the changes to the code that never resulted in a publication, either because they are too minor to be interesting, or because they are part of someone's PhD. Even if you go back to all the papers, and the manuals, there will be lots of gaps.” What thus applies to the software applies also to the model set-up, because Fluidity has a huge set of options. “I don't have all the details written down. They are all there in my Diamond file,” says QS. “If I wrote down all the parameters that are in my Diamond file that would be several tens of sheets of A4.”

As a project grows, the gaps between it and the discourse about it proliferate. This creates a problem of “witnessing”, of access and openness[395]. As the inadequacy of the written archive is progressively exposed, the software itself starts to function as a primary repository for the science itself. It is better

394 Lanier, quoted in Rosenberg, *Dreaming in Code*, 310.

395 Kelty, ‘Free Science’, 427.

to go straight to the source code than to look at descriptions of it. But as we have seen, this move is more than moving from one body of text to another. It takes us from a body of text to a field of practice.

Because the software pushes the limits of what is possible to say in publications, all you can do, says QS, is "to give supplemental information saying "all my input files are here", "the model is here"... If they really wanted to, a reviewer could go and run it for themselves." This is one of the principle reasons that Fluidity was made open source in 2010, something that required extensive negotiations with the university, which had to give up its intellectual property rights in the system in the process.

In an influential piece in Nature Climate Change, Kurt Kleiner notes that for climate science it is becoming essential to move beyond the past paradigm of releasing results, end products of analysis and visualisation, to a new level of openness in which the raw data itself is made available[396]. This sentiment was echoed at AMCG. "I don't think you can ever perform good science in a closed source environment simply because you need to show how you got the results. You can't write it all down in your paper and even if you do, you might get it wrong," says GY. Rather than rely on the traditional discourse of the public sphere, scientists have been opening up new channels, ones that are less limited by size constraints and by problems of translation, new media channels for the direct dissemination of code. As the apparatus for the research becomes less and less amenable to description in natural language, there is a growing demand to distribute it directly. From having been an empirical supplement to the knowledge disseminated in publication, the terms change places and publications become the textual supplement to the research primarily embodied in apparatuses and the systems of practice they inhabit.

With new channels for distribution, open source code and data, we are not just looking at a globally connected science[397]. We may have to reorganise the basic lenses through which we approach science. What would be an epistemology without paper? In other words, what is a digital epistemology? As well as transforming our appreciation of science into the vast scales and high speeds of computer processes, this also opens up a redeeming possibility for how we understand all science, pre- and post-computational science. Undermining the dominance of data-theory, we move to a focus on practice, on modelling, on phenomenotechnique. This kind of science, while globally distributed and highly technically mediated, nevertheless draws attention back to the concrete practices of its production, because it defies convertibility into textual description.

7.7 Conclusion

Code cannot be assimilated into discourse. It does not embody knowledge in the sense of containing it, ready to spit it out again, were it to be needed. They are not mutually convertible. It embodies

396 Kleiner, 'Data on Demand'.

397 Tiles, 'Technology and the Possibility of Global Environmental Science'; Edwards, *A Vast Machine*; Schroeder, 'e-Sciences as Research Technologies'.

knowledge in the sense that it participates in the kind of play of presence and absence addressed in chapter 5. In a representative play, it is and it stands for the practical nexus of research itself.

With computational science we encounter an intriguing possibility for the kind of practice theoretical rationalism endorsed here: the possibility that practice itself comes increasingly to the surface because the textual outputs, the publications disseminated after research has been done, are less and less effective as objectifications of the research process. The eye is drawn to those processes themselves. Software systems are continually shifting scaffolds for practice, and require users and developers to directly engage with their materialities. In their continual transformation and momentum the movement of research is grasped directly rather than indirectly through methodological write-ups. The software as a text does not describe what happened; it is what happened. It is happening and mutating all the time. It is possible, for this reason, that simulation science is the best possible ally for the kind of theory that science studies scholars and others interested in practice have been espousing for many years now.

The next chapter picks up on the threads of this one, and explores software as a material for research. We put the question of publication and knowledge on one side, and look at materials of practice, worlds of software with different tone and texture, facilitating different kinds of interaction, and different kinds of research in their domain.

8 Workability and Habitability

8.1 Introduction

> "Without materiality mediation is empty"[398].

In this chapter we take a further step into the world of scientific software development, and look at the materiality of research systems. What kind of world is this? What is software like as a material for doing science? What does it mean to be already thrown, already entangled, in a world of software?

I am interested in the skilled work within highly developed socio-technical systems that comprises the daily investigative work of scientists. Different kinds of media and material provide different possibilities for manipulation, for creativity and for recombination. Experimental work, in this sense, provided the classic example of "hands-on" research, the practical reasoning of thinking through doing. But work with software is equally "hands-on". It too builds skills, requires competencies, practical thinking, the kind of practical reason instantiated in working things out by playing around with them. For scientists whose everyday existence is one of working with software, it has a number of very tangible properties that emerge in the practices of its use which warrant the term "materiality".

While this practical element is often forgotten in studies of simulation, which often naïvely regard it as a domain of free invention[399], it has been highlighted in recent philosophy of modelling. "Manipulability" and "workability" have been flagged up as important features in how models fulfil an epistemic role[400]. Knuuttila has pushed this point by emphasising that materiality is important for even the most "ideal" kinds of modelling. Citing Klein's work on "paper tools"[401], she stresses that "even in the case of symbols and diagrams, the fact that they are materially embodied as written signs on a paper accounts partly for their manipulability"[402]. This precedent opens a still largely unexplored field with respect to understanding digital materialities, whose many dimensions remain

398 Knuuttila, 'Models, Representation, and Mediation', 1267.

399 For example, Heymann, 'Modeling Reality'.

400 Morgan and Morrison, *Models as Mediators*; Knuuttila, 'Models as Epistemic Artefacts'.

401 Klein, *Experiments, Models, Paper Tools*.

to be charted. How does software become efficacious as a medium for thought and action? How, conversely, does it threaten research by becoming unwieldy?

This chapter aims to answer these questions by telling the story of the growth of Fluidity, and drawing out “material” dimensions threatening workability: brittleness, bureaucratic sluggishness, portability and durability. Finally, I go on to draw these pressures together into the general question of “habitability” outlined by Richard Gabriel, which will bring us towards the final chapter, in raising the question of the stability and openness of efficacious environments for action.

8.2 The environment for doing computational science

The material culture of computational science is distinctive for its sparse aesthetic. The laboratory can hardly even be perceived as a rich material culture without a great effort of estrangement. So ubiquitous is the office environment, with its uniform colours, its swivel chairs, filing cabinets, monitors, keyboards, mice, it is hard to see it at all. Like the settings of so much modern work, AMCG is based in a connected set of open plan offices, with low dividers to break the gaze of seated occupants, while allowing the airy feel of mutual surveillance to dominate at higher eye-levels. The majority of scientists work in these divisions, peppered with post-it notes, stationary and personal paraphernalia. Those more senior have shared or individual offices. The open workspace blends into the kitchen area. Private meeting rooms sit off to one side. Carpet tiles, ceiling tiles. The gentle hum of whispered conversation and computer fans. Bookshelves store reference material and provide a platform for in-trays and printers. Whiteboards adorn the walls, on which sit ever-changing tangles of scribbled functions, graphs and diagrams.

It is sometimes hard to remember that these physical environments are as deeply affecting as any other. This is a symptom of their ubiquity and of their association with the bureaucracy that traverses contemporary culture, routine so routine that it is not really seen. David Graeber and Annelise Riles have begun the ambitious project of turning over the stones of this largely implicit ground for contemporary Western existence. Graeber points out the “interpretive depth” of the kinds of rituals that anthropologists have historically studied most, a depth that is distinctly lacking in Western bureaucratic procedures, which seem to gain their efficacy precisely from cutting exegesis short[403]. Riles' studies of paperwork and documents also follow this thread, and are informative for understanding any culture, like that of computational science, that has such a strong anti-interpretive effect that its spaces of work appear paradoxically to contain no culture whatsoever[404].

These features of computational science point to its contemporaneity, a truly Twenty-First Century science. It is an office job. This goes some way to explaining the bemusement of some of my

402 Knuuttila, ‘Modelling and Representing’, 269; see also Goody, *The Logic of Writing and the Organisation of Society*; Netz, *The Shaping of Deduction in Greek Mathematics*.

403 Graeber, ‘Beyond Power/Knowledge’.

404 Riles, *Documents*.

informants when I explained I wanted to do an ethnography of their group: “What are you going to see? We just sit here tapping on keyboards!” (QY)

So despite this resistance, how can we start to appreciate the nature of this work, of this “tapping on keyboards”? How do we start to appreciate the subtle nuances that show themselves throughout the day, the hopes and frustrations, the resistances and accommodations, under the surface of all this routine, that is research with software, research working upon software. Pointing to the sterile appearance of the environment tells us why it is that software as a material is easily ignored, but we need to go much further if we are to transform this ignorance to a new kind of engagement.

8.3 Software as a material

Most scientific software consists of environments or libraries for research. Fluidity is more like a flexible toolkit than a singular device. This is why I have called it a “modelling framework” rather than simply a “model”. It can generate innumerable simulations of many different kinds. No simulation will use all the tools in the toolbox, nor will different simulations use them in the same way. Different kinds of simulations will make use of different functions within Fluidity's libraries. Fluidity also draws on many external libraries for its solver routines, its parallelisation and its adjointing capabilities. It is integrated with several other software systems which provide its compilation, its set-up, its mesh-generation, and its post-processing. Fluidity is not one object that can be taken out of storage, placed on a work bench, and probed and prodded. It is the major element of the research systems for studying fluids at AMCG. It provides a concrete setting for this research.

Philosophical debates over simulation have long stressed questions of materiality, but remain caught in a singular problematic, one for which the issue is the nature of the inferences made by scientists who are working with different kinds of model or experiment. And the materiality in question is almost always the materiality of the actual electronic processes occurring in the computer hardware. With experiments, says Guala, scientists play upon the argument that the laboratory set-up is made of the “same stuff” as a broader set of physical, chemical or biological systems, and thus inferences can be extended outwards[405]. The “same material causes” are at work in one instance as in another[406]. In contrast, for idealised models, and for computational models, only a “formal” correspondence can be posited. Various exchanges played out between the alternative positions on this question. For Wendy Parker, for example, simulations must be thought of as material processes, and are thus not to be treated by analogy with abstract mathematical models[407]. Mary Morgan extended the debate, by bringing in hybrid cases involving experimental and simulation elements, and proposed a typology of intermediates[408].

405 Guala, ‘Models, Simulations, and Experiments’.

406 See also Winsberg, *Science in the Age of Computer Simulation*, 55–56.

407 Parker, ‘Does Matter Really Matter?’.

408 Morgan, ‘Experiments Without Material Intervention’.

But these questions follow a very particular line of enquiry. They are about the role of materiality in reasoning about experiments and simulations. To the extent that I am interested in reasoning, it is not this post-hoc reasoning that justifies what statements can be made about simulations. It is the practical reasoning that occurs in the very writing of scientific software, in work with this medium. This requires us to look beyond the materiality of hardware, beyond the kinds of inferences that may or may not be made based on empirical occurrences in a computer. What we need to do instead is to try to find ways to talk about the materiality of software, about the ways in which software systems are for the scientist like clay for a potter, a medium which needs to be experienced, to be learned from, to be played around with, and practical competencies developed as more elaborate or elegant forms get produced.

Both Tarja Knuuttila and Mary Morgan have started thinking along these lines in recent years. For Knuuttila, it is important that we don't get sidetracked by the representational relationship between the simulation and whatever it might or might not represent. Such concerns are certainly legitimate but they have historically overwhelmed equally interesting features of models and simulations, their role as what Knuuttila calls “epistemic artefacts”, as things that are, to paraphrase Levi-Strauss, “good to think with”. Knuuttila encourages us to explore models' “constraints and affordances”, what they provide to the user by way of possibilities of manipulation[409].

The term “constraint” has been the subject of some dispute, largely revolving over the question of whether it implies an external, pre-existing property impinging on scientific practice, or whether conversely it should imply a resistance immanent to practice[410]. For present purposes, it is somewhere in between. The materiality of software systems involves properties of those systems that are not necessarily bound to any specific practical situation. But at the same time, as will become clear, these materialities are only fully graspable when seen unfolding in practice.

The term “affordance” was coined by James Gibson to talk about the possibilities offered by the immediate environment relative to an embodied being[411]. It is not just that the environment allows certain actions for me (such as sitting, climbing, hiding, etc.). These actions are also perceptible possibilities for me[412]. Where models are concrete, for example in the case of plastic models of molecules used in chemistry classrooms, the affordances are what suggest possibilities for manipulation[413]. If the plastic balls are too small or too big, they are less easily manipulable, too fiddly or cumbersome. But if they are of a medium size, in relation to the hands and body, the user who is playing around with them can easily see and feel new ways of connecting things, new isomers, for example. This brings in another key dimension, which must not be forgotten. The embodied being is not preformed in its capacities, but has continuously evolving dispositions.

409 Knuuttila, ‘Models as Epistemic Artefacts’; Knuuttila, ‘Modelling and Representing’.

410 Galison, ‘Context and Constraints’; Pickering, ‘Beyond Constraint’.

411 Gibson, *The Ecological Approach To Visual Perception*, 127–143.

412 Ibid., 128.

413 Justi and Gilbert, ‘Models and Modelling in Chemical Education’.

Manipulability is relative not just to the body, but to the whole field of practice, of bodily cultivation. Where models are realised in software, this “playing around” becomes more abstract, in the sense that it is about the manipulation of digital entities and processes rather than of physical objects. But the software system nevertheless offers certain affordances for its manipulation or modification. If we follow Giere in his use of “distributed cognition”, even the most abstract models become tied to possibilities of working with things[414].

Mary Morgan uses the term “workability” to stress the importance of the manipulability of models to their role in the accomplishment of scientific research[415]. To be effective, they must be workable, relative to available potentialities for grasping them. With this in view, we start to understand something of the variation native to software development, something of its idiosyncrasies, of its resistances and openings, in and through which scientific work is more or less productive. There are different kinds of software systems, and software systems which change in kind, and in each case, something can be said about what they offer to the user as a perceptible potential for doing good research.

This work with software is a practical engagement that we can call, with Bachelard, the “objective meditation” of research practice, the thought inherent in working in a laboratory as opposed to the “subjective meditation” of pure or purified thought[416]. “Objective meditation” is phenomenotechnique, reason in practice. With Morgan and Knuuttila we can stress the relativity to bodies that such practical deployment implies. Some ways of putting concepts into practice, whether in experiment or modelling, will be extremely efficacious. Others will be unproductive because they are unmanageable, not so easily handled.

Software systems come in many shapes and sizes, but anyone who has been involved in software development will be able to empathise with the general problems of workability that it poses. It is possible to create highly complex systems with software, but these can easily cease to be workable. Things can “get out of hand”. Workability is always relative to the kind of work being carried out. What concerns me here is not primarily the “black box” use of software interfaces, that we find explored in human-computer interaction studies, but the actual intervention in their source code, writing and modifying code, finding your way among complex systems of writing, tracing their logics and relating what is expressed there to the output, the computations that system can be made to generate.

In the world of software development, bugs are a fact of life. Frustrations are a fact of life. Software can easily become unwieldy. It is worth beginning with the simple observation that the failure rate for large software projects is extremely high. Humphrey cites figures to the tune of an 85% failure

414 Giere, *Scientific Perspectivism*, 96–116.
415 Morgan, ‘Models as Working Objects in Science’.
416 Bachelard, *The New Scientific Spirit*, 171.

rate for projects between 3 and 6 million dollars, and over 90% for those over 6 million, where a failed project either goes significantly over budget or is abandoned with no deliverables delivered[417].

In the following sections I consider three major axes of workability: brittle/robust software, the tar pit, and hardware portability. Each of these has a major effect on the ways in which software can be put to work in science, and strongly conditions how that research proceeds.

8.4 The troubles with growing software

Where it is under active development, scientific software grows in a piecemeal fashion, a consequence of the dominant funding model. It is very rare to get funding to build a large software system from scratch. What is feasible is usually to build a small software system, within the terms of individually funded research projects, a system tailored to the specific goals of that project. Once built, however, this can serve as a proof of concept for further projects, which extend the software in new directions, adding new functionalities, exploring new possibilities. Success in producing results thus tends to “grow” software within scientific contexts[418]. This has been the experience of the scientists who built Fluidity.

Fluidity began its life in the late 1980s as part of a PhD project looking at small scale fluid phenomena for industrial applications. The original author of Fluidity, CK, brought his code to Imperial College a few years later when he joined the Applied Modelling group, which at that time was primarily concerned with studying radiation transport problems for nuclear safety engineering. It soon became apparent that Fluidity would be usefully integrated into the tool kit of the group, because this opened the door to studying nuclear applications involving fluids, such as those found in “fissile solution” reactors. With the assistance of a new coupling code, “FETCH”, Fluidity was coupled to “EVENT”, at that time the main AMCG nuclear modelling code.

This emerging interest in fluids within AMCG sparked much further development of Fluidity and in the late 1990s new staff were brought on board to extend the code to new applications such as multiphase flow (where fluids move between liquid and gas phases) and to render it capable of running efficiently on highly parallel supercomputer architectures. At this point, AMCG was composed of 15 scientists, and was roughly evenly divided in its work between radiation problems and fluids problems. In the early 2000s, however, a major new funding initiative began which doubled the size of the group within four years, and rapidly accelerated the pace of development on the code. The main part of this project was a major extension of Fluidity from its previous incarnation as principally an engineering code, to a huge new realm of geophysical applications, from oceans, to coasts, rivers, atmospheres and mantle dynamics.

417 Humphrey, ‘Why Big Software Projects Fail’.

418 Basili et al., ‘Understanding the HPC Community’, 29.

This time of great expansion reveals a lot about what software is like as a material, and about the ways in which workability can become threatened. As the group expanded, the fluids side of things became by far the largest area of study within AMCG, and by 2010, there were around 30 scientists actively developing Fluidity. The code became a different kind of material for science, subject to different kinds of pressures. The engineering and small-scale computational fluid dynamics research continued, while geophysical applications accounted for more than half of what this much bigger development team were studying. The beginnings of this expansion in 2001-2003 saw the commencement of many new geophysical research projects but it also saw a major shift in the way that the science was done.

The expansion of the group was an expansion in the number of people involved, but this implied an expansion in the number of different agendas to which Fluidity was being put. The group experienced new demands for collaboration and communication. At the same time, the increase in scale resulted in an increased rate of change for the Fluidity code base. The software was being changed often, by many people working on many different projects. These stresses placed a burden on the team that was exacerbated by the fact that much of the code had never originally been written with an expectation of future massive expansion. The priority during the early life of Fluidity had always been to deliver the short-term aims of specific research projects, and there had been little time or money available to make the system maximally extendible. Furthermore, as is the case for many computational science teams, the majority of scientists working on Fluidity had not come from a computer science background[419]. They were predominantly mathematicians, engineers or physical scientists. Even now, most new PhD students coming to AMCG have never written any software before they arrived. They learn on the job and as KU pointed out, this has good and bad effects: "when you have a team of good programmers around you you learn by osmosis. But when you have a team of bad programmers around you you learn by osmosis as well!" (KU). While Fluidity was a small or medium sized enterprise, there was a degree of tolerance for idiosyncratic working practices, but as it grew, this tolerance shrank.

> "As a mathematician I was never interested in coding practices. It didn't bother me. As long as my code ran fast and did what it was supposed to do I was happy. So my codes were often a tremendous mess. Someone like WS would have been none too happy seeing that. But then that is no problem when you are on your own: no-one else has to look at it." (IM)
>
> "Over the years Fluidity had been adapted... bit by bit for different applications. We ended up with quite an unmanageable mess, what coders call "the code becomes brittle". There were so many hidden assumptions in the code that as soon as you change one detail the whole thing breaks down" (HP)

The trouble with the code in the early days of the 2000s expansion was expressed in very tangible terms. The code had become "brittle". This term is a commonplace within software cultures. Brittle code is the opposite of robust code. Brittle code breaks more often than robust code. But the

419 Ibid.

fundamental problem with brittle code is not so much that it breaks more often – all code can be expected to break when you are developing it – it is that when it does break it is hard to figure out exactly why. Then, when you have figured out why it broke, it is hard to fix it. And when you do implement the fix, there is a high probability of that causing further problems. “Brittleness” is a somewhat loose term that captures the experience of coders struggling with working in what has become a very difficult and frustrating medium.

> “Have you seen the old code? It was written without any levels of abstraction whatsoever. That makes the code very difficult to understand because what you see is a whole load of very low level mathematical operations and then you have to work out for yourself what the big picture is and what all the bits are doing. And that is hard to see. It was missing an awful lot of modern software engineering, which is about making code that is either error free by design or at least easy to define what the errors are. So there were undiagnosed errors that had been in there for years because you can't find them. (WS)

Brittleness seriously compromises workability. Working with and upon a brittle Fluidity rendered research difficult and time consuming. If lots of time is spent fixing bugs and tracing the logic of convoluted sections of source code, there is very little sense of the code as a flexible medium. It is not conducive to just “trying things out” in the set-up of a simulation, following a whim, an informal inclination. Not only do these processes take much longer, it is harder for new collaborators to be brought on board. One complaint with the old Fluidity was that it was hard for new PhD students to properly become experts in the code in the 3 year span of their doctorate.

> “The interface originally was very cumbersome. Only two or three people in the group were able to use it. It was a big text file with lots of random names. They all had to be six letters so it used all sorts of crazy acronyms. It was very hard to modify the option system but they were constantly adding new functionality so basically the numbers became encoded in more and more complicated ways. For example, if you put a minus sign in the time step that might mean something special. Or you put in a very large number with different meanings for the different digits. The option system basically dated back to the days of punch cards and it was very complicated to set up meshes. Nobody really knew why it broke when you changed something. It was a big project to step by step change things to see where it broke and figure out why” (HP)

8.5 The scale of software

What makes a software system brittle? Fluidity became brittle as it became big and as the group developing it became larger, so brittleness is often relative to scale. It is, however, difficult to talk rigorously about the size of software. There is no perfect measure. The most common measure is that of the number of lines of code in the source code repository. This is the software written by humans, before it is compiled into binary code that can be executed on hardware. But counting lines of source code is somewhat arbitrary: different software languages use different rules of formatting that take up different amounts of space. This is compounded by the fact that Fluidity involves different languages working together. According to some relatively standard calculations, however, Fluidity gets currently estimated at over a million lines of code. At the point of the rewrite it already comprised more than a quarter of a million.

But the size of the source code is not a good measure for the complexity of the code[420]. This can be split into two issues: the complexity of the software architecture and the complexity of the processes to be computed. The complexity of the software architecture had become a problem for the Fluidity developers. Because the code had grown in piecemeal fashion there was very little large-scale planning at the start, and many additions were tacked on in inconsistent ways. A code that is very large in terms of lines of code need not be complex for a human reader to find their way around. It may be very intuitively organised. Conversely, even a relatively small code may have a torturous organisation with many headaches in store for a navigator.

The old Fluidity was big but the real difficulty came from its organisation. It is impossible to expect the developers of a young software project to anticipate its future directions in such a way that they can pre-emptively provide a structure into which each new piece of functionality can fit. This being said, the old Fluidity implemented very few design principles that are pretty much standard practice in software development outside of science.

> There was also no concept of orthogonality in the code: one of the design principles you want when you design code is that things that are conceptually unrelated are also unrelated in the code, so that you can play around with the code in this area without having to worry about the other areas. But if the code is all mixed up that doesn't work. They are all on top of each other." (WS)

Another property that captures an element of the scale of software is the complexity of the processes that it computes. Some software performs tasks that require a lot of space to be encoded, but which are relatively conceptually simple, whereas something like Fluidity is computing a lot of very complex mathematics, and this exerts a large influence on how manageable its source code is.

The fourth aspect of software size and complexity is its rate of change. If the code is being updated on a daily basis and brand new parts added in regularly, any problems dealing with size and complexity become much more serious. It is one thing to know your way around a complex code well enough that it is "workable" for a research agenda. It is another thing to maintain this level of engagement while that code is constantly changing. Such is the rate of change for a project like Fluidity, it is necessary to think of Fluidity itself not as a code that is sometimes updated, but rather as something perpetually changing.

420 Booch, 'Measuring Architectural Complexity'.

8.6 Making Fluidity robust

As the group's expansion got under way it became clear that the code would only become more brittle as more scientists carried on adding new functionality. Some scientists were already refusing to work with the latest version of the code because they were sick of results that they had got one week no longer being reproducible a week later. The response was two-fold. Firstly there would be a complete rewrite of the code, completely reorganising its architecture and the style of programming in which it had been written. Secondly, a number of supplementary practices and technologies would be adopted to help co-ordinate the work and to change the working style of the group.

> “When the grant to develop the ICOM [Imperial College Ocean Model] model arrived, all of a sudden a lot of people were working on the development side. Code organisation and management became critical. We hired people who had good ideas about testing, code structures, modularisation” (HP)

The complete rewrite of Fluidity took a small team about a year. While there was no funding earmarked specifically for this task, a number of new projects were just getting off the ground and this work could be built into their early phases. The rewrite required a huge amount of work, but its pay-off was to be a newly efficacious, much more workable code. It is an extreme measure to rewrite a code from scratch, but the new code reimplemented the same core algorithms that had been developed over the previous 15 years of Fluidity's lifetime. While the work of expressing these algorithms in software had to be redone, all the work that it took to devise them in the first place did not need to be repeated.

The “new Fluidity” did what the old Fluidity did, but it expressed it in a more reader-friendly and manipulator-friendly way, organising the core algorithms so it was much easier to find your way around the source code. The new code used a newer version of Fortran and employed many new stylistic techniques, even down to implementing new rules for naming variables in a consistent manner. The modularity of the new code allowed new functionalities to be added easily and for them to interface with existing modules in a standardised manner. A new options system with a clean and clear graphical user interface complemented this process, speeding up the process of running new simulations and tweaking them in the course of their investigation.

> Things have moved on dramatically in the last few years.... People came in too who knew more about software development and modern programming. Slowly what happened was that good practice and rewriting the code in modern Fortran happened. The code we have now has effectively zero lines in common with the code that I started on. The algorithms are the same – these are the important part, the hard part. The early stages of working on these codes are getting the algorithms. But then get an algorithm and there are hundreds of ways that you can code that, lots of choices of how you implement it: different languages, different structures, different orders of things that you do things in. One of the big things that happens now in dev. [developers'] meetings are debates over this kind of thing because there are lots of ways of doing something: What is the best way to do it? And people have strong

> feelings about this, making sure that things are useful for other people and future-proof. In the old days you were only worried about getting something working. From 2005 onwards there was lots more emphasis on producing a product that now we have released [as open source software] (NK)

The modularity of the new version of Fluidity was taken as far as possible. WS advocated this tactic in a presentation to representatives from the Natural Environment Research Council. Modularity in its extreme means that parts of Fluidity could in theory be transplanted into very different modelling frameworks without having to be retooled for the new job. For example, in the main body of the code,

> “we don't code anything about the discretisation, so we can let things change without having to recode. The code doesn't even know it is a fluids model. It just knows it is solving PDEs [Partial Differential Equations].” (WS)

It is interesting to note that while a complete rewrite of a software system is a radical tactic, in principle it doesn't have any bearing on the epistemic legitimacy of the research that had been done with the old version of the code. Exactly the same simulation, the same computational process, may be executed by a brittle code as by a robust code. Or if these processes differ, they may differ in no significant manner. Just because a code is robust does not mean that the simulation it produces will give a more legitimate answer to a given question. This issue is one of validation and verification, which pertain to a finished simulation and thus bear little on the actual process of writing and manipulating software.

In practice, however, the robustness of software does matter. It matters because a robust code facilitates a greater degree of manipulation, and therefore offers a lot more potentiality to surprise the scientists, to lead him or her down an unexpected path of exploration, towards innovative research. It also matters because the justification of results is accomplished by a drawing together of a patchwork of arguments. Verification and validation are very important elements here, and probably the major components of this patchwork. But they are never conclusive. A code that is known to be robust and well-written is likely to include fewer bugs, so less likely to involve bugs affecting the solution that have eluded detection by verification and validation.

These considerations point to the real difference between the “artefactual” approach to modelling (espoused by Knuuttila) and more mainstream epistemology of simulation. The latter has largely focussed on validation and verification as strategies for justifying propositions about a final simulation[421], while the former draws our attention to the research process *before* the final simulation has been arrived at. It is here that we can talk of scientific practice as skilled activity within a material environment, which provides specific ways of working and certain very tangible constraints and affordances on the cognitive processes of making and manipulating.

421 For example, Bailer-Jones, ‘When Scientific Models Represent’.

8.7 Programming systems products

Fluidity was made robust, or at least significantly more robust, by the full re-write. But it would be a mistake to think about this kind of manipulability solely as a property of the software itself. Everything depends on what you are going to use the code for, and how you are going to work with it. There is no straightforward way to isolate the technology from the techniques through which brought into use. The overcoming of the brittleness of Fluidity was also the overcoming of the frustrations that arose from working in a larger group. During the same period in which the code was rewritten a number of what we could call “social technologies” were implemented within the group. Its rewriting was also a reconfiguring of the working procedures of the researchers who used it.

> “Part of our quality control process for the code is all this testing and that is both in order to make sure that new features that people develop don't cause problems but also to make sure that people haven't sort of messed it all up or literally made a spelling mistake or get a wrong syntax” (QH)

A full version control system tracked every change made to the source code, an automated testing suite checked every new version of the code for errors. More recently, the group pushed these trends forwards with a code review system, so that every new section of software is reviewed by another developer to try to pick up errors or just problems of style. A manual and user guide, and set of example simulations, is kept up to date so that new users can get to grips with the code quickly and easily.

> “Some reviews are extensive... [The time commitment] can be expensive but probably not as expensive as a bug getting in. We have a couple of pernicious bugs in the code at the moment; we are trying to ferret them out” (HU)

Since November 2010, Fluidity has been open source, so new users from around the world can download the software and they inevitably need to be provided with support. By expanding the user-base, being open source also has the advantage of enhancing the probability that bugs will be discovered.

> There are major advantages [to open source] in the sense that hopefully we will have people across the world using our code. And they will find bugs, guaranteed. They will report those bugs and we can fix them and that is all going to improve the validity of the code (IW)

Going public with the code is a bold strategy, because it exposes the innermost workings of your research practice to scrutiny.

> The fact that people can see your code means that you will be more careful with it. It ensures that your code is good. It is also like the [Microsoft] Windows release cycle. They release SP1 [Service Pack 1], SP2, SP3 in response to error reports people have sent in. The public do the error testing... (GN)

These technologies help keep Fluidity reliable and stable for its users. The code is changing all the time, but if you got a certain result last week, even if it doesn't work this week you know exactly the version of the code in which it did work and you know who is responsible for the change, and the automated testing system probably already alerted you to the problem as soon as it happened, rather than having to find out further down the line.

On the other hand, the introduction of all these systems has had a huge impact on working practices. While they make the code more easily manipulable, they also make a huge number of new demands on scientists' working lives. For every new piece of code, a test must be written, the manual must be updated, and a review must take place. Requests for help on the email lists and chat channel must be responded to. Because the style and structure of the code must be kept consistent, the group started having weekly developers' meetings in order that everyone can be kept up to date with all the other research projects that are currently going on and so that proposed changes and additions can be discussed before they are implemented. These meetings must be attended. Work with the old Fluidity could be quite individualistic, with a great freedom to work how the scientists liked, in control of their own time. But the social environment of the new Fluidity is more akin to a bureaucratic organisation, with constant meetings and communications. A more workable code seemed to bring with it constraints on work and time that left little remainder for the kind of playing around that the new code facilitated.

These pressures are not new, and they are not confined to scientific contexts. They were the subject of a classic essay in software engineering called “The Tar Pit”, by Frederick Brooks Jr.[422]. He wanted to understand the complaints of software developers who were working in big bureaucratic organisations, of the sort that developed big projects like operating systems. He wanted to understand their frustrations with all the management, all the meetings, all the regulations that were imposed on them. The feeling was that if only they could be left alone, if only they could just be free of all this bureaucracy, this feeling of wading through a tar pit of daily routine, then they could be so much more productive. The fresh air of freedom and they could write many more lines of code; they could create something amazing. If brittleness is one dimension of the materiality of software, the tar pit is another, the feeling of density and sluggishness imposed by the bureaucratic structures that grow up around the central task of writing big software.

“One occasionally reads newspaper accounts of how two programmers in a remodelled garage have built an important program that surpasses that best efforts of large teams. And every programmer is prepared to believe such tales, for he knows that he could build *any* program much faster than the 1000 statements/year reported for industrial teams”[423]. So, Brooks asks, “[w]hy then have not all industrial programming teams been replaced by dedicated garage duos?”[424] This myth, of the couple of guys (and in the myth they are of course male) sitting in their garage, writing something of great

422 Brooks, Jr., ‘The Tar Pit’.
423 Ibid., 4.
424 Ibid.

genius, is still prevalent today, not least through the figure of the Google founders, who wrote something amazing and ended up taking over the world.

The problem, says Brooks, is that this way of thinking is mistaken about what it is that is actually being created in the different contexts. What gets written in the garage is a program. A program “is complete in itself, ready to be run by the author on the system on which it was developed”[425]. It does the core task, and may do it exceptionally well. But it works for that person, on that computer. What the big organisations are after is something quite different: a “programming systems product”. This “can be run, tested, extended or repaired by anybody... usable in many operating environments, for many sets of data... written in a generalisable fashion... [with] thorough documentation”[426]. It also must conform with standard interface design, with control on memory usage, tested in all permutations with multiple other systems with which it is going to need to be able to coexist and interact[427]. Brooks' estimate is that to create a programming systems product requires nine times as much work as a program, and it is this amplification of effort that gives the tangible feeling of wading through a tar pit when working on big software compared with working on an individual pet project.

In other words, the transition to big software brought with it a number of changes to the technologies used to do research at AMCG, and while these technologies made the software workable in new ways, they also imposed new kinds of burdens. While it is possible to make every effort to smooth out working processes, big software does simply require a lot of additional effort in order to keep it manipulable. While the software may be easier to play around with in the research situation, less time is available for that play because of the extra demands of the bureaucratic machinery.

For these reasons, there remains a subdued but still perceptible tension within the group between the pioneers of the bureaucratic transformation of the group, and those who would like to work in a more individualistic manner. For them, it seems, the efficacy of their practice is compromised by an exaggerated emphasis on bureaucracy, which is constantly distracting attention from the core task of research.

> “There are also people... I call them “the code police”... These are people that really understand the code to a level that others don't. Any changes that aren't suitable they will say “look, that is useless,” often with my changes, and say “do it this way, do it that way”.” (IW)
>
> “There are some people here who are very interested in how the code is written so they like to have all these meetings and discuss things all the time and update people. I understand you need various things in place. But from a personal perspective I just want to do science.” (IM)

425 Ibid.
426 Ibid., 5–6.
427 Ibid., 6.

CA tries to balance these feelings. “Sometimes there is a feeling of being too software engineering focussed but in the long run you need to do that, otherwise you will end up with codes that are impossible to run” (CA).

Reflecting upon the broader significance of these ideas, it is interesting to posit a fundamental division in computational science between that which gets done with programs, and what, on the other hand, only gets done with programming systems products. A lot of computational science is done in small groups, with relatively short-term goals for the software, and a more flexible attitude to its longevity and portability. This software is written in the scientific equivalent of a garage (probably the office of a principle investigator with a couple of his or her PhD students tacked on, maybe a couple of postdocs). In most of these cases there is no need to create a programming systems product. You don't need to worry about new people being able to get on board quickly because the whole team was on board from the start. You don't need to worry about other groups using the software because they could just write a similar application for themselves from scratch. You don't need the software to run on any computer but the facility that you have access to for that project, and for which you have been writing from the start.

But for some significant subset of problems, the kinds of systems that are to be the focus of research are so complex that a large software framework is required. Some geophysical problems involve multiple interacting processes, operating on many scales. Software above a certain size has to be a programming systems product approach if it is to stay workable, to remain material for productive research. A big team needs to be co-ordinated around it. The project will take long enough that some of the founders will leave and new people will have to come on board part way through. The time it takes to build in all the necessary functionality and to validate all the parts of the model is so great, that the software itself needs future-proofing. It is not just the money; careers are being invested. The legacy needs to be more than the paper output of publications. It extends to the software itself, the provision of a platform for further studies into the future. In these cases, and climate science is probably the best example, software is an output of scientific activity in its own right, and for that to be the case, the software and the frameworks that surround it take a different form to small-scale endeavours. It must be legible, extendible, and widely-compatible, a programming systems product. Problems of brittleness must be overcome, and ways must be found to live with consequent processes of bureaucratisation.

8.8 Portability and code generation

One of the most important properties of a programming systems product is its portability. It is designed to work on a range of different computers. But portability is a relative concept. Most portabilities relate to operation on different operating systems, with different compilers, computers with different numbers of processors, and so on. A great deal of work is invested in keeping software functional when travelling across these kinds of gaps, taming these differences. But there is always the possibility of new gaps emerging in the landscape of evolving technologies that require radically different approaches, that make previous portability seem comparatively minor, as if for all the effort it entailed it never left the comfort of home at all. One such abyss is on the horizon of computational science. It is caused by a seismic shift in fundamental technologies that threatens the integrity of research practice. This challenge is provoking serious methodological reflection on the part of AMCG in order to prepare the group for a future with very different kinds of supercomputers. Making Fluidity a programming systems product was one thing; keeping it one in the face of wider shifts in the technical environment is another. Workability can only be provisionally assured, and requires continual vigilant maintenance work.

A handful of members of AMCG are currently working in tandem with a team from the computer science department. They are investigating automated code generation, trying to rethink the way that computer simulations are written. The core work involves adapting the existing language Unified Form Language (UFL) developed by the FEniCs project, developing its libraries such that finite-element simulations can be written at a new higher level of abstraction, using a new language to specify the equations to be solved and the methods to be used to solve them. The point is to provide new flexibility at the level of coding, so that Fluidity can be adapted with minimal effort to run on new kinds of supercomputer architectures.

The automated code generation project will ultimately produce a new system of code, a new rewrite, scheduled to be integrated into Fluidity from 2014. But this very practical output is also accompanied by a profusion of theoretical computational science papers, as in the course of the project the team studies the fundamentals of computational science software, its challenges and practice, contributing to the science of method, what could be called the “theory of doing computational science”.

The UFL project responds to challenges to workability, but in this case these challenges are *not yet* faced by computational scientists. The new supercomputer architectures are yet to make it into the mainstream. But they are in the pipeline, and hence, the consequent response from the software side of things is also in development. Faced with this future threat to workability, the UFL project is an exercise in pre-emption. As an expert in high-performance computing and one of the original developers who parallelised Fluidity during the late 1990s, TT explains the situation: “There are paradigm shifts in the technology... Supercomputers are going to become much much more complicated beasts to program” (TT).

The hardware of supercomputers changes all the time as better components are produced, but some shifts count as what TT calls “paradigm shifts”. They change the nature of the machine such that it will require different kinds of software to be produced. Currently, Fluidity's software embodies a whole set of implicit assumptions about what kind of machine will run it, rendering it poorly transferable to novel kinds of architecture.

> “It is always one thing to get things to run on this specific architecture, and a second thing to really run them efficiently. This [writing for novel architectures] is a bit different from what programming CPU [Central Processing Units – the standard architecture] has become... CPUs offer a very high level of comfort because they take care of a lot of things internally which you as a programmer don't really need to think about. But you have limited opportunity to influence things like caching and instruction-level parallelism that GPUs [Graphics Processing Units] offer today. Programming a CPU today is much closer to the hardware and that is also the problem and why we want to introduce these abstractions because it is hard to efficiently program for GPUs.” (UI)

GPUs are one of the elements of these anticipated future exotic architectures. As the name suggests, they are designed to handle graphics computations, which can be rendered massively parallel by dividing the computer screen into pixels, or regions of pixels. While desktop CPUs today tend to have two or four, or perhaps eight processors, GPUs have evolved to have thousands of relatively slow processors, with a shared memory cache. Soon after their appearance in the marketplace, however, GPUs were put to other uses than for graphics. They became, in the words of UI, “the poor man's supercomputer”, used by computational scientists as a cheap way to run massively parallel programs. GPUs are expected to provide some of the key technology for novel supercomputer architectures, and have thus taken on even greater importance, as placeholders for an uncertain future, testing grounds for new techniques.

Part of the problem lies in the fact that it is hard to program for parallel processing. It is difficult to think in parallel. Fluidity, like the vast majority of software, is written in a serial fashion, but in such a way as to render its operations decomposable during the compilation process, to be distributed across many processors. Because writing this software in such a way that it can easily be made parallel requires making assumptions about what the target architecture is like, WS will refer to it as “hand coding parallel”.

> “Hand coding parallel involves making decisions that are heavily driven by the sort of machine that it is going to run on. So the code you write is not parallel for different sorts of parallel machines. [In the future] the architecture frameworks will change radically and individual machines will have different sorts of parallelism. At the moment you have paradigms of shared memory machines or a distributed memory machine. But the future paradigm is that there will be a distributed layer and each of those nodes is itself quite big and within that node there might be one or more layers of shared memory parallelism. And there might also be something distributed there.” (WS)
>
> “One of the things that is happening is that with these graphics cards... if you have multiple graphics cards on one node, they can't really talk to each other, so they talk to each other indirectly via the node – so that's two layers of distribution – then each graphics card is itself a distributed machine which works on a sort of shared memory model, but it is a shared

> memory model that itself has two or three layers of memory. So we can easily get up to five different layers of parallelism in one code. It is already hard to program for that and the next machine that comes along will have a different five layers of parallelism” (WS)

Dealing with all these hardware architectures is the major problem that the automated code generation project is supposed to address. It is not just that the new supercomputers are complicated. The threat to workability comes from their variety, and there is little chance that this variability will settle down to a singular new conventional architecture to replace the CPUs that have dominated the whole history of digital computing thus far. TT continues:

> “You have all these innovative technologies such as hardware accelerators, graphics cards, and there are multiple different competing technologies. The one thing that they all have in common is that they are all vastly more complicated to program... Automatic code generation for these target platforms is going to become more and more important. WS is collaborating with KP over in computer science on this and they will talk about “software technologies” rather than software.” (TT)
>
> “[We would be] developing these algorithms rather than hand coding them ourselves to our target machine, where we are very almost actually touching the silicon: that's how close we are, the level of detail which you have to program. [In the future] the direction would shift entirely and you would start writing your algorithms in a much higher level language, and a whole new layer of technology would place itself between you and the silicon, so you are dealing with a much higher level language and you are mapping that to much more complicated computing devices.” (TT)

The question is not one of dealing with the divide between ordinary computers and supercomputers, which has largely been a question of scalability, of scaling software to run on increasing numbers of processors. It is a question of exotic architectures, which makes the previous paradigm, present day PCs and supercomputers alike, look monolithic.

> “HECToR [the UK national supercomputing resource] is made up of components like CPUs that are no different from your laptop and the piece of code that we would write would be no different from the one that we would run with the same compiler on your laptop. There is no major shift there – and that is common with almost all supercomputers – but there are trends coming along where the whole beast that you are running on is shifting fundamentally. This is a big shift in terms of everything we have seen for the last, say, thirty years in terms of computing and a feature of that is that the computers are going to become much more complicated in terms of the kind of code you are going to have to write for that.” (TT)
>
> “It is no longer business as usual from a programmer's perspective, for all us writers of software... On a GPU our software is just not going to work. It has to be rewritten entirely, so this puts up a massive technical barrier and most likely what is going to happen is that there is just going to be a paradigm shift that will shift the way that we develop models so that we shift to having domain specific languages and you would write your algorithms for these domain specific languages. At the moment we write code and the same compiler that is used for developing software like your word processor is the same type of compiler that we use for our climate models. It generates assembly code that is going to run on your silicon.

> In the future when everything is much more complicated you simply don't have the workforce to implement everything for all these different types of machines. You are going to end up with a new level of complexity... We are going to start writing in a language that is specific to climate modelling or to other applications in our field. And then you have a new layer of technology that is going to get that and convert it into a much more complicated source code. And then a compiler would get that and convert it to your architecture. So a whole new field of people is institutionally going to get inserted in the middle." (TT)

While the automated code generation project produces discursive outputs and software outputs, one of its most significant practical effects is to be social. Writing software encodes specialised understandings, and the interrelations of systems of software involves the interrelatedness of communities of specialists. In the new world of UFL, WS envisages a new relationship between computer science and computational science. In the current arrangement, computational scientists produce code, which they then compile and run on hardware. Computer scientists complement this work by developing the systems that manage the hardware and developing compilers and assemblers that will transform the code into effective binaries. But there is a limit to how much a computer scientist can manipulate the code because of the large amount of implicit mathematics of the approximations used to simulate the physics of the continuum. The aim is to change the way that computational scientists develop their models, so that their relationship to computer scientists can be changed, in such a way as to enable computer scientists "not to have to know" about the mathematics of the simulation in order to be able to work on the problem of how to run its software on computer hardware. The research project has thus opened up the institutional question of whether or not it is a problem that there is limited communicability between computer science and computational science disciplines, and whether a technical "paradigm shift" might prepare the ground for a new, much closer, relationship.

> "The mathematical skills that go into [computer science] are what I would describe as being "discrete": set theory, counting, combinatorics, graph theory... Whereas scientific computing [i.e. computational science] is very much about differential equations. That is the starting point. You have to discretise [the equations] to put it on a computer because there is no continuum on a computer chip. In a maths department they teach both types of mathematics but in a computer science department they would only teach the discrete side of things because that is the foundation of it." (CE)

The possibilities for utilising the latest optimisation tricks are therefore limited because many of the techniques of the computer scientist would risk "breaking the abstractions" within the code. This barrier between computer science and computational science is just one instance of a practical limitation on individual expertise. Very few scientists are able to straddle the two disciplines. It is therefore important to see the technologies developed as part of the automated code generation project as *social technologies* as much as they are calculating machines. They facilitate new kinds of collaboration between different kinds of scientists, new ways to integrate different kinds of work.

> "The amount of stuff that a PhD student needs to have in their head to do this stuff is getting unmanageable. And not just the PhD student. People who work in these fields need more information than they can reasonably expect to have mastery of. So part of this [project] is carving up the fields so that people who are experts in something can bring their expertise to the problem." (WS)

The automated code generation project is aiming to introduce a higher level of programming language (UFL), in which the computational scientist would write the core mathematics of their numerical methods. This discretisation scheme would then be parsed by a system that would analyse its structure such that it can be optimised for the specific kind of hardware that it will run on, automatically generating code that is amenable to all the latest techniques. When the hardware changes, this intermediary system can be changed accordingly, without requiring a rewrite of the top level mathematics. "I see this as an opportunity to specialise, not a dumbing down. It is an opportunity to get computer scientists involved, especially compiler experts" (WS). The field could become even more interdisciplinary, enabling compiler experts to participate, producing simulations at the very cutting edge of advances in computer science.

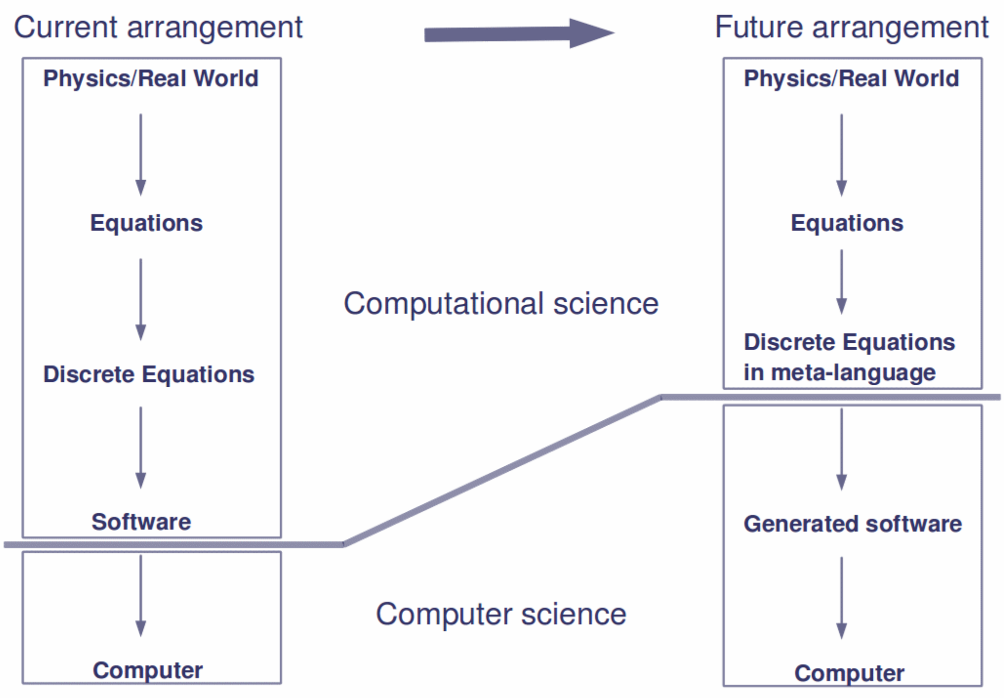


Figure 12: Changes in disciplinary interface anticipated by the code generation project

In this case, the project emerges from the urgency of radical technical change. Several years ago, processors stopped getting faster, due largely to thermodynamic constraints, so the computer hardware industry started to turn to increasingly parallel architectures[428]. This preserved Moore's Law, but it creates a headache for programmers. Rather than having a machine that gets faster at doing a calculation as new hardware is developed, we have machines that can do more and more concurrent calculations at the same speed. Contemporary supercomputers comprise thousands of such “cores”. This is one of the biggest current headaches for large scale simulation, because running a simulation on eight cores as opposed to one will not give an 8-fold increase in speed if at any point within the calculation one processor has to wait for another to be finished with its calculation before it can proceed. This has been a huge problem, for example, for the UK Met Office GCM (General Circulation Model), and will only get worse as new architectures exploit possibilities for parallelization on a much larger scale. But these problems are nothing compared with what looks likely with the next generation of architectures, a change in technical condition, which may force fundamental changes in how computational scientists write and work with their codes. Looking towards this uncertain future, the Met Office is one of the organisations funding the AMCG automated code generation project, with a view to exploring possibilities for its next but one dynamic core rewrite.

> “The design brief is “must run on whatever [hardware] the Met Office is running on in 2019,” about which we and they have no clue. Well, we have very rough ideas about the general direction and one of the issues is that by 2019 the Met Office might not even get to procure the machine [currently they have their own 'in house' supercomputer used for both their weather and climate simulations]. There might be some big machine which is either a European weather resource of some description or it might be a UK national resource of some description. The Met Office is part of the Ministry of Defence... it is possible that they might have to share with other bits of defence. And in either case they might not get to design it. The performance portability thing we are trying to make work here is something that is going to become very important for them.” (WS)

“Portability” is one of the key concepts explored through this research. Investigation begins from the fact that exactly how software systems can best be rendered portable, and thus *durable* in the face of hardware variability, is not something that is well known. This research is intervening in the very heart of the accomplishment of computational science, studying the systems used, and their potential reorganisation in order to bring new technologies and new disciplinary specialisms into the fold. It is fundamental “science of method” research.

Computational science exists within a shifting landscape of technologies. Workability is not something that can be established and assured for the future. It must be continually maintained in a dynamic milieu, and as we have seen with these radical upheavals, past and future, in the nature of research itself, this whole field of practice is as dynamic as are its conditions. It is also not only a case of shifts elsewhere causing consequent shifts in research practice, but pre-emptive transformations based on early tremors and warning signs. Research responds not only to the open

428 Sutter, ‘The Free Lunch Is Over’.

future of its accomplishments, but also to the open future of the wider frame in which are seen feint glimmers of transformed conditions of its possibility.

8.9 Habitability

We can grasp these various dimensions of workability with the concept of “habitability” proposed by Richard P. Gabriel. Habitability captures something of the overall effect of contrary pressures, and the negotiations through which workability is maintained.

Gabriel's concept emerges from a critique of what he saw as an unquestioned and deeply problematic assumption, shared by many programmers, about what it is that software developers are doing. Most, he says, have a “monumental” model. The software is going to be a perfect and magnificent final product, and all the work that goes into it is oriented towards creating this product. But, he says, this isn't really representative. Very little software is ever finished and left alone. Most software is continually modified[429]. This change of emphasis flags up the importance of software as an environment for the developer's work, something that is often hidden behind concerns about the experience of the end user. “Habitability makes a place liveable, like home. And this is what we want in software – that developers feel at home, can place their hands on any item without having to think deeply about where it is”[430].

“Programs live and grow,” he says, and instead of being all about a monumental final product, “their inhabitants – the programmers – need to work with that program the way the farmer works with the homestead”[431]. A house is built, and then maybe a storage shed. Later, a barn is built, and then maybe an extension for grandma. The whole lot needs to remain coherent, but it also needs to remain open for further changes. You may, in the end, get a less effective configuration of buildings than if you had started with foreknowledge of all the requirements that would stack up over time. You may end up having to pull the barn down and rebuild it elsewhere because it no longer makes sense. But the point is that you cannot know beforehand how things will grow, and the important thing is not so much the efficiency of any “final” state of affairs (after all, at what point would we decide we have reached such a state?); it is that at each point along the way, as the structures lived and grew, the environment was effectively good to live in.

What Gabriel points out about software in general is even more true of scientific software. As Rheinberger has pointed out, change is an essential feature of productive research.

> “Unprecedented events are about things and concatenations not sought for. They come as a surprise but nevertheless do not just happen. They are made to happen through the inner

429 Matsumoto, ‘Treating Code as an Essay’, 478.
430 Gabriel, *Patterns of Software*, 11.
431 Ibid., 12.

> workings of the experimental machinery for making the future. And yet they may commit experimenters to completely changing the direction of their research activities”[432].

Gabriel echoes this sentiment, but for pragmatic rather than epistemic reasons. “Software needs to be habitable because it always has to change. Software is subject to unpredictable events: Requirements change because the marketplace changes, competitors change, parts of the design are shown to be wrong by experience, people learn to use the software in ways not anticipated”[433]. All the materialities discussed in this chapter speak to the *habitability* of the social and technical fields in which scientific research is done.

Scientific software is very rarely developed to any kind of specification[434]. Indeed, where specifications dominate software cultures, it is hard to see the creative side of development practice[435]. In opening up this inherent indeterminacy and openness, scientific software lets us see creativity in greater clarity. It serves as a very good example of the kind of thing Gabriel is interested in, the kind of code that is never finished, nor is it crafted all along by a pre-set idea of its destination. Easterbrook and Johns' study of the development practices around the UK Met Office Unified Model concluded that this work bore closest resemblance to elements of Open Source development and the Agile school of thought[436]. Agile methodologies are constructed around the premises set out in the influential “Agile Manifesto”, which is based on a rejection of predetermining projects with a prior specification, suggesting instead that it is better to keep things open, following an iterative cycle as the customer and developers revise their expectations in the course of things. Far from predetermining the product at the start of the development process, the manifesto's signatories declare that they “[w]elcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage”[437].

While it is not usually developed with any idea of competitive advantage in mind, scientific software is necessarily agile, because it is funded by lots of small projects that contribute iteratively to the building of the larger whole. It is agile because it is rare to set out completely clear goals for a project at the start. It is very interesting that one of the original authors of the manifesto, Brian Marick, recently claimed that “the Agile style of work is readily and satisfyingly described by the terminology of Andrew Pickering's *The Mangle of Practice*”[438]. It is revealing that an account of scientific practice, in which software did not play a major role, should resonate so with this movement, which was itself born from the acknowledgement of the idiosyncrasies of actual software practice, compared with the rigidities of the plans and goals managers attempt to formulate around it.

432 Rheinberger, *Epistemic Things*, 134.

433 Gabriel, *Patterns of Software*, 13.

434 Segal and Morris, ‘Developing Scientific Software’, 18.

435 Evens, ‘Object-Oriented Ontology’.

436 Easterbrook and Johns, ‘Engineering the Software for Understanding Climate Change’.

437 Beck et al., ‘Manifesto for Agile Software Development’.

438 Marick, ‘A Manglish Way of Working’, 185.

An agile approach is one that embraces change, and that embraces uncertainty, as parts of what makes the process of creating software interesting and productive. In contrast, for traditional, non-agile approaches, change and uncertainty are seen as threats to success. This fits beautifully with the aspects of scientific practice that I develop in the next chapter, which looks at the key role of openness to surprise in research. The malleability projects, allowing goals to be redrawn according to what emerges in phenomenotechnical practice, the “objective meditation” of work in the laboratory.

“Habitability” captures that side of workability that renders big software systems efficacious. It is when software development becomes an un-inhabitable environment that its efficacy as a material to think with is compromised. And these threats come from all sides. “Software is hard”[439]. Threats come from the success-driven expansion of software rendering it brittle. But they also come from the reaction to this, from the wholesale bureaucratisation of research practice. They come from the growth of the research group and its response to its changing formation. But they also come from outside, from technical conditions originating elsewhere in industry, in the wider world to which scientific practice is intimately connected. The experience of scientists at AMCG seems to me to be one of treading fine lines, working to preserve that space in-between, a balance in which good creative research can continue to be done on a daily basis.

8.10 Conclusion

The sense of “materiality” here has been necessarily vague, spanning concepts such as “workability”, “habitability” and “brittleness”. Each of these plays on intuitive metaphors, attempting to grasp something of the tangible reality of doing scientific research. The materiality in question is something akin to that described by Rheinberger, when he states that “[m]y emphasis is on the materialities of research... I would like to convey a sense of what it means for the participants in the endeavour to be engaged in epistemic practices, that is, in irrevocably experimental situations”[440]. It is this sense that I have been attempting to convey in this chapter, but to build on Rheinberger's approach, because it is the distinctive materialities of different fields of research that we can capture by complementing the existing studies of laboratory practices with new studies of *what it is like* to do science with software.

Habitability extends the concept of workability and of Knuuttila's conception of models as epistemic artefacts. It draws these concepts closer to a framework of practice because it moves us away from looking at a model as a singular thing, with its own properties, and towards a broader environment in which research *takes place*. This environment brings together technologies, artefacts, concerns, and embodied orientations into the comprehensive field of practice that is constitutive for the unfolding of research in real time that we may call its practical reason.

439 Donald Knuth, quoted in Rosenberg, *Dreaming in Code*, iii.
440 Rheinberger, *Epistemic Things*, 26.

9 Stability and Surprise

9.1 Introduction

The previous three chapters have taken a tour around the various materials (visual, textual, computational) that furnish the computational science laboratory with the means to work and to get a grip on itself. In this final chapter we return to some key theoretical questions: What is the relationship between the stability of the systems of research that get assembled around scientific projects and on the other hand the power of that research to destabilise, to produce surprising results? In grasping surprise, novelty, and the lacks and absences through which new and surprising events emerge, we reach to the heart of the temporality of scientific practice, and start to find tentative grounds that set science apart from other activities.

The first three sections look at how stability is produced in the laboratory setting, starting with an analysis of two images of technology theorised by Gilbert Simondon. I use these images to conceptualise the technical dimension of science in terms of the regularities of practice. I then will relate this discussion to one of the central aspects of scientific work, the production of stability by the careful measuring and balancing of the unknown. The unknown threatens stability by introducing a precariousness to research, but through incorporating these instabilities into its fold, capitalising on their potentialities as generators of surprises, modelling becomes efficacious as a research practice. It becomes hopeful and exciting, a generator of innovations.

The remaining sections attempt to draw out the ways in which surprise irrupts against this background, firstly in terms of the epistemic thing. This concept of Rheinberger's is the very material embodiment of the productive unknown in research, the thing that gathers attention around it, the motivating locus of promise for the future of research. While approximations serve as a mundane unknown used to construct systems of investigation, the epistemic thing is far from mundane, a very different sort of unknown, an inspiring and inviting unknown. With this concept, we grasp research's primary orientation, towards an open future of unanticipated events. The following sections analyse how such events can be identified, and trace their philosophical significance, leading towards the conclusion of the thesis as a whole, in which we take stock of the widest conceptual implications of the argument. The final consideration that brings us towards these grander conclusions is whether a concept of novelty in the present opens the door to a new concept of the scientific real, a capacity to be other immanent to things, displacing ideas of a hidden domain of nature to which we have only partial access.

9.2 Regularity and technology

"Research produces futures," says Rheinberger, "and it rests on differences of outcome. In contrast, technical construction aims at assuring presence, and it rests on identity of performance"[441]. Rheinberger will thus criticise ideas of technoscience for "the tendency to lump together what should be understood in interaction"[442]. He points to the intuitive difference: technical activities produce reliable pre-determined outcomes, whereas research opens a space in which what happens is in some fundamental respect unanticipated. This interplay of repetition and difference is at the heart of scientific research, but it is an interplay, not a complete merger[443]. Technology is found throughout science, but what is its role in practice?

The first thing to note is that we must be cautious of assigning a singular essence to the technical. In a piece unpublished in his lifetime, Gilbert Simondon identifies two forms of technology that provide powerful "cognitive schemas" for thought: "technology manifests in successive waves a power of analogical interpretation that is sui generis; indeed, it is not hemmed in by the limits of repartition of essences or of domains of reality"[444]. Practice is technical, following this kind of analysis, where it exhibits a technical form. The first schema such technologies provide is that of the Cartesian mechanism:

> "A building, stone upon stone, row upon row, in a transfer of the "certum quid et inconcussum", – the resistance of the stone of the foundations – all the way to the top, through successive levels that each act as the foundation for the immediately following higher level. This intelligibility of the transfer without losses that mechanizes ideally and analogically (but also in reality, by virtue of the Cartesian conception of knowledge) all the modes of the real, applies not only to the RES EXTENSA but also to the RES COGITANS: the "long chains of reasons" carry out a "transport of evidence" from the premises to the conclusion, just like a chain carries out a transfer of forces from the anchoring point to the last link "[445].

Not only do machines provide the background of reliability against which scientific activity can make its mark, they also provide the resources for thinking of reliability in the first place, for thinking about consequence, transfer and repetition. But practices that are completely mechanically

441 Ibid., 31–32.

442 Ibid., 31; see, for example, Haraway, *Modest_Witness@Second_Millennium.FemaleMan_Meets_OncoMouse*; Law, *Aircraft Stories*; Hayles, *Nanoculture*.

443 Rheinberger, *Epistemic Things*, 79; cf. Deleuze, *Difference and Repetition*.

444 Simondon, 'Technical Mentality', 18.

445 Ibid.

regular harbour little of the openness of research. Research is indeed highly regulated, but it is also excessive. It can take unpredictable twists and turns as the things in its domain manifest their capacities to take researchers off guard, to surprise them and lead them down fresh avenues[446].

> “We are confronted with a seeming paradox: the realm of the technical is a prerequisite of scientific research. On the other hand and at any time, the technical conditions tend to annihilate the scientific objects in the sense attributed to this notion. The solution to the paradox is that the interaction between scientific object and technical conditions is eminently nontechnical in its character. Scientists are, first and foremost, *bricoleurs* (tinkerers), not engineers. In its nontechnicality, the experimental ensemble of technical objects transcends the identity condition of its parts”[447].

This passage needs some commentary. The bricoleur and engineer are figures from Levi-Strauss's famous essay *The Science of the Concrete*, in which he opposes mythical thought (the bricoleur/a tinkerer or handyman) to Western thought (the engineer). Levi-Strauss says:

> “Mythical thought, that 'bricoleur', builds up structures by fitting together events, or rather the remains of events, while science, 'in operation' simply by virtue of coming into being, creates its means and results in the form of events, thanks to the structures which it is constantly elaborating and which are its hypotheses and theories”[448].

Rheinberger swaps things around. For him, the scientist is the bricoleur, and it is bricolage that produces events[449]. The reason of science is not a rationalisation, an elaboration of structures. Nor is it a structure built from the empirical. It is a creative tinkering making space for novelty. Scientific work is indeed carried out through technical apparatuses, machines, data-readings, and all the regularities and repetitiveness of the mechanistic schema. But as bricolage, it subjects these many regularities to a procedure, a tinkering, a play, that breaks the repetition, concerned less with the extension of structures, but rather with creating a play between their many dimensions such that surprising and interesting results may be on the cards.

The mechanism schema captures the efficacy of techniques and routines in the laboratory, procedures that can be “black-boxed” because they harbour little further interest. But we should not leave our discussion of the technical here. Science is technical not simply because it employs technologies. It is also technical because what is set in motion by practical processes of bricolage are systems for research, complex dynamic wholes of techniques and procedures that exhibit a kind of self-regulation. Here we start to tap into an image of technology that is much more contemporary: the cybernetic system.

> “Cybernetics, which was born from the mathematisation of the automatic regulation apparatuses [dispositifs] – particularly useful for the construction of automatic equipment of airplanes in flight – introduces into this the recurring aim of information on a relay apparatus

446 Latour, *Aramis*, 72.

447 Rheinberger, *Epistemic Things*, 32.

448 Lévi-Strauss, *The Savage Mind*, 22.

449 See also Knorr, ‘Tinkering Toward Success’.

> as the basic schema that allows for an active adaptation to a spontaneous finality. This technical realization of a finalized conduct has served as a model of intelligibility for the study of a large number of regulations--or of regulation failures--in the living, both human and non-human, and of phenomena subject to becoming, such as the species equilibrium between predators and preys, or of geographical and meteorological phenomena: variations of the level of lakes, climatic regimes”[450].

While mechanism gave an image of regularity, cybernetic systems give an image of system-level relative regulation, extending to unstable and mutating systems that remain systematic insofar as they maintain relative coherence within a changing environment. Here, rather than being the mechanical return of the same, it is the interplay of stability and instability that is critical. Whereas the mechanism schema gave an image of complete transferral of forces, this system-based schema embraces the complexity and mess of the world, accounting for the ability of systems to maintain integrity despite changing pressures from the outside, or to shift as a result towards new regimes of stability. Here it is the system of research as a whole that has a self-reproducing rhythm, not any single constituent part. For Rheinberger, one of the key features of experimental systems is their self-reproductive capacity, dynamic systems endowed of their own “intrinsic time”[451]. The schema is readily apparent in one of his most evocative passages:

> “Basically, there is no all-encompassing theoretical framework, no overarching political program, no homogenizing social context effective enough to pervade and coordinate this universe of drifting, merging, and bifurcating [experimental] systems. Where the systems do get linked, the links do not form stable connections; rather, transient interfaces are generated by the differential reproduction of the systems and the constellation of their ages. There is no common ground, source, or principle of development from which they would all spring, no hierarchy in which they would all be encapsulated. The constitution and constellation of differently aged experimental systems as a whole is u-topic and a-chronic. It is a de-centred reticulum, a rhizomatic structure in which connecting capillaries and anastomoses constantly are formed and dissolved and where attractors permanently shift”[452].

When we look at the overall self-reproduction of systems of research, we find no straightforward repetition of the same. While a stabilised piece of laboratory apparatus may provide a reliable measure every time it is used, the investigative milieu of which it is a part is much more dynamic. It self-regulates in response to many different kinds of influence and tends towards local regimes of stability. It own internal trajectories, novel results obtained in another laboratory, a political intervention or an industrial invention may well cause great ripples, as practices reorientate with respect to the new circumstance. But the coherence of their system is manifest in their eventual assimilation of the difference, finding its place within other constellations of concepts and techniques.

450 Simondon, ‘Technical Mentality’, 18.
451 Rheinberger, *Epistemic Things*, 180.
452 Ibid., 181.

For Rheinberger the important thing about an experimental system is that it maintains itself in a region of metastability. The dynamics of research involve a fine balance between regularity and difference, and it is here, far from equilibrium, that we see the possibility of the crucial open future-bound striving that is at the heart of science.

> "To remain productive in an epistemic sense, an experimental arrangement must be sufficiently open to generating unprecedented events by incorporating new techniques, instruments, model compounds, and semiotic devices. At the same time it must be sufficiently closed to prevent a breakdown of its reproductive coherence. It has to be kept at the borderline of its breakdown"[453].

Productive research occurs in arrangements that are composed of well-studied elements, reliable techniques, results and methods, but which also incorporate some quotient of objects or processes that are in important respects unknown, or at least unexplored in some fundamental dimension, harbouring a constitutive lack, a precariousness that is a source of threat and promise in equal measure. "If we accept the thesis that *research* is the basic procedure of the modern sciences," says Rheinberger, "we are invited to explore how research gets enacted at the frontiers between the known and the unknown"[454]. There is no need for us to get hung up on questions about the nature of knowledge here. The known and unknown manifest themselves in terms of the stable and the unstable, a play of presence and lack at the heart of practice.

9.3 Taming unknowns: approximations

Setting up a productive research situation means carefully opening a space of lack, of non-knowledge. Research takes place in circumstances where there is enough unknown, or only vaguely known, to create a potentiality for surprise, but not so much that research practice is threatened by chaos. This productive situation is "metastable", a precarious region of relative stability existing between poles of order and disorder, in which the system of research is kept from descending into chaos by self-regulating processes[455].

Against popular misconceptions that scientists deal in proof and certainty, Keller asserts that "uncertainty and doubt are the daily diet of scientific researchers "[456]. In numerical simulation this situation is maintained by a number of measures that incorporate and "tame" ignorance, that domesticate it so that it becomes calculable and manipulable, that the right balance of determinacy and indeterminacy. At the heart of this is a complex scaffold of approximation that goes into creating a modelling framework. These "tame" errors made in the course of modelling are crucial in the way that this kind of research creates a space for surprise. In this respect, this discussion aims to build on the foundation established in chapters 6 and 8, in which we saw the constant attention devoted to

453 Ibid., 80.
454 Ibid., 25.
455 Cf. Prigogine and Stengers, *Order Out of Chaos*.
456 Keller, 'What Are Climate Scientists to Do?', 21.

dealing with “wild” errors, bugs which creep in and threaten everything with their unknown consequences for the project. Approximations are more deliberately introduced but they nevertheless require a great deal of work if they are to be kept manageable.

Computational science begins from a mathematical treatment of the systems being modelled. An ocean, or river, or industrial apparatus, is idealised to the extent that it is taken to embody the solution to a system of equations. But this idealisation is not a transfer to a mental realm; it occurs in the reciprocal embodiment of the system into a series of mathematical and computational systems, rendering it manipulable for the researcher. The constituent imprecisions in this process are readily apparent in the simulation of fluids, because of the coexistence of different mathematical frameworks (molecular mechanics, continuum mechanics and even potentially quantum mechanics). The continuum equations are known to be inaccurate on certain scales of motion. They are also generally analytically intractable. Numerical simulation, to put it simply, is the use of computational methods to gain good approximate solutions to intractable problems. In this sense, it is possible to claim that all continuum simulations are wrong[457].

> “In a way, your fluid is made up of lots and lots of little molecules, so it is a discrete system. You could, if it was possible, model each of those on a computer but there are way too many. So you make an approximation that these molecules are acting like a continuum. And then again [when you discretise these equations for the computer] you are moving back into a discrete system. It is only on really small scales that the first approximation breaks down.” (AY)
>
> “I think the pure way of looking at it is if you associate the model with being the pure idea, the pure theoretical model. And if we are talking about oceans or nuclear reactors that is typically your partial differential equations – your calculus – something described in terms of calculus and in terms of the continuum. Then we discretise that model into a numerical discrete model, where we have rewritten it in terms of some discretisation or quanta... which makes it amenable to being run on a computer, makes it finite and so on. By this stage you have already incurred lots of errors. Your new numerical model is therefore only valid over a limited degree of physics because you had to make some compromises, and that is the point where you can program directly.” (NK)

Approximating starts from the very beginning of modelling, not just in terms of how to solve the equations, but also in terms of which equations, which “bits of physics”, are deemed appropriate to the problem in question, and hence what needs to be approximated in the first place.

> “There is a quotation from Einstein that a model should be as simple as possible but not too simple. You are never going to want to throw in all the physics. It doesn't make sense if you are looking at an ocean problem to throw in a quantum physics model. The point is that you start off with your simplest possible model and then you throw in additional terms.” (TT)

457 Turkle, *Simulation and Its Discontents*, 81.

This exercise in building up approximations is not just a matter of pre-empting what will be needed. There is an experimental dialectic at work between the implementation of the model and the measurement of its imprecisions that feeds back into a process of revision and extension.

> “So you look at where are the biggest mistakes that you are making, then you introduce that bit of physics, so your are increasing your model incrementally, and with that your model becomes more complex and more expensive to execute. So there is a balance here between feasibility and what errors you are willing to tolerate. We are dealing with a finite machine: people say we never have enough computing. You are always adding in that extra bit of physics that you will still be able to compute on the machines that you have.” (TT)

The continuum model of the problem, already an idealisation, is a huge generator of unknowns, and the exploration of this space involves constructing computational models that themselves balance further unknowns against each other. Approximations are made in several distinct ways, and the estimation of the error involved in each is one of the major themes of the science of method. In essence, there is no point in gaining accuracy in one part of the process if it will only be eclipsed by the cumulative effect of another error from elsewhere. During the 2010 training course for Fluidity developers, the overall error was broken down into four constituent components:

1. Spatial discretisation error: the error consequent from the division of the continuum of space into a mesh of polyhedra. Having more, smaller polyhedra would be a way to reduce this error, but this of course will require additional computing time.

2. Temporal discretisation error: the error from the division of time into a series of discrete steps. Taking smaller steps would likewise be the initial way to reduce this error. Again, this places more demands on the available computing resources.

3. Geometric error: the error that derives from the approximation of structures in space by boundaries within the mesh. For example, the sea floor and coastline in oceans simulations are highly topographically complex, and will be imperfectly resolved. On the other hand, some kinds of simulations will have a very small geometric error, when for example the problem is already an idealised one (e.g. backwards facing step), or when the domain modelled is particularly amenable to geometric approximation, such as in the case of some kinds of industrial machinery. Reducing this error can be done by gaining a better fit to the geometry, but this may be constrained by the information available rather than by the capability of the mesh to accurately resolve the features.

4. Algebraic error: the simulation ultimately comes down to the solution of a massive set of linear equations. These often can be solved precisely, but because an exact solution at this point would itself be only an approximation, algorithms are used which save computer resources by providing an approximate solution. What degree of accuracy is required is a question balanced against the errors already present from the other three factors.

The study of approximations is a science of technical practice, research into the methods used to build computational models. Everything is bound to the fundamental practical issue of limited computer power. The driving principle is not so much to gain as good an approximation as possible, but rather to gain a “good enough” approximation for as little computer power as possible.

> “Ultimately everything boils down to solving a matrix. All the difficulties are captured in the word “approximation”, with linear solvers solving a big set of simultaneous equations and each of those are representing a physical process. You need to prioritise which ones are important, which ones to focus on more. The computer is very good at solving a linear matrix of simultaneous equations so we need to find a way to bring it down to the set of simultaneous equations that we think is the fastest to solve.” (AY)

Figuring out exactly what kind of stresses a given problem places on the scaffold of approximations is one of the key processes in this kind of research.

> With a wide aspect ratio problem it could be that some values are very large and others are very small. You can also end up trying to solve for values that are very close to each other so it is more important to do it accurately. Even in trying to solve simultaneous equations you are again trying to make an approximation because usually it is too expensive to solve it exactly, although you can solve it exactly. (AY)

Many different kinds of techniques have been created and subjected to wide ranging analysis for the role they can play in balancing speed and error.

> There are multigrid methods – geometric multigrid where you coarsen the domain, solve over that and then use that information to solve the next level down, and then down and down. There is some question about how much you iterate each level and whether you go down and up again or down and down to the smallest level. Fluidity has an algebraic multigrid which is similar but not in the spatial domain – you coarsen the matrix: you just give the solver a matrix and it might take out all the large values and then use that as a precondition to solve the next level down” (AY)

The key thing that renders this kind of error manageable and manipulable is its quantification. When working with algorithms, the error can often be calculated purely on paper. Other errors can are identified through more “experimental manipulations” and can only be approached by running simulations and comparing their results with other data[458]. For example, it is very common to run the same simulation several times with different mesh resolutions: some with a very fine mesh and then with coarser and coarser elements. This yields a statistical measure of how much the spatial and geometric errors are contributing to the general mix, and allows the researcher to find a good balance of speed and accuracy for subsequent simulation runs.

> “There is a trade off between computational costs and error that I would allow. So I can say “at this very fine resolution I get 85% the same as this other result but if I use a coarser resolution I lose some of the accuracy but it runs in half a day so if I use that I can do all these other runs and therefore I have used this one, the coarser resolution.” There is no point

458 Galison, ‘Computer Simulations and the Trading Zone’, 137.

> in running a model which is very accurate and produces the real world or whatever you want beyond the accuracy that you even need and takes three months run. That is no use to no-one. Because after all you are modelling. It is a representation of a process rather than an actual process." (AN)

Developing novel strategies for modelling is often a matter of finding new ways to reduce errors which are comparatively computationally cheap. AMCG's flagship discretisation scheme is P1DGP2, an element type which combines piecewise linear discontinuous functions for pressure with continuous quadratics for velocity.

> "Even though you are approximating a continuous function, if it jumps on a level that is smaller than the grid scale, you can represent it better with a discontinuous function than with a linear interpollant. What you can do instead with discontinuous elements you can get a much better answer." (KU)

Other examples of quantifying error can be found in validation, where data output is compared with some empirical data. But even this data carries along its own sources of error, which also need to be folded into the mix.

> "The problem is that you have errors on the parameters in a lab, so you might turn on the fan at 1 m/s but how accurate is that? We are talking about little tweaks so you have got that source of error you have got to contend with" (QS)
>
> "If you say "I am 1% out from the right answer," people say "The right answer? What is the right answer?" because I am comparing to experiments which in turn have an error or other models which have an error." (AN)

Processes for quantifying error themselves harbour their own sources of error. A quest for total stability would be led down an infinite regress. But the key thing is that the system of research be made relatively stable, that it be stable enough. It may still be vague. It may still be unruly, but it is a manageable unruliness, a productive unruliness ready to be set into practice.

There is, however, between stability and instability, a more fundamental, more profound lack that underlies the whole enterprise, that gives it its drive and its fortitude, that orients it towards the future of its own unfolding. This is not a lack in the sense of an unobtainable entity, but rather a present absence of the thing that focusses attention, that sets the wheels of investigation in motion, that thing whose power, whose potential to be other, drives forward the very exercise of research. This is what Rheinberger calls the "epistemic thing".

9.4 The epistemic thing

The epistemic thing is the embodiment of the unknown as a locus of concern and attention, the central pivot for a scientific project. While the unknowns addressed above are subjected to procedures of management and measurement, carefully balanced as part of the research's own selfregulatory action, here the unknown is embodied in a thing, which provides a window onto an open future, a thing which inspires, motivates and drives investigation onwards.

The epistemic thing may help fill a gap in practice theory; in the 1980s, Sherry Ortner wrote that a "theory of practice requires some sort of theory of motivation "[459]. With the epistemic thing, we start to grasp practice as something that involves orientations within a field of techniques, bodies and signs, an affective dimension that speaks of the way in which research is driven somewhere, heading somewhere, kept dynamic and hopeful, motivation beyond the planning for the future, beyond intentionality and beyond interest. It is the absent centre of scientific practice, the seed for the crystallisation of concern.

For Knorr-Cetina it is this orientation towards things, and the interaction of things and practices associated with them, that marks research apart from other kinds of work. "Research work seems to be particular in that the definition of things, the consciousness of problems, etc., is deliberately looped through objects and the reaction granted by them"[460]. In this respect, Rheinberger's theory of epistemic things participates in the "object turn" in contemporary theory[461].

The epistemic thing is the focus of research. It is this thing's occurrence within a structure of concern which orients practice. It is important to note, however, that what Rheinberger is interested in is not a present Cartesian object. One of the classic ways to displace such a concept and to open up a more subtle understanding, is to turn to Heidegger's philosophy.

An epistemic thing is not what Heidegger called a present-at-hand object[462]. It is not a spatio-temporal object unambiguously present and available for scrutiny. Such an object is hardly going to embody the kind of potential for surprise that is interesting for scientists. It is epistemic insofar as its presence and coherence are only fleetingly guaranteed. It is a risky object. Its existence is manifest in the potentiality to gather the concern of researchers around it, to co-ordinate efforts around a promising but non or not yet present centre, the promise of an immanent future. It is less an object than a "thing" in the sense that Heidegger gave the term in one of his later essays:

> "The Old German word *thing* or *dinc*, with its meaning of a gathering specifically for the purpose of dealing with a case or matter, is suited as no other word to translate properly the Roman word *res*, that which is pertinent, which has a bearing... In English "thing" has still preserved the full semantic power of the Roman word: "He knows his things," he understand the matters that have a bearing on him; "He knows how to handle things," he knows how to go about dealing with affairs, that is, with what matters from case to case; "That's a great thing," that is something grand (fine, tremendous, splendid), something that comes of itself and bears upon man"[463].

459 Ortner, 'Theory in Anthropology Since the Sixties', 151.

460 Knorr-Cetina, 'Objectual Practice', 184.

461 See, for example, Marres, *Material Participation*; Lash and Lury, *Global Culture Industry*; Bennett, *Vibrant Matter*; Brown, 'Thing Theory'; Pels, Hetherington, and Vandenberghe, 'The Status of the Object'; Preda, 'The Turn to Things'.

462 Heidegger, *Being and Time*, 68–69, 99–102.

463 Heidegger, *Poetry, Language, Thought*, 173.

This displacement of the Cartesian object, as object of knowledge, of perception or experience, for something more entwined in the structures of an already constituted world, has been picked up by Latour, who opposes things ("matters of concern") to objects ("matters of fact")[464]. "What would happen, I wonder, if we tried to talk about the object of science and technology... as if it had the rich and complicated qualities of the celebrated *Thing*?"[465] It is important to note that Heidegger believed that science only dealt with things as objects, as objects to be fixed in a knowing gaze, torn apart, charted, examined and depleted[466]. But he did not give scientific practice the same depth of attention he invested in artistic or architectural practices. It goes wholly against his intention to enlist his discussion of the thing into a theory of scientific research, but nevertheless I would maintain that this is where his philosophy can be put to the best use. But what is the relationship between these elusive things and the potential to surprise that is to important to the dynamics of research?

When he put forward his theory of the tacit dimension of scientific practice, Polanyi heralded it as the key to resolving Meno's paradox, a dilemma that had famously been posed by Plato[467]. Likewise, this paradox provides an excellent window into the theoretical function of the concept of the epistemic thing. How do you recognise a new productive problem that is deserving of your scientific attention? How do you find an avenue to explore worthy of concerted effort? Either the problem is recognisable, in which case it is trivial since being already known sufficiently to be recognised it is thus not authentically new, or the problem is unrecognisable, in which case it would be genuinely novel, but ungraspable in our present situation. How do you grasp an unknown as that thing towards which your project will strive to realise? In contrast to Plato's theory, in which he resolves the paradox by claiming that new knowledge is never really new because it is gained through anamnesis (literally "unforgetting"), Polanyi espouses a theory of embodied, tacit knowledge, "tacit foreknowledge of yet undiscovered things"[468]. There exists an informal, unspeakable side of practice resistant to being put into words, that nevertheless operates as the strongest of guides.

> "The anticipation of discovery, like discovery itself, may turn out to be a delusion... To accept the pursuit of science as a reasonable and successful enterprise is to share the kind of commitments on which scientists enter by undertaking this enterprise. You cannot formalize the act of commitment, for you cannot express your commitment non-committally. To attempt this is to exercise the kind of lucidity which destroys its subject matter."[469]

Rheinberger's theory of the epistemic thing does much of the same work as Polanyi's notion of tacit knowledge, with the important difference that while Polanyi is mainly concerned with embodiment, Rheinberger also wishes to stress the constitutive role of material practices:

464 Latour, 'Why Has Critique Run Out of Steam?'.

465 Ibid., 233.

466 Heidegger, 'The Question Concerning Technology'.

467 Polanyi, *The Tacit Dimension*, 22.

468 Ibid., 23; it is also worth noting that the anamnesia solution can still be accepted, in the work of Bernard Stiegler, for whom anamnesia captures precisely that manner in which large amounts of knowledge is implicitly stored in technical systems, which are for him, memory systems, tertiary retentions in the terms of Husserl: Stiegler, *Technics and Time*.

469 Polanyi, *The Tacit Dimension*, 25.

> "It is in the fabric of properly "tuned" experimental systems that scientific events materialize. It is in the nature of an event that it cannot be anticipated. Novelties are always the result of spatiotemporal singularities. Experimental systems are precisely the arrangements that allow scientists to create spatiotemporal singularities. They allow researchers to arrive at unprecedented, surprising results"[470].

Rheinberger aims to capture this sense of temporal unfolding, of the scientists' relationship to the uncertain in the future of their investigations. He thus writes of epistemic things, scientific objects which "present themselves in a characteristic, irreducible vagueness. This vagueness is inevitable because, paradoxically, epistemic things embody what one does not yet know"[471]. He stresses the actual openness and vagueness of things. Lack of knowledge has long been considered a primary motivation for activities of model building[472], but Rheinberger will extend this principle as something at the heart of research in general.

The vagueness, or lack of presence of epistemic things is not the kind of vagueness that inhabits everyday experience. It is a particular vagueness that is the result of painstaking work in assembling systems of investigation around a common core, which concerns scientists in that it is suggestive of a door onto a future of new findings, new ways of doing things, new concerns. It draws attention in and motivates practice. Knorr-Cetina describes it as "a lack in completeness of being that takes away much of the wholeness, solidity, and the thing-like character they have in our everyday conception"[473]. This theory of epistemic things draws our attention to the nonplaces of research, the implicit foundation, the lack that serves as wellspring of exploration and experimentation[474]. It requires us to look at the plurality of practices in science, because epistemic things exist as such only inasmuch as a system of practice is assembled around them. We thus replace our understanding of the object of science as something objectified with an epistemic thing as a nexus of concern that by definition eludes objectification.

Epistemic things in applied modelling are constituted where computational science borders empirical studies and experimental practices. Oceanographic studies of ocean currents use global complexes of sensing apparatus to trace temperatures, velocities and salinities across the oceans. These provide an indication of the types and variations of currents and an indication of the mechanisms that drive them. Experimental studies also feed in to this area, although it is very difficult to scale down these kinds of phenomena such that they can be represented in a laboratory. Numerical simulations bring another, very different kind of approach to the table, conceptualising the ocean first and foremost as a mathematical system whose state space can be explored and probed through approximate solution on the computer. Epistemic things inhabit this between-space, this middle ground between practical systems of investigation that pose their object in very different manners and that probe it using very

470 Rheinberger, 'Experimental Complexity in Biology', S246–S247.

471 Rheinberger, *Epistemic Things*, 28.

472 For example, Ehrenberg, 'Models of Fact: Examples from Marketing'.

473 Knorr-Cetina, 'Objectual Practice', 190.

474 Cf. Bosteels, 'Nonplaces'.

different techniques. They embody the practical articulation of different kinds of research, the effect of which is a shared sense that despite differences, something exists in common between them. There is no privileged representation of the epistemic thing because none of the practices that are co-ordinated around it has the authority to speak for all of them. Thus its curious non-presence that nevertheless motivates research.

Applied modelling, however, is not the only kind of research that gets done in computational physics. The other kind is what I called in section 2.14 a science of method, a science of modelling techniques. Here we see epistemic things emerge from technical conditions, and because these studies directly feed into the repository of modelling techniques, epistemic things often eventually return to the technical background. They are, however, like those in applied modelling, co-ordinated with other practices, with other kinds of modelling practice, with computer science, and with areas of applied mathematics such as linear algebra and graph theory.

In the research documented in this thesis, epistemic things exist where research practices assemble coherent systems of investigation around a focal point of concern. In computational science, this usually means bringing together and working on an assemblage of techniques within and outside the modelling framework, assembling data sets and qualitative descriptions of phenomena, mathematical results, hardware resources, supporting infrastructures, a group of scientists with relevant specialisms, and iterative systems of planning and management of the unfolding research. Epistemic things exist on many scales; they co-exist and interact. The investigation of the Severn Estuary that we encountered in chapter 4 opened onto an investigation of wetting and drying methods, and later turned into a study of optimisation techniques, eddies of “science of method” research that spun off from the central stream of research. This seems to be a very common feature of this kind of research, where the central focus point of work shifts as the investigation of the original problem reveals certain techniques to be crucial factors. These techniques, not yet reliable as mechanisms for the study of the original problems, invite exploration, and draw applied modelling into a science of method. Eventually, their productivity as a nexus of concern is on the wane and they can be concretised, “black-boxed”, their realisations in code becoming conditions for the exploration of further issues. Epistemic things may, as Rheinberger points out, “become obsolete as targets of research” but they may also “become transformed into stable, technical objects that may define the boundary conditions of further epistemic objects”[475]. The integration of what once offered open productive potential back into the circuits of the laboratory is an important process which keeps the systems of research dynamic. While it is very obvious that new computational methods will eventually become part of the technical background for further research, the same is also true of enterprises in applied modelling, which provide key test cases for grasping the effects (positive or negative) of future manipulations of the code, and which provide crucial validation resources, platforms of success to support the studies that follow.

475 Rheinberger, ‘Reply to Bloor’, 407.

At the very top level, almost all of the research at AMCG can be read as contributing, in one way or another, to an exploration of the Finite Element Method (FEM). In some scientific disciplines, particularly in structural engineering, this is one of the most well-established techniques available. Here it acts as a technical construction, routinely put to work to solve engineers' equations. It does not really harbour much mystery or potentiality. But research at AMCG transports this method in fluids applications, geophysical applications and radiation applications. Here it is much less certain what the effect of FEM will be. It remains to be seen what its effects will be when applied to many different problems. It is a nexus of hope to which all these scientists are committed, on which they stake their careers. We need to recognise that as much as it holds out promise, FEM equally harbours precariousness, the possibility that it turns out to have been a false hope, no longer interesting as a set of techniques. There may be fundamental obstacles that prevent it from living up to its heralding by scientists at AMCG as a "next generation" method. It may, alternatively, become uninteresting because it has been supplanted by other techniques. But more likely it will eventually become exhausted as an epistemic thing at the point where it has been extensively applied and studied, at which point it will continue to serve researchers but as a tool for studying other kinds of things.

In giving us this thing-based theory of research orientation, Rheinberger is perhaps the best contemporary heir to the historical epistemologists' theories of "the open-ended future in which new objects and systems of concepts are developed"[476]. His theory is productively read as going beyond Bachelardian phenomenotechnique, because while Bachelard stressed the temporality of scientific practice in his philosophy of error, his was a largely backward-facing and negative theory. Rheinberger on the other hand directs us towards the hope embodied in scientific orientations. Like other post- or non-positivists, both thinkers disavow any idea of a pre-set destination for scientific research, but Rheinberger gives us the sense of the inspiration such an open future presents, and lets us understand something of the motivation that drives research onwards.

9.5 Discerning error and event

The epistemic thing embodies the potentiality of the unexpected, the immanent future in research. In this respect, it gives us a thing-based theory of surprise. This idea of novelty is "social", in the sense that it locates novelty outside the creative mind, in the wider domain of practice. It also therefore moves away from the idea of lone geniuses. But in being social, it attributes central importance to indeterminacy of practice itself, and refuses to explain novelties as an inevitable product of a historical moment[477].

However, there is nothing straightforward about surprise. Events may turn out as novelties openings doors to new lines of enquiry and new ways of doing things. But this is not the only thing that can happen. They may also manifest themselves as problematic disruptions of the research process, as

476 Hyder, 'Foucault, Cavaillès, and Husserl', 123.

477 See Simonton, *Creativity in Science*, 3–13 for an interesting typology of perspectives on creativity.

threats to its stability. This two-fold character has been one of the most important results to come out of what has become known as the "sociology of expectations"[478]. Michael, Wainwright and Williams identify the affective side of this negative possibility as, "in counterpoint to the hopes embodied in expectations, a wariness that one might be embroiled in hype"[479]. A novel result might turn out to have been an error all along, and thus nothing warranting getting excited about. Hope and wariness, inspiring novelty and destabilising error are two sides of a coin that at the point of the emergence of an event is still balanced on its edge.

There is no simple or straightforward mechanism for the discrimination between good and bad surprises. This has long been pointed out in science studies[480]. For Polanyi, one major aspect of laboratory routine is the suppression of surprising results, their interpretion as errors. If what Rheinberger calls "unprecedented events" are going to emerge, it must happen against the grain of this tendency.

> "The process of explaining away deviations is in fact quite indispensable to the daily routine of research. In my laboratory I find the laws of nature formally contradicted at every hour, but I explain this away by the assumption of experimental error. I know that this may cause me one day to explain away a fundamentally new phenomenon and to miss a great discovery. Such things have often happened in the history of science"[481].

In the next section I provide an example from AMCG, CE's study of discretisation schemes, in which we see the initial response to a surprising result being the assumption that this was a sign that something had gone wrong in the set-up of the simulation. Even with skill and experience the discrimination between events and errors is provisional and precarious. Hacking points to the role of practice in this context. He argues that this skill cannot be based on the scientist internalising an exhaustive theory of experiment. It cannot be explained by the idea that biologists, for instance, when they use microscopes, have a theory of optics that lets them see the artefacts as artefacts.

> "Hardly any biologists know enough optics to satisfy a physicist. Practice – and I mean in general doing, not looking – creates the ability to distinguish between visible artifacts of the preparation or the instrument, and the real structure that is seen with the microscope. This practical ability breeds conviction"[482].

The skill of distinguishing the two is not something that can be looked up in a rule book, or explained by reference to theoretical knowledge, but rather requires the kind of engagement with the environment that develops practical sensitivities. And these moments are always to some extent irreducibly tenuous, precarious, improvised. At the heart of Bloor's famous symmetry principle is precisely this injunction, to avoid assuming that there is any outside objective ground for discerning

478 See, for example, Brown and Michael, 'A Sociology of Expectations'; Borup et al., 'The Sociology of Expectations in Science and Technology'.

479 Michael, Wainwright, and Williams, 'Temporality and Prudence', 377.

480 See, for example, Pickering, *Constructing Quarks*, 9.

481 Polanyi, *Science*, *Faith and Society*, 31.

482 Hacking, *Representing and Intervening*, 191.

between error and event[483]. The fact that one surprise ended up being a motor of success and another swept under the carpet as an instance of equipmental error cannot be explained by the one being real and the other an artefact. Such a tactic would assume a retrospective position that begs the whole question of how, in the actual happening of research, the line between the two gets drawn. To explore this work of discrimination of event and error, and to look further into the management of error, we need to look at the immanent unfolding of practice.

9.6 Surprising results in practice

> “[S]urprising events in research are often something to which scientists aspire in their activities since it means a window to new and unexpected knowledge. Scientific methods thus should allow researchers to surprise themselves as well as their peers”[484].

One of my informants' stories illustrates the role of surprise in practice nicely. CE is a member of the group who largely works on fundamental discretisation methods. While any narrative can be expected to fail in capturing the real indeterminacy of what happens in the research process, this account gives a glimpse of what could otherwise have been lost.

> “The thing about the finite element method is that you represent the solution, so the wind, velocity, temperature, whatever, as some kind of way of interpolating between points. And then once you have described what you want these functions to look like, so whether they have jumps between edges or whether they are quadratic or linear or whatever degrees of freedom there are, after that it is all pretty much well defined and you just throw that into the computer and the very flexible code generates the matrices to actually implement the method. So the actual properties of whether it is a good or a bad choice [of discretisation] depends on analysing mathematically how those functions interact.” (CE)

He went on to describe a particular result that was significant for the unanticipated nature of its eventual outcome: “In this case CK chose a particular way of discretising the winds and the pressure that makes it particularly nice for geophysical applications, so oceans and atmosphere and things like that”. CK's choice was based on an unformalised intuition. Looking at the equations, a particular discretisation seemed to harbour the right kind of potentiality. “You look at terms and you want them to have a sort of similar representation. He had the intuition at that level” (CE).

Later on, CE collaborated with his colleague, WS, in order to investigate the method further. “WS and I did some numerical investigations... and it was working kind of unaccountably well” (CE). This was an instance of a method which works well, and it can be shown that it works well, but nobody has as yet investigated exactly why it is so good. This is the case for many of the methods used at AMCG, but as NK pointed out, this kind of empirical reliability can be open to question. But on the other hand, pressures to generate new applications often lead to these avenues being unexplored.

483 Bloor, *Knowledge and Social Imagery*, 177.
484 Gross, *Ignorance and Surprise*, 1.

> “When you are developing algorithms you could in principle go and prove why it works so well. So there is suspicion of methods that work well but its not clear why. If you have an algorithm and proved something you know it is true. If you develop an algorithm and you demonstrate that it works well on that problem all you have demonstrated is that it has worked well on that problem. You don't know if it will work as well on other problems. You could sit down for a year and work out and prove why. But that isn't going to push you forward, so there is a conflict there. We could stop working on fluidity now in terms of developing it and just analyse it, improve it, optimise it.” (NK)

In the case of CE's discretisation, it happened that he stumbled on a surprising phenomenon that led him towards a proof, one which would not require this year of working, leading him instead on a shorter route.

> “Then what happened was that a few years ago... I had a spare week and WS suggested some tests that I could run [on the new discretisation] and in these tests it should stay steady. Typically there will be some oscillations and how steady it stays is a measure of how good it is for atmosphere and ocean modelling. So I coded it up, ran it in this environment in which it is very easy to throw these things together...”
>
> “I ran the code and nothing was happening at all. It was staying completely flat. Because it was so steady I actually called WS in Vienna [he was at a conference] and said “There is a bug in the code! Can you help me?” When we looked at it a bit more we realised it was actually completely steady. A few weeks later I was stuck [in a hospital waiting room] and had a pen and paper and managed to work out a proof of why it actually should be completely steady. So once we had that we were able to show a lot of other things about it and that led to getting the funding for a new postdoc”

The immediate response to the surprising steadiness of the result was to take it as a sign that the code was broken, but when it was investigated further this opened the door to a surprisingly neat mathematical treatment.

> “In general, if you are lucky you might be able to prove some sort of bound on the amount of jigglyness [of the line that should remain steady]. But these things are typically much much harder than if it happens to be absolutely perfect. It is easier thing to manipulate equals relations than inequalities [less than/more than]. Typically when you are working with methods you have to do a lot of messing around with inequalities. All the most famous proofs of this kind of thing are very difficult and involve a sequence of about fourteen inequalities.” (CE)

The new method was chosen because it seemed likely to produce some interesting results when investigated further. It embodied the right kind of lack of knowledge that gives research its momentum. The tests were chosen for their ability to open out this potentiality and put the method on display. The research created a surprising result, but as CE remarked, there is a fine line between success and failure. He was not over-reacting by immediately assuming that something had gone wrong. Bugs crop up regularly when it is the fundamentals of the code that are being manipulated. But rather than seeing this as a deficiency, we can regard precariousness as a necessary precondition for the production of genuine novelties, events which break with any preconception of what will happen. If research is so stable that errors could be ruled out, little space for novel and interesting results remains.

9.7 The philosophy of unanticipated events

Scientific practice occurs within a non-foreclosed space, one in which unexpected twists and turns are able to play out. As outlined in chapter 3, it is always possible to undermine novelty by reading backwards from the result and thus seeing its seed in the past. But in the real time of research, the surprising result is not present in embryonic form. Its potentiality is not the potentiality of any determinate occurrence. Rather than calling it a possibility, we could name it a virtuality, in the sense that Gilles Deleuze gave to the term, when he wrote that "the virtual is not opposed to the real; it possesses a full reality by itself"[485]. The virtual is supposed to free us from thinking in terms of identity and resemblance, and turn instead to difference and repetition.

> Such is the the defect of the possible: a defect which serves to condemn it as produced after the fact, as retroactively fabricated in the image of what resembles it. The actualisation of the virtual, on the contrary, always takes place by difference, divergence or differenciation. Actualisation breaks with resemblance as a process no less than it does with identity as a principle. Actual terms never resemble the singularities they incarnate. In this sense, actualisation or differenciation is always a genuine creation"[486].

The relationship between potential and actual has been a subject of philosophical debate at least since Aristotle. While Deleuze, following Bergson, injects creativity into reality itself[487], many other thinkers wish to conceptualise the new as a radical singular irruption. Alain Badiou, for example, presents us with a picture in which the event is an ontological rupture, a revolutionary break with the completely foreclosed and structured present situation[488]. This kind of theory of the event is the heir to structuralism, which originally established itself as a synchronic theory of the coherence of structures. It thus displaced rival evolutionary perspectives on language, a displacement that was won, however, "at the high price of ahistoricity"[489], says François Dosse, a detachment of structure and change "that led to aporias because the links between diachrony and synchrony were not set in any dialectical relationship"[490]. In contrast to this stark opposition between structure and revolution, with Deleuze, and with a theory of practice, we can think about temporality and change in more modest terms. For Foucault, too, it is as important to avoid overemphasising the radical irruption of events as it is to avoid overemphasising the coherence of structure.

485 Deleuze, *Difference and Repetition*, 263.
486 Ibid., 263–264.
487 Cf. Bergson, *Creative Evolution*.
488 Badiou, *Being and Event*.
489 Dosse, *History of Structuralism, Volume 1*, 48.
490 Ibid.

> "One can agree that structuralism formed the most systematic effort to evacuate the concept of the event, not only from ethnology but from a whole series of other sciences and in the extreme case from history... The important thing is to avoid trying to do for the event what was previously done with the concept of structure. It's not a matter of locating everything on one level, that of the event, but of realising that there are actually a whole order of levels of different types of events differing in amplitude, chronological breadth, and capacity to produce effects"[491].

From the point of view of practice, the situation of research is not one of a wholly fixed or structured world, but rather one full of mess, of complexity, of vagueness. It is a world that supports relative orderings through the practical subjection of certain aspects of it to strict scientific protocols, but which within productive research situations retains the capacity to surprise. Within the frame of practice, we can point not only to creative processes, but we can also talk of the affective significance of potentiality.

> "A potential discovery may be thought to attract the mind which will reveal it – inflaming the scientist with creative desire and imparting to him intimations that guide him from clue to clue and from surmise to surmise. The testing hand, the straining eye, the ransacked brain, may all be thought to be labouring under the common spell of a potential discovery striving to emerge into actuality"[492].

For Rheinberger, it is not quite right to call what will emerge "foreknowledge". The potentiality in question is less a property of a researcher's subconscious, and more a property of the technical and practical arrangement of research systems. Researchers are inspired by the systems they assemble, but this inspiration does not amount to straightforward intentionality.

> "Unprecedented events are about things and concatenations not sought for. They come as a surprise but nevertheless do not just happen. They are made to happen through the inner workings of the experimental machinery for making the future. And yet they may commit experimenters to completely changing the direction of their research activities"[493].

The things that are created and manipulated in these arrangements, like the methods that CE and WS were experimenting with, are *epistemic* because they harbour this kind of potentiality. Their reality is not assured because they agree with established categories of knowledge, but rather because they harbour future-bound efficacy. "The reality of epistemic things lies in their resistance, their capacity to turn around the (im)precisions of our foresight and understanding"[494]. Rheinberger's words recall those of Bachelard:

> "For science, truth is nothing other than a historical corrective to a persistent error, and experiment is a corrective for common and primary illusions. The intellectual life of science depends dialectically on this differential of knowledge at the frontier of the unknown. The very essence of reflection is to understand that one did not understand before. The non-

491 Foucault, *Power/Knowledge*, 114.
492 Polanyi, *Science, Faith and Society*, 14.
493 Rheinberger, *Epistemic Things*, 134.
494 Ibid., 23.

> Baconian, non-Euclidean, and non-Cartesian philosophies are historical dialectics that grew out of the correction of an error, the extension of a system, or the completion of an idea"[495].

It is the journey that is important. In terms that resonate with Heidegger's definition of truth as poesis[496], truth for Bachelard is a process, not a static relation. We never arrive at truth, but rather set it in motion through practice. For Bachelard, the entry of a innovative element in a scientific field is what drives the process of revealing past and existing knowledge in the light of error, and pushes research forward according to its inner rational dynamic, of phenomenotechnique.

495 Bachelard, *The New Scientific Spirit*, 172, translation modified.
496 Heidegger, 'The Origin of the Work of Art'.

10 Conclusion

10.1 Revelation and invention

One of the key effects of a practice theoretical rationalism is the sense it gives to the potentiality of research beyond the straight-jacket of the concept of revelation. Traditional theories of science as well as prevalent popular intuitions would posit the unknown as existing in a hidden realm, a latent nature waiting to be brought to light, bit by bit, by scientific progress. In contrast, the novelty to which a practice theoretical perspective directs our attention is one which does not originate in a hidden realm. It is not just new “for us” but new “in itself”, an irruption that undermines the very distinction between the two. It is a novelty immanent to an unfolding research process. As Rheinberger argues, this requires us to abandon the concept of discovery.

> “Science is an exploratory attitude toward knowledge about the world. Knowledge by revelation was the theological mode. When science emancipated itself from theology, it claimed—possibly it had to claim in order to do the job of secularization—to be the better revelator about nature. This is still reflected today in our mainstream theories of science, which are all about the “discovery” of “things.” Now I think it is time to stop conceiving of science in the borrowed mode of a secularized theology”[497].

This final issue brings us back to the issue of abduction we encountered in chapter 3. It is about how we relate to events once they have already happened. Badiou notes that statements that refer to the coming into being of events are spoken in the “future anterior”: what “will have been” the case[498]. From the point of view of what eventually happened, present processes are known in terms of their results, and thus even the indeterminate present can be spoken of as the past of a hypothetical future in which the outcome will have stabilised into some objectifiable resolution. The effect of this abduction is that the new can very easily be taken as an index of a prior reality “not yet” discovered, and the incompleteness of our projects of discovery indicating human finitude and the incompleteness of History.

497 Rheinberger, ‘Reply to Bloor’, 409.
498 Badiou, *Being and Event*, 398.

If one side of the positivist coin is the idea of a singular channel of scientific progress, the idea of discovery as revelation is the other side. In this view, all research is a matter of shifting the boundary between what is explicated, and what left latent. The result is a stratified ontology in which nature is a reserve, giving its “givens” but largely holding itself back from us, who patiently, according to our means and abilities, work to uncover its various domains[499].

If, in contrast, we start to regard the real in terms of a process, a capacity to surprise, we are working with a very different vision, in which a grand ontological partition is no longer necessary. For Polanyi, “[r]eal is that which is expected to reveal itself indeterminately in the future”[500]. A similar view found expression in Bachelard's writings too:

> “The philosophers, for their part, hold out to us the idea of communion with an all-enveloping reality, to which the scientist can hope for nothing better than to return, as to a philosophy original and true. But if we really want to understand our intellectual evolution, wouldn't we do better instead to pay heed to the anxiety of thought, to its quest for an object, to its search for dialectical opportunities to escape from itself, for opportunities to burst free of its own limits? In a word, wouldn't we do better to focus on thought in the process of objectification? For if we do, we can hardly fail to conclude that such thought is creative”[501].

These attempts to establish an alternative space for thinking about the creative unfolding of research resonate with some contemporary versions of non-representational realism in continental philosophy. Quentin Meillassoux, for example, has been pushing an intellectual project seeking to break out of the bounds of human finitude[502]. For him, we need to regard the capacity to turn out otherwise not as a sign of human inability to fully reveal nature, but rather as the primary reality of things. Using the concept of “facticity” to write of the lack of absolute foundation of situations of knowing, he asserts that we “must grasp in facticity not the inaccessibility of the absolute but the unveiling of the in-itself and the eternal property of what is, as opposed to the mark of the perennial deficiency in the thought of what is”[503]. If we grasp, at the heart of research, the careful opening of a space for indeterminacy, we must in turn be careful to distinguish between a regressive interpretation that reads such a space as a space for the revision of human knowledge, and a radical interpretation in which it is a space for the actual production of new things, new effects, new phenomena, new configurations which themselves harbour the potentiality be otherwise, to drive practice onwards, towards an open future of further research. The creativity of practice is the creativity of things as much as it is the creativity of humans; in practice the two cannot be separated.

In disqualifying a realm of hidden reasons lying beneath, Meillassoux is thus giving us a new kind of radically immanent ontology in which we “put back into the thing itself what we mistakenly took to

499 Cf. Heidegger, ‘The Question Concerning Technology’.

500 Polanyi, *Science, Faith and Society*, 10.

501 Bachelard, *The New Scientific Spirit*, 176. Note, however, that Bachelard’s term ‘objectification’ does not have quite the same connotations that we have been giving it through Bourdieu.

502 Meillassoux, *After Finitude*.

503 Ibid., 52.

be an incapacity in thought”[504]. It is only when the becoming other of epistemic things is taken as a sign of human deficiency, that it serves as an index of a latent domain from which the new emerged. If it is a property of the Real, we have no such problem, and there is no need to commit to a stratified ontology. This ontological reading of the real potentiality to be otherwise serves, therefore, as a principal candidate for a radical alternative to the secularised theology of positivism. What is at stake is a rendering of lack in a new way; rather than lack being a failure manifest at the disjuncture between two levels of reality, it becomes a generative kernel of difference manifest in a lack of closure, an opening, a dynamic real.

These questions are of course much older than positivism. The very notion of the inaccessibility of the “thing-in-itself” speaks of this fundamental ontological stratification between beings as they appear and beings as they are, a division that provides us with a view of the world in which simulation can only ever be subordinate to the “authentic” encounter with nature in experiment or empirical observation. This long echo of Plato's Republic is still with us. Throw the poets out of the Republic, for they deal in base imitations! The simulationists can follow them into exile, for all they do is create imperfect copies of the real thing. Nothing that is good and proper can follow from such pedallers of deceit, of appearances that fail to live up to the hidden reality beneath.

My account has aimed to shift such perceptions, to open a space for the appreciation of the creative dynamics of simulating, of modelling, of working with and among complex systems of software and data. Perhaps we should reverse the order of the issue, and ask: Given that simulating harbours a potentiality at least as profound as experimenting, what does this tell us about the nature of things? Does this indicate that reality is no hidden domain of the in-itself? That it is rather the capacity of things, in highly skilled circumstances of technical engagement, to be-other, a power of becoming that speaks of immanence not dualism? If so, practices of simulating may speak to the heart of Western culture and world-view, to its internal tensions and contradictions, its own resources to be other than what it seems.

10.2 Reason and representation

The route we took through this thesis analysed scientific practice in terms of the open future embodied in the potentiality of epistemic things to lead the meandering path of research along unanticipated and interested avenues of enquiry. It showed the inherent dynamics of research, folded upon itself in phenomenotechnical practice, self-articulations which open a field of potentiality that renders practice future-bound and hopeful. There are in this view some modest normative dimensions available, ways to approach questions of science as good or bad, research as more or less efficacious. Scientific practice requires highly developed systems of materials, techniques and skills, and it requires the maintenance of these systems' coherence under conditions of their own production of anomalies and disturbances (bugs and social problems) and of external technical change (in

504 Ibid., 53, emphasis removed.

hardware, for example). We can talk about the workability of different practical systems of research, how flexible and amenable they are to play, whether they exhibit the right kinds of balance between stability and surprise. In none of these respects can we judge research according to what it is doing, or where it is going, but we can grasp something of the efficacy of its intrinsic movement.

I have argued that computational science is no more disembodied than experimental practice, that it involves forging new articulations of materials and technologies in epistemic situations. Scientific software is no idealised system. It is a set of technologies for research, which have characteristic materialities. Work with software is always going to be embodied and social. This is important for detaching discussions of simulation from discussions of formal theories, a connection which is made all too easily because of how amenable source code is to techniques of formalisation.

I also argued that it is necessary to see simulation in the wider frame of the research practice in which it is made and put to use, that it makes very little sense to stand a simulation on its own as an object of enquiry. To this extent the object of my study has been simulating, rather than simulation. Both philosophy of science and continental philosophy have had a tendency to isolate simulation from its context. It thus appears to them in terms of a single relationship of representation, rather than as an element in technical and epistemic practices. The sub-title could have been "beyond simulacra".

Practice is the heir to post-positivism, but an heir which redefines itself in new terms, distanced from reactionary critique. It enables the unpacking of the greatest insights of the post- or non-Kantian tradition, a transcendental brought down to earth, inserted into the immanent unfoldings of material and bodily engagements that we call skilled work. Practice theory also enables us to speak again of terms once discarded for their association with the enlightenment: reason and representation.

We encountered a concept of reason that is immanent to the concrete unfolding of skilled engagements. Reason happens. Reason happens in constellations of bodies and things, in situations. This is a decentring of the human. The wider fields in which human existence plays out supplant the organism, subject, individual or soul as the locus for enquiry. The human is decentred but not rendered symmetrical in the manner of actor-network theory. This is still human reason, but human reason playing out in a milieu of people and things, reason that owes its efficacy to the disposition and distribution of that milieu.

Representation also shows itself as something that happens, as no master concept capable of capturing science in its scope, but as a feature of practical activities, of processes of modelling. It is revealed as a play of differences, a play of presence and absence in which source and target occupy the strange territory between Same and Other. Representation may well be one of the most important dimensions to the material practices of research, but representations are always more than representations. They are parts of complex fields of practice, and serve research in many ways, as substrates for its realisation, materials to work with, and reflections of its accomplishments.

What we are toying with here is a basic idea of process as the primary stuff of science, and all objectifications of that process, such as representations and discursive descriptions, are themselves practical ways in which scientists attempt to get a grip on what it is that they have been doing, a grip on this sense of movement that carries their work forwards. Truth is not manifest in the representation, but in the manifest insufficiency of any capture of research in accounting apparatuses. If there is a meaning to the concept “truth”, then it too is one in which truth happens. “But how does truth happen?”[505] Not as a question of correspondence. Truth cannot be revealed and left open, stored up and called upon when needed. Truth is the whole process of happening of creative research, of setting things in motion. If there is truth in science, it is here, in the intrinsic dynamics of skilled engagements.

10.3 A changed world

Our world is one of massive connectivity, ubiquitous globalised media. The processes and transfers that occur among these circuits, however, are not transparent and frictionless. As we incorporate computational processes into fields of practice, they serve not only as accelerators or amplifiers, but as materials for doing things, as tangible stuff with a specificity wholly its own. The digital computer is a very particular apparatus incorporated into wider systems of action in highly specific ways.

The contemporary world is also a world of large-scale threats, of climatological and ecological systems of vast complexity, each with an interdependence, a precariousness that is hard to fathom, let alone measure and protect. The recognition of the complexity of interlinked physical, geological, biological and social processes has emerged alongside the stamping of the human mark across the world through our information systems and networks, and it is these systems that form an essential element of the front line of research into systemic threats.

Between the great extension of calculative capacity represented by computational media and the large scale efforts to grasp the connections between processes, is a common interplay between the opaque and the transparent, between the tenuousness of the practical grasp of scientific problems and of the practical grasp of technical apparatuses. What we need to recognise is that the instability and precariousness of research itself, like the differential precariousness of ecological and climatological systems, is not a mark of insufficiency. It is a mark of a fundamental dynamic, a real, a capacity to be-other, to become-other, to extend towards unforeseen areas, draw new materials in, create radically new configurations, and transform itself and thus the world of which it is a part. If the greatest existential demand for humans today is to change their situation, we could do well to look to science, not for the “answers” but for inspiration about how to think, do, and make things otherwise.

505 Heidegger, ‘The Origin of the Work of Art’, 54.

Bibliography

Alac, Morana, and Edwin Hutchins. ‘I See What You Are Saying: Action as Cognition in fMRI Brain Mapping Practice’. *Journal of Cognition & Culture* 4, no. 3/4 (2004): 629–661.

Althusser, Louis. ‘Philosophy and the Spontaneous Philosophy of the Scientists’. In *Philosophy and the Spontaneous Philosophy of the Scientists & Other Essays*, edited by Gregory Elliott. London: Verso, 1990.

Appadurai, Arjun, ed. *The Social Life of Things: Commodities in Cultural Perspective*. Cambridge: Cambridge University Press, 1986.

Austin, John L. *How to Do Things with Words: The William James Lectures Delivered in Harvard University in 1955*. Edited by J. O. Urmson and Marina Sbisà. 2nd ed. Oxford: Oxford Paperbacks, 1976.

Bachelard, Gaston. *Le Rationalisme Appliqué*. Paris: Quadrige / Presses Universitaires de France, 1949.

———. ‘Noumena and Microphysics’. Translated by David Reggio. *Angelaki* 10, no. 2 (2005): 73–78.

———. *The Formation of the Scientific Mind*. Translated by Mary McAllester Jones. Manchester: Clinamen Press, 2002.

———. *The New Scientific Spirit*. Translated by Arthur Goldhammer. Boston: Beacon Press, 1984.

———. *The Philosophy of No: A Philosophy of the New Scientific Mind*. Translated by G. C. Waterston. New York: The Orion Press, 1968.

Badiou, Alain. *Being and Event*. Translated by Oliver Feltham. London: Continuum, 2007.

Bailer-Jones, Daniela M. ‘When Scientific Models Represent’. *International Studies in the Philosophy of Science* 17, no. 1 (2003): 59–74.

Balibar, E. ‘From Bachelard to Althusser: The Concept of “Epistemological Break”’. *Economy and Society* 7, no. 3 (1978): 207–237.

Barnes, Barry. *Scientific Knowledge and Sociological Theory*. Boston: Routledge, 1974.

Barnes, Bary, and Steven Shapin. *Natural Order: Historical Studies of Scientific Culture*. London: Sage, 1979.

Barry, Andrew. ‘Technological Zones’. *European Journal of Social Theory* 9, no. 2 (2006): 239–253.

Basili, Victor R., Daniela Cruzes, Jeffrey R. Carver, Lorin M. Hochstein, Jeffrey K. Hollingsworth, Marvin V. Zelkowitz, and Forrest Shull. ‘Understanding the High-Performance-Computing Community: An Software Engineer’s Perspective’. *IEEE Software* 25, no. 4 (2008): 29–36.

Batterman, Robert. ‘Idealization and Modeling’. *Synthese* 169, no. 3 (2009): 427–446. doi:10.1007/s11229-008-9436-1.

Baudrillard, J. *Simulations*. Cambridge: MIT Press, 1983.

Becker, Gary. ‘A Theory of Social Interactions’. *The Journal of Political Economy* 82, no. 6 (1974): 1063–1093.

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, et al. ‘Manifesto for Agile Software Development’, 2001. agilemanifesto.org.

Bennett, Jane. ‘The Force of Things: Steps Towards an Ecology of Matter’. *Political Theory* 32, no. 3 (2004): 347–372.

———. *Vibrant Matter: A Political Ecology of Things*. Durham: Duke University Press, 2010.

Bergson, Henri. *Creative Evolution*. Translated by Arthur Mitchell. New York: Random House, 1998.

Berry, David M. *The Philosophy of Software: Code and Mediation in the Digital Age*. Basingstoke: Palgrave Macmillan, 2011.

Bloch, Ernst. *The Principle of Hope*. Translated by Neville Plaice, Stephen Plaice, and Paul Knight. Cambridge: The MIT Press, 1986.

Bloor, David. *Knowledge and Social Imagery*. London: Routledge, 1976.

———. ‘Sichtbarmachung, Common Sense and Construction in Fluid Mechanics: The Cases of Hele-Shaw and Ludwig Prandtl’. *Studies in History and Philosophy of Science* 39 (2008): 349–358.

Bolduc, Jean-Sébastien, and Gérard Chazal. ‘The Bachelardian Tradition in the Philosophy of Science’. *Angelaki* 10, no. 2 (2005): 79–87.

Booch, Grady. ‘Measuring Architectural Complexity’. *IEEE Software* 25, no. 4 (2008): 14–15.

Borup, Mads, Nik Brown, Kornelia Kondrad, and Harro Van Lente. ‘The Sociology of Expectations in Science and Technology’. *Technology Analysis & Strategic Management* 18, no. 3/4 (2006): 285–298.

Bosteels, Bruno. ‘Nonplaces: An Anecdoted Topography of Contemporary French Theory’. *Diacritics* 33, no. 3/4 (2003): 117–139.

Bourdieu, Pierre. *Outline of a Theory of Practice*. Translated by Richard Nice. Cambridge: Cambridge University Press, 1977.

———. ‘The Peculiar History of Scientific Reason’. *Sociological Forum* 6, no. 1 (March 1991): 3–26.

Bowker, Geoffrey C. *Memory Practices in the Sciences*. Cambridge: MIT Press, 2005.

Bowker, Geoffrey C., and Bruno Latour. ‘A Booming Discipline Short of Discipline: (Social) Studies of Science in France’. *Social Studies of Science* 17, no. 4 (1987): 715 –748. doi:10.1177/030631287017004006.

Brooks, Jr., Frederick P. ‘No Silver Bullet: Essence and Accident in Software Engineering’. In *The Mythical Man-Month: Essays on Software Engineering*, 177–203. Anniversary Edition. Boston: Addison-Wesley, 1995.

———. ‘The Tar Pit’. In *The Mythical Man-Month: Essays on Software Engineering*, 3–12. Anniversary Edition. Boston: Addison-Wesley, 1995.

Brown, Bill. ‘Thing Theory’. *Critical Inquiry* 28, no. 1 (2001): 1–22.

Brown, Matthew J. ‘John Dewey’s Logic of Science’. *HOPOS: The Journal of the International Society for the History of Philosophy of Science Pp. 258-306* 2, no. 2 (2012): 258–306.

Brown, Nik, and Mike Michael. ‘A Sociology of Expectations: Retrospective Prospects and Prospecting Retrospects’. *Technology Analysis & Strategic Management* 15 (2003): 3–18.

Bueno de Mesquita, Bruce, and David Lalman. *War and Reason: Domestic and International Imperatives*. New Haven: Yale University Press, 1992.

Butler, Judith. *Excitable Speech: A Politics of the Performative*. London: Routledge, 1997.

Callon, Michel. ‘Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St Brieuc Bay’. In *Power, Action and Belief: A New Sociology of Knowledge?*, edited by John Law, 196–233. London: Routledge, 1986.

Callon, Michel, and Bruno Latour. ‘Don’t Throw the Baby Out with the Bath School! A Reply to Collins and Yearley’. In *Science as Practice and Culture*, edited by Andrew Pickering, 343–368. Chicago: University of Chicago Press, 1992.

Carnap, Rudolf. ‘Empiricism, Semantics, and Ontology’. *Revue Internationale De Philosophie* 4 (1950): 20–40.

Cartwright, Nancy. *How the Laws of Physics Lie*. Oxford: Oxford University Press, 1983.

Carusi, Annamaria. ‘Computational Biology and the Limits of Shared Vision’. *Perspectives on Science* 19, no. 3 (2011): 300–336.

Castelão-Lawless, Teresa. 'Phenomenotechnique in Historical Persepective: Its Origins and Implications for Philosophy of Science'. *Philosophy of Science* 62 (1995): 44–59.

De Certeau, Michel. *The Practice of Everyday Life*. Translated by Steven Rendall. Berkeley: University of California Press, 2002.

Chadarevian, Soraya de, and Nick Hopwood. *Models: The Third Dimension of Science*. Stanford: Stanford University Press, 2004.

Clark, Andy, and David Chalmers. 'The Extended Mind'. *Analysis* 58, no. 1 (1998): 7–19.

Clifford, James, and George E. Marcus, eds. *Writing Culture: The Poetics and Politics of Ethnography*. Berkeley: University of California Press, 1986.

Collins, Harry, and Steven Yearley. 'Epistemological Chicken'. In *Science as Practice and Culture*, edited by Andrew Pickering, 301–326. Chicago: University of Chicago Press, 1992.

Da Costa, Newton C. A., and Steven French. *Science and Partial Truth: A Unitary Approach to Models and Scientific Reasoning*. Oxford: Oxford University Press, 2003.

Cubitt, Sean. *Simulation and Social Theory*. London: Sage, 2001.

Daston, Lorraine. 'Historical Epistemology'. In *Questions of Evidence: Proof, Practice, and Persuasion Across the Disciplines*, edited by James Chandler, Arnold Ira Davidson, and Harry D. Harootunian, 282–289. Chicago: University of Chicago Press, 1994.

Daston, Lorraine, and Peter Galison. *Objectivity*. New York: Zone Books, 2007.

Davidson, Arnold I. 'Styles of Reasoning, Conceptual History, and the Emergence of Psychiatry'. In *The Disunity of Science: Boundaries, Contexts, and Power*, edited by Peter Galison and David J Stump, 75–100. Stanford: Stanford University Press, 1996.

Deleuze, Gilles. *Difference and Repetition*. Translated by Paul Patton. London: Continuum, 2004.

———. *The Fold: Leibniz and the Baroque*. Translated by Tom Conley. Minneapolis: University of Minnesota Press, 1992.

Deleuze, Gilles, and Felix Guattari. *Anti-Oedipus*. London: Continuum, 2004.

Denning, Peter J. 'Computing Is a Natural Science'. *Proceedings of the Association for Computing Machinery* 50, no. 7 (2007): 13–18.

Derrida, Jacques. *Dissemination*. Translated by Barbara Johnson. London: Continuum, 2004.

———. *Of Grammatology*. Translated by Gayatri Chakravorty Spivak. Baltimore: John Hopkins University Press, 1974.

Dexter, Scott, Melissa Dolese, Angelika Seidel, and Aaron Kozbelt. 'On the Embodied Aesthetics of Code'. *Culture Machine* 12 (2011): 1–23.

Dosse, Francois. *History of Structuralism: Vol. 1: The Rising Sign: 1945-1966*. Translated by Deborah Glassman. Minneapolis: University of Minnesota Press, 1997.

Dowling, Deborah. 'Experimenting on Theories'. *Science in Context* 12, no. 2 (1999): 261–273.

Dunbar, Kevin. 'How Scientists Think: On-Line Creativity and Conceptual Change in Science'. In *Creative Thought: An Investigation of Conceptual Structures and Processes*, edited by Thomas B. Ward, Steven M. Smith, and Jyotsna Vaid, 461–493. Washington: American Psychological Association, 1997.

Durkheim, Emile. *The Elementary Forms of the Religious Life*. London: George Allen and Unwin, 1971.

Easterbrook, Steve M., and Timothy C. Johns. 'Engineering the Software for Understanding Climate Change'. *Computing in Science Engineering* 11, no. 6 (2009): 65 –74. doi:10.1109/MCSE.2009.193.

Edwards, Paul N. *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. Cambridge: The MIT Press, 2010.

Ehrenberg, A. S. C. 'Models of Fact: Examples from Marketing'. *Management Science* 16, no. 7 (1970): 435–445.

Engineering and Physical Sciences Research Council. 'International Review of Research Using HPC in the UK'. EPSRC, December 2005.

Evans-Pritchard, Edward Evan. *Witchcraft, Oracles and Magic Among the Azande*. Oxford: Oxford University Press, 1976.

Evens, Aden. 'Object-Oriented Ontology, Or Programming's Creative Fold'. *Angelaki* 2, no. 1 (2006): 89–97.

Foucault, Michel. *Discipline and Punish: The Birth of the Prison*. Edited by Alan Sheridan. London: Penguin, 1977.

———. 'Introduction'. In *The Normal and the Pathological*, by Georges Canguilhem, translated by Robert Cohen. New York: Zone Books, 1989.

———. *Power/Knowledge: Selected Interviews & Other Writings 1972-1977*. Edited by Colin Gordon. New York: Pantheon Books, 1980.

———. *The Order of Things: An Archaeology of the Human Sciences*. 2nd ed. London: Routledge, 2002.

Van Fraassen, Bas. *Scientific Representation: Paradoxes of Perspective*. Oxford: Clarendon Press, 2008.

———. *The Scientific Image*. Oxford: Oxford University Press, 1980.

Franklin, Allan. *Experiment Right or Wrong*. Cambridge: Cambridge University Press, 1990.

———. *The Neglect of Experiment*. Cambridge: Cambridge University Press, 1986.

Franklin, James. 'The Formal Sciences Discover the Philosophers' Stone'. *Studies in History and Philosophy of Science* 25, no. 4 (1994): 513–533.

Fraser, Zachary Luke. 'The Category of Formalization: From Epistemological Break to Truth Procedure'. In *The Concept of Model: An Introduction to the Materialist Epistemology of Mathematics*, by Alain Badiou, xi–lxv. edited by Zachary Luke Frazer and Tzuchien Tho, xi–lxv. Re. Press, 2007.

French, Steven. 'Keeping Quiet on the Ontology of Models'. *Synthese* 172 (2010): 231–249.

French, Steven, and James Ladyman. 'Reinflating the Semantic Approach'. *International Studies in the Philosophy of Science* 13, no. 2 (1999): 103–121. doi:10.1080/02698599908573612.

Frigg, Roman. 'Models and Fiction'. *Synthese* 172 (2010): 1–18.

Frigg, Roman, and Julian Reiss. 'The Philosophy of Simulation: Hot New Issues or Same Old Stew?' *Synthese* 169 (2009): 593–613.

Fry, Jenny. 'Coordination and Control of Research Practice Across Scientific Fields: Implications for a Differentiated E-Science'. In *New Infrastructures for Knowledge Production: Understanding E-Science*, edited by Christine M. Hine, 167–187. London: Information Science Publishing, 2006.

Fuller, Matthew. 'Introduction, the Stuff of Software'. In *Software Studies: A Lexicon*, 1–13. Cambridge: MIT Press, 2008.

———. *Software Studies: A Lexicon*. Cambridge: MIT Press, 2008.

Gabriel, Richard P. *Patterns of Software: Tales from the Software Community*. Oxford: Oxford University Press, 1996.

Galison, Peter. 'Computer Simulations and the Trading Zone'. In *The Disunity of Science: Boundaries, Contexts, and Power*, edited by Peter Galison and David Stump, 118–157. Stanford: Stanford University Press, 1996.

———. 'Context and Constraints'. In *Scientific Practice: Theories and Stories of Doing Physics*, edited by Jed Buchwald, 13–41. Chicago: Chicago University Press, 1995.

———. *How Experiments End*. Chicago: University of Chicago Press, 1987.

———. *Image and Logic: Material Culture of Microphysics*. Chicago: University of Chicago Press, 1997.

Gelfert, Axel. 'Model-Based Representation in Scientific Practice: New Perspectives'. *Studies in History and Philosophy of Science* 42 (2011): 251–252.

Gell, Alfred. *Art and Agency: An Anthropological Theory*. Oxford: Oxford University Press, 1998.

———. 'The Technology of Enchantment and the Enchantment of Technology'. In *Anthropology, Art and Aesthetics*, edited by Jeremy Coote and Anthony Shelton, 40–66. Oxford: Clarendon Press, 1992.

Gibbons, Michael, Camille Limoges, Helga Nowotny, Simon Schwartzman, Peter Scott, and Martin Trow. *The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies*. Stockholm: Sage Publications Ltd, 1994.

Gibson, James J. *The Ecological Approach To Visual Perception*. Hillsdale: Lawrence Erlbaum Associates, 1986.

Giere, Ronald N. *Explaining Science: A Cognitive Approach*. Chicago: University of Chicago Press, 1988.

———. 'How Models Are Used to Represent Reality'. *Philosophy of Science* 71, no. 5 (2004): 742–752.

———. *Scientific Perspectivism*. Chicago: University of Chicago Press, 2006.

Gingras, Yves. 'Following Scientists Though Society? Yes, but at Arm's Length!' In *Scientific Practice: Theories and Stories of Doing Physics*, edited by Jed Buchwald, 123–148. University of Chicago Press, 1995.

———. 'From the Heights of Metaphysics: A Reply to Pickering'. *Social Studies of Science* 29, no. 2 (1999): 312–315.

———. 'The New Dialectics of Nature'. *Social Studies of Science* 27 (1997): 317–334.

Gingras, Yves, and Benoit Godin. 'The Experimenters' Regress: From Skepticism to Argumentation'. *Studies in History and Philosophy of Science* 33 (2002): 137–152.

Godfrey-Smith, Peter. 'The Strategy of Model-Based Science'. *Biology and Philosophy* 21 (2006): 725–740.

Goldsman, David, Richard E. Nance, and James R. Wilson. 'A Brief History of Simulation'. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rosetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 310–313, 2009.

Goody, Jack. *The Logic of Writing and the Organisation of Society*. Cambridge: Cambridge University Press, 1986.

Graeber, David. 'Beyond Power/Knowledge: An Exploration of the Relation of Power, Ignorance and Stupidity'. presented at the Malinowski Memorial Lecture, London School of Economics, May 25, 2006.

Gramelsberger, Gabriele. 'What Do Numerical (Climate) Models Really Represent?' *Studies in History and Philosophy of Science* 42 (2011): 296–302.

Grieffenhagen, Christian, and Wes Sharrock. 'Logical Relativism: Logic, Grammar, and Arithmetic in Cultural Comparison'. *Configurations* 14 (2006): 275–301.

Gross, Matthias. *Ignorance and Surprise: Science, Society, and Ecological Design*. Cambridge: MIT Press, 2010.

Guala, Francesco. 'Models, Simulations, and Experiments'. In *Model-based Reasoning: Science, Technology, Values*, edited by Lorenzo Magnani. New York: Kluwer Academic, 2002.

Gutting, Gary. *Michel Foucault's Archaeology of Scientific Reason*. Cambridge: Cambridge University Press, 1989.

Hacking, Ian. 'Language, Truth and Reason'. In *Rationality and Relativism*, edited by Martin Hollis and Steven Lukes, 48–66. Oxford: Blackwell, 1982.

———. 'Making Up People'. *London Review of Books* 28, no. 6 (August 17, 2006): 23–26.

———. *Representing and Intervening: Introductory Topics in the Philosophy of Natural Science*. Cambridge: Cambridge University Press, 1983.

———. *The Social Construction of What?* Cambridge: Harvard University Press, 1999.

Haraway, Donna. *Modest_Witness@Second_Millennium.FemaleMan_Meets_OncoMouse: Feminism and Technoscience*. London: Routledge, 1997.

Harman, Graham. 'The Importance of Bruno Latour'. *Cultural Studies Review* 13, no. 1 (2007): 31–49.

Hartmann, Stephan. 'The World as Process: Simulations in the Natural and Social Sciences'. In *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, edited by R. Hegselmann, Ulrich Mueller, and Klaus G. Troitzsch, 77–100. Dordrecht: Kluwer Academic Press, 1996.

Hayles, N. Katherine, ed. *Nanoculture: Implications of the New Technoscience*. Portland: Intellect Books, 2004.

Heidegger, Martin. *Being and Time*. Translated by John Macquarrie and Edward Robinson. Oxford: Blackwell, 1978.

———. *Poetry, Language, Thought*. Translated by Albert Hofstadter. New York: Perennial Classics, 2001.

———. 'The Age of the World Picture'. In *The Question Concerning Technology, and Other Essays*, translated by William Lovitt, 115–154. New York: Harper & Row, 1977.

———. 'The Origin of the Work of Art'. In *Poetry, Language, Thought*, translated by Albert Hofstadter, 15–86. New York: Perennial Classics, 2001.

———. 'The Question Concerning Technology'. In *The Question Concerning Technology, and Other Essays*, translated by William Lovitt, 3–35. New York: Harper & Row, 1977.

Heintz, Christophe. 'Introduction: Why There Should Be a Cognitive Anthropology of Science'. *Journal of Cognition & Culture* 4, no. 3/4 (2004): 391–408. doi:10.1163/1568537042484922.

Henderson, Kathryn. 'The Political Career of a Prototype: Visual Representation in Design Engineering'. *Social Problems* 42, no. 2 (1995): 274–299.

Hesse, Mary B. *Models and Analogies in Science*. Notre Dame: University of Notre Dame Press, 1966.

Heymann, Matthias. 'Modeling Reality: Practice, Knowledge, and Uncertainty in Atmospheric Transport Simulation'. *Historical Studies in the Physical and Biological Sciences* 37, no. 1 (2006): 49–85.

Hine, Christine M. 'Computerization Movements and Scientific Disciplines: The Reflexive Potential of New Technologies'. In *New Infrastructures for Knowledge Production: Understanding E-Science*, edited by Christine M. Hine, 26–47. London: Information Science Publishing, 2006.

Hollis, Martin, and Steven Lukes, eds. *Rationality and Relativism*. Oxford: Blackwell, 1982.

Humphreys, Paul. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford: Oxford University Press, 2004.

———. 'The Philosophical Novelty of Computer Simulation Methods'. *Synthese* 169 (2009): 615–626.

Humphrey, W.S. 'Why Big Software Projects Fail: The 12 Key Questions'. *STSC Crosstalk, March* (2005).

Husserl, Edmund. *The Crisis of European Sciences and Transcendental Phenomenology: An Introduction to Phenomenological Philosophy*. Translated by Daviv Carr. Evanston: Northwestern University Press, 1970.

Hutchins, Edwin. *Cognition in the Wild*. Cambridge: MIT Press, 1995.

Hyder, David. 'Foucault, Cavaillès, and Husserl on the Historical Epistemology of the Sciences'. *Perspectives on Science* 11, no. 1 (2003): 107–129.

Ihde, Don. *Expanding Hermeneutics: Visualizing Science*. Evanston: Northwestern University Press, 1999.

Ingold, Tim. *Being Alive: Essays on Movement, Knowledge and Description*. Oxford: Routledge, 2011.

———. 'Beyond Art and Technology: The Anthropology of Skill'. In *Anthropological Perspectives on Technology*, edited by Michael Schiffer, 17–31. Albuquerque: University of New Mexico Press, 2001.

———. 'Bringing Things to Life: Creative Entanglements in a World of Materials'. Vol. Working Paper #15. Manchester: ESRC National Centre for Research Methods, 2010.

Ingold, Tim, ed. *Key Debates In Anthropology*. London: Routledge, 1996.

Jones, Mary McAllester. 'Introduction'. In *The Formation of the Scientific Mind*, by Gaston Bachelard, 1–16. translated by Mary McAllester Jones, 1–16. Manchester: Clinamen Press, 2002.

Justi, Rosaria, and John Gilbert. 'Models and Modelling in Chemical Education'. In *Chemical Education: Towards Research-based Practice*, edited by John Gilbert, Onno Jong, Rosária Justi, David Treagust, and Jan Driel, 17:47–68. Dordrecht: Kluwer Academic Press, 2003.

Kant, Immanuel. *Critique of Pure Reason*. Translated by Marcus Weigelt and Max Müller. London: Penguin, 2007.

Kaufmann III, William J., and Larry L. Smarr. *Supercomputing and the Transformation of Science*. New York: Scientific American Library, 1993.

Keller, Evelyn Fox. 'Models, Simulation, and "Computer Experiments"'. In *The Philosophy of Scientific Experimentation*, edited by Hans Radder, 198–215. Pittsburgh: University of Pittsburgh Press, 2003.

———. 'What Are Climate Scientists to Do?' *Spontaneous Generations: A Journal for the History and Philosophy of Science* 5, no. 1 (2011): 19–26.

Kelty, Christopher. 'Free Science'. In *Perspectives on Free and Open Source Software*, edited by Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani, 415–430. Cambridge: MIT Press, 2005.

Kleiner, Kurt. 'Data on Demand'. *Nature Climate Change* 1 (2011): 10–12.

Klein, Ursula. *Experiments, Models, Paper Tools: Cultures of Organic Chemistry in the Nineteenth Century*. Stanford: Stanford University Press, 2002.

Knorr-Cetina, Karin. *Epistemic Cultures: How the Sciences Make Knowledge*. Harvard Univ Pr, 1999.

———. 'Objectual Practice'. In *The Practice Turn in Contemporary Theory*, 184–197. London: Routledge, 2001.

———. *The Manufacture of Knowledge: An Essay on the Constructivist and Contextual Nature of Science*. Oxford: Pergamon Press, 1981.

Knorr, Karin. 'Tinkering Toward Success'. *Theory and Society* 8 (1979): 347–376.

Knuuttila, Tarja. 'Modelling and Representing: An Artefactual Approach to Model-based Representation'. *Studies in History and Philosophy of Science* 42 (2011): 262–271.

———. 'Models as Epistemic Artefacts: Toward a Non-representationalist Account of Scientific Representation'. *Philosophical Studies of the University of Helsinki* 8 (2005).

———. 'Models, Representation, and Mediation'. *Philosophy of Science* 72, no. 5 (2005): 1260–1271.

Krieger, Martin H. *Doing Physics: How Physicists Take Hold of the World*. Bloomington: Indiana University Press, 1992.

Kruger, Erin. 'Visualizing Uncertainty: Anomalous Images in Science and Law'. *Interdisciplinary Science Reviews* 37, no. 1 (2012): 19–35.

Kuhn, T. S. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press, 1970.

Lacan, Jacques. 'The Instance of the Letter in the Unconscious'. In *Écrits*, translated by Bruce Fink, Héloïse Fink, and Russell Grigg, 412–441. Complete edition. New York: W. W. Norton & Company, 2006.

Lahsen, Myanna. 'Seductive Simulations? Uncertainty Distribution Around Climate Models'. *Social Studies of Science* 36, no. 6 (2005): 895–922.

De Landa, Manuel. *Philosophy and Simulation: The Emergence of Synthetic Reason*. London: Continuum, 2011.

———. 'Virtual Environments and the Concept of Synergy'. *Leonardo* 28, no. 5 (1995): 357–360.

Lash, Scott, and Celia Lury. *Global Culture Industry: The Mediation of Things*. Cambridge: Polity Press, 2007.

Latour, B. 'Why Has Critique Run Out of Steam? From Matters of Fact to Matters of Concern'. *Critical Inquiry* 30, no. 2 (2004): 225–248.

Latour, Bruno. *Aramis or the Love of Technology*. Translated by Catherine Porter. Cambridge: Harvard University Press, 1996.

———. 'Force and the Reason of Experiment'. In *Experimental Inquiries: Historical, Philosophical and Social Studies of Experimentation in Science*, edited by Homer Le Grande, 49–80. Dordrecht: Kluwer Academic Press, 1990.

———. 'Give Me a Laboratory and I Will Raise the World'. In *Science Observed: Perspectives on the Social Study of Science*, 141–170. Minneapolis: University of Minnesota Press, 1983.

———. 'How to Talk About the Body? The Normative Dimension of Science Studies'. *Body and Society* 10, no. 2/3 (2004): 205–229.

———. *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford: Oxford University Press, 2005.

———. *Science in Action: How to Follow Scientists and Engineers Through Society*. Cambridge: Harvard University Press, 1988.

———. *We Have Never Been Modern*. Cambridge: Harvard University Press, 1993.

Latour, Bruno, and Peter Weibel. *Iconoclash: Beyond the Image Wars in Science, Religion and Art*. Cambridge: The MIT Press, 2002.

Latour, Bruno, and Steve Woolgar. *Laboratory Life: The Construction of Scientific Facts*. Princeton: Princeton University Press, 1986.

Law, John. *Aircraft Stories: Decentering the Object in Technoscience*. Durham: Duke University Press, 2002.

Lecourt, Dominique. *Marxism and Epistemology: Bachelard, Canguilhem and Foucault*. Translated by Ben Brewster. London: New Left Review Editions, 1975.

Leibniz, Gottfried Wilhelm. *New Essays Concerning Human Understanding*. Translated by Alfred Gideon Langley. New York: Macmillan, 1896.

Lenhard, Johannes. ‘Computer Simulation: The Cooperation Between Experimenting and Modeling’. *Philosophy of Science* 74, no. 2 (2007): 176–194.

Lenoir, Tim. ‘Epistemology Historicized: Making Epistemic Things’. In *An Epistemology of the Concrete : Twentieth-century Histories of Life*, by Hans-Jörg Rheinberger. Durham: Duke University Press, 2010.

Lévi-Strauss, Claude. *The Elementary Structures of Kinship*. Edited by Rodney Needham. Translated by James Harle Bell and John Richard von Sturner. Boston: Beacon Press, 1977.

———. *The Savage Mind*. Translated by John Weightman and Doreen Weightman. Chicago: University of Chicago Press, 1966.

Lynch, Michael. ‘Allan Franklin’s Transcendental Physics’. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 2 (1990): 471–485.

———. ‘Science in the Age of Mechanical Reproduction: Moral and Epistemic Relations Between Diagrams and Photographs’. *Biology and Philosophy* 6 (1991): 205–226.

Lynch, Michael, Eric Livingston, and Harold Garfinkel. ‘Temporal Order in Laboratory Work’. In *Science Observed: Perspectives on the Social Study of Science*, 205–238. Minneapolis: University of Minnesota Press, 1983.

Lynch, Michael, and Steve Woolgar, eds. *Representation in Scientific Practice*. Cambridge: The MIT Press, 1990.

MacKenzie, Adrian. *Cutting Code: Software And Sociality*. New Tork: Peter Lang, 2006.

MacKenzie, Donald. *Mechanizing Proof: Computing, Risk, and Trust*. Cambridge: MIT Press, 2001.

Magnani, Lorenzo, and Nancy J. Nersessian, eds. *Model-Based Reasoning: Science, Technology, Values*. New York: Kluwer Academic, 2002.

Manovich, Lev. *The Language of New Media*. New Ed. Cambridge: MIT Press, 2002.

Marick, Brian. ‘A Manglish Way of Working: Agile Software Development’. In *The Mangle in Practice: Science, Society, and Becoming*, edited by Andrew Pickering and Keith Guzik, 185–201. Durham: Duke University Press, 2008.

Marres, Noortje. *Material Participation: Technology, the Environment and Everyday Publics*. Basingstoke: Palgrave Macmillan, 2012.

Marx, Karl. *Capital: A Critical Analysis of Capitalist Production*. Translated by Samuel Moore and Edward Aveling. Vol. 1. 3 vols. Translated from the third German edition. London: William Glaisher, 1886.

Matsumoto, Yukihiro. ‘Treating Code as an Essay’. In *Beautiful Code: Leading Programmers Explain How They Think*, edited by Andy Oram and Greg Wilson, 477–481. Cambridge: O’Reilly, 2007.

Mattingly, James, and Walter Warwick. ‘Projectible Predicates in Analogue and Simulated Systems’. *Synthese* 169, no. 3 (2009): 465–482. doi:10.1007/s11229-008-9433-4.

Maurer, Bill. ‘Does Money Matter? Abstraction and Substitution in Alternative Financial Forms’. In *Materiality*, edited by Daniel Miller, 140–164. Durham: Duke University Press, 2006.

Mauss, Marcel. ‘Les Techniques Du Corps’. *Journal De Psychologie* XXXII, no. 3–4 (1934).

McGuffie, Kendal, and Ann Henderson-Sellers. *A Climate Modelling Primer*. Third Edition. Chichester: Wiley, 2005.

Meillassoux, Quentin. *After Finitude: An Essay on the Necessity of Contingency*. Translated by Ray Brassier. London: Continuum, 2009.

Merleau-Ponty, Maurice. *Phenomenology of Perception*. Translated by Colin Smith. London: Routledge, 2002.

Merz, Martina. ‘Embedding Digital Infrastructure in Epistemic Culture’. In *New Infrastructures for Knowledge Production: Understanding E-Science*, edited by Christine M. Hine, 99–119. London: Information Science Publishing, 2006.

———. ‘Multiplex and Unfolding: Computer Simulation in Particle Physics’. *Science in Context* 12, no. 02 (1999): 293–316. doi:10.1017/S0269889700003434.

Michael, Mike, Steven P. Wainwright, and Clare Williams. ‘Temporality and Prudence: On Stem Cells as “Phronesic Things”’. *Configurations* 13 (2007): 373–394.

Miller, Daniel, ed. *Materiality*. Durham: Duke University Press, 2006.

Monteiro, Marko. ‘Reconfiguring Evidence: Interacting with Digital Objects in Scientific Practice’. *Computer Supported Cooperative Work* 19, no. 3–4 (2010): 335–354.

Morgan, Mary, and Margaret Morrison. *Models as Mediators: Perspectives on Natural and Social Sciences*. Cambridge: Cambridge University Press, 1999.

Morgan, Mary S. ‘Experiments Without Material Intervention: Model Experiments, Virtual Experiments, and Virtually Experiments’. In *The Philosophy of Scientific Experimentation*, by Hans Radder, 216–235, 216–235. Pittsburgh: University of Pittsburgh Press, 2003.

———. ‘Models as Working Objects in Science’. presented at the Models & Simulations 5, Helsinki, 14 2012.

———. *The World in the Model: How Economists Work and Think*. Cambridge: Cambridge University Press, 2012.

Morrison, Margaret. ‘Models as Autonomous Agents’. In *Models as Mediators: Perspectives on Natural and Social Science*, edited by Mary S. Morgan and Margaret Morrison, 38–65. Cambridge: Cambridge University Press, 1999.

Myers, Natasha. ‘Animating Mechanism: Animations and the Propagation of Affect in the Lively Arts of Protein Modelling’. *Science Studies* 19, no. 2 (2006): 5–30.

Nelson, Alan. ‘How Could Scientific Facts Be Socially Constructed?’ *Studies in History and Philosophy of Science* 25, no. 4 (1994): 535–547.

Nersessian, Nancy J. ‘How Do Engineering Scientists Think? Model-Based Simulation in Biomedical Engineering Research Laboratories’. *Topics in Cognitive Science* 1 (2009): 730–757.

Netz, Reviel. *The Shaping of Deduction in Greek Mathematics: A Study in Cognitive History*. Cambridge: Cambridge University Press, 2003.

Oreskes, Naomi, Kristin Shrader-Frechette, and Kenneth Belitz. ‘Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences’. *Science* 263, no. 5247 (1994): 641–646.

Ortner, Sherry B. ‘Theory in Anthropology Since the Sixties’. *Comparative Studies in Society and History* 26, no. 1 (1984): 126–166.

Ostrom, Thomas M. ‘Computer Simulation: The Third Symbol System’. *Journal of Experimental Social Psychology* 24 (1988): 381–392.

Parker, Matthew. ‘Computing the Uncomputable; or, The Discrete Charm of Second-order Simulacra’. *Synthese* 169, no. 3 (2009): 447–463. doi:10.1007/s11229-008-9441-4.

Parker, Wendy. ‘Does Matter Really Matter? Computer Simulations, Experiments, and Materiality’. *Synthese* 169, no. 3 (2009): 483–496. doi:10.1007/s11229-008-9434-3.

Pauwels, Luc, ed. *Visual Cultures of Science: Rethinking Representational Practices in Knowledge Building and Science Communication*. Hanover: Dartmouth College Press, 2006.

Pels, Dick, Kevin Hetherington, and Frédéric Vandenberghe. 'The Status of the Object: Performances, Mediations, and Techniques'. *Theory, Culture & Society* 19, no. 5/6 (2002): 1–21.

Petersen, Arthur C. *Simulating Nature: A Philosophical Study of Computer-simulation Uncertainties and Their Role in Climate Science and Policy Advice*. Second edition. Boca Raton: CRC Press, 2012.

Pickering, Andrew. 'Beyond Constraint: The Temporality of Practice and the Historicity of Knowledge'. In *Scientific Practice: Theories and Stories of Doing Physics*, edited by Jed Buchwald, 42–55. Chicago: Chicago University Press, 1995.

———. *Constructing Quarks: A sociological history of particle physics*. Edinburgh: Edinburgh University Press, 1984.

———. 'From Science as Knowledge to Science as Practice'. In *Science as Practice and Culture*, edited by Andrew Pickering, 1–26. Chicago: University of Chicago Press, 1992.

———. 'In the Land of the Blind... Thoughts on Gingras'. *Social Studies of Science* 29, no. 2 (1999): 307–311.

———. 'New Ontologies'. In *The Mangle in Practice: Science, Society, and Becoming*, edited by Andrew Pickering and Keith Guzik, 1–14. Durham: Duke University Press, 2008.

———. *The Cybernetic Brain: Sketches of another future*. Chicago: University of Chicago Press, 2010.

———. *The Mangle of Practice: Time, Agency, and Science*. Chicago: University of Chicago Press, 1995.

———. 'The Politics of Theory: Producing Another World, with Some Thoughts on Latour'. *Journal of Cultural Economy* 2, no. 1 (2009): 197.

Pitkin. *The Concept of Representation*. New Ed. University of California Press, 1992.

Polanyi, Michael. *Science, Faith and Society: A Searching Examination of the Meaning and Nature of Scientific Inquiry*. Chicago: Chicago University Press, 1964.

———. *The Tacit Dimension*. Gloucester: Peter Smith, 1983.

Preda, Alex. 'The Turn to Things: Arguments for a Sociological Theory of Things'. *Sociological Quarterly* 40, no. 2 (1999): 347–366.

Prigogine, Ilya, and Isabelle Stengers. *Order Out of Chaos: Man's new dialogue with nature*. London: Flamingo, 1985.

Quine, Willard Van Orman. 'Main Trends in Recent Philosophy: Two Dogmas of Empiricism'. *The Philosophical Review* 60 (1951): 20–43.

———. *Word and Object*. Cambridge: The MIT Press, 1964.

Rabinow, Paul. *Anthropos Today: Reflections on Modern Equipment*. Princeton: Princeton University Press, 2003.

Radder, Hans. 'Toward a More Developed Philosophy of Scientific Experimentation'. In *The Philosophy of Scientific Experimentation*, edited by Hans Radder, 1–18. Pittsburgh: University of Pittsburgh Press, 2003.

Ranciere, Jacques. 'Democracy, Republic, Representation'. *Constellations* 13, no. 3 (2006): 297–307.

Randall, David A., and Bruce A. Wielicki. 'Measurements, Models, and Hypotheses in the Atmospheric Sciences'. *Bulletin of the American Meteorological Society* 78, no. 3 (March 1997): 399.

Rheinberger, Hans-Jörg. 'Experimental Systems: Historiality, Narration and Deconstruction'. *Science in Context* 7, no. 1 (1994): 65–81.

Rheinberger, Hans-Jörg. *An Epistemology of the Concrete : Twentieth-century Histories of Life*. Durham: Duke University Press, 2010.

———. 'A Reply to David Bloor: "Toward a Sociology of Epistemic Things"'. *Perspectives on Science* 13, no. 3 (n.d.): 406–410.

———. 'Experimental Complexity in Biology: Some Epistemological and Historical Remarks'. *Philosophy of Science* 64, Proceedings of the 1996 Biennial Meetings of the Philosophy of Science Association. Part II: Symposia Papers (1997): S245–S254.

———. ‘From Microsomes to Ribosomes: “Strategies” of “Representation”’. *Journal of the History of Biology* 28, no. 1 (1995): 49–89.

———. *On Historicizing Epistemology: An Essay*. Translated by David Fernbach. Stanford: Stanford University Press, 2010.

———. *Toward a History of Epistemic Things: Synthesizing Proteins in the Test Tube*. Stanford: Stanford University Press, 1997.

Riles, Annelise. *Documents: Artifacts of Modern Knowledge*. Michigan: University of Michigan Press, 2006.

Rohrlitch, Fritz. ‘Computer Simulation in the Physical Sciences’. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 1990, no. 2 (1990): 507–518.

Rorty, Richard. *Philosophy and the Mirror of Nature*. Princeton: Princeton University Press, 1981.

Rosenberger, Robert. ‘Quick-Freezing Philosophy: An Analysis of Imaging Technologies in Neurobiology’. In *New Waves in Philosophy of Technology*, edited by Jan-Kyrre Olsen, Evan Selinger, and Søren Riis, 65–82. Basingstoke: Palgrave Macmillan, 2009.

Rosenberg, Scott. *Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software*. New York: Three Rivers Press, 2008.

Rouse, Joseph. *Engaging Science: How to Understand Its Practices Philosophically*. Ithaca: Cornell University Press, 1996.

———. ‘Understanding Scientific Practices: Cultural Studies of Science as a Philosophical Program’. In *The Science Studies Reader*, edited by Mario Biagioli, 442–456. London: Routledge, 1999.

Salanskis, Jean-Michel. ‘Phenomenology and Epistemology: War and Marriage’. Oxford: The British Society for Phenomenology, 2010.

Schatzki, Theodore. ‘Introduction: Practice Theory’. In *The Practice Turn in Contemporary Theory*, 10–23. London: Routledge, 2001.

———. ‘Materiality and Social Life’. *Nature and Culture* 5, no. 2 (2010): 123–149.

———. *Social Practices: A Wittgenstein Approach to Human Activity and the Social*. Cambridge: Cambridge University Press, 1996.

Schroeder, Ralph. ‘e-Sciences as Research Technologies: Reconfiguring Disciplines, Globalizing Knowledge’. *Social Science Information* 47, no. 2 (2008): 131–157.

Segal, Judith, and Chris Morris. ‘Developing Scientific Software’. *IEEE Software* 25, no. 4 (2008): 18–20.

Shapin, Steven, and Simon Schaffer. *Leviathan and the Air-Pump: Hobbes, Boyle, and the Experimental Life*. Princeton: Princeton University Press, 1985.

Sigaut, François. ‘Technology’. In *Companion Encyclopedia of Anthropology*, edited by Tim Ingold, 420–459. New York; London: Routledge, 2002.

Simondon, Gilbert. ‘Technical Mentality’. Translated by Arne De Boever. *Parrhesia* 7 (2009): 17–27.

———. ‘The Position of the Problem of Ontogenesis’. Translated by Gregor Flanders. *Parrhesia* 7 (2009): 4–16.

Simonton, Dean Keith. *Creativity in Science: Chance, Logic, Genius, and Zeitgeist*. Cambridge: Cambridge University Press, 2004.

Sismondo, Sergio. ‘Models, Simulations, and Their Objects’. *Science in Context* 12, no. 2 (1999): 247–260. doi:10.1017/S0269889700003409.

Spencer, Matt. ‘Image and Practice: Visualisation in Computational Fluid Dynamics Research’. *Interdisciplinary Science Reviews* 37, no. 1 (2012): 86–100.

———. ‘Trouble with Images in Computational Physics’. *Spontaneous Generations: A Journal for the History and Philosophy of Science* 6, no. 1 (2012): 34–42.

Star, Susan Leigh, and James R. Griesemer. ‘Institutional Ecology, “Translations” and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, 1907-39’. *Social Studies of Science* 19 (1989): 387–420.

Stengers, Isabelle. *Cosmopolitics I*. Translated by Robert Bononno. Minneapolis: University of Minnesota Press, 2003.

———. *The Invention of Modern Science*. Minneapolis: University of Minnesota Press, 2000.

Stewart, Kathleen. 'Atmospheric Attunements'. *Environment and Planning D: Society and Space* 29, no. 3 (2011): 445–453.

Stiegler, Bernard. *For a New Critique of Political Economy*. Cambridge: Polity Press, 2010.

———. 'Our Ailing Educational Institutions'. Translated by Stefan Herbrechter. *Culture Machine* 5 (2003): NP.

———. *Technics and Time*. Translated by Richard Beardsworth and George Collins. Stanford: Stanford University Press, 1998.

———. *Technics and Time 2: Disorientation*. Translated by Stephen Barker. Stanford: Stanford University Press, 2008.

Strathern, Marilyn. 'Artefacts of History: Events and the Interpretation of Images'. In *Culture and History in the Pacific*, edited by Jukka Siikala, 25–44. Helsinki: Finnish Anthropological Society, 1990.

———. 'Discovering "Social Control"'. *Journal of Law and Society* 12, no. 2 (1985): 111–134.

———. *Partial Connections*. Updated edition. Walnut Creek: AltaMira Press, 2004.

Suárez, Mauricio. 'An Inferential Conception of Scientific Representation'. *Philosophy of Science* 71, no. 5 (December 1, 2004): 767–779.

———. 'Scientific Representation'. *Philosophy Compass* 5, no. 1 (2010): 91–101.

Suppe, Frederick. 'The Search for Philosophic Understanding of Scientific Theories'. In *The Structure of Scientific Theories*, edited by Frederick Suppe, 3–242. 2nd ed. Champaign: University of Illinois Press, 1977.

Suppes, Patrick. 'A Comparison of the Meaning and Uses of Models in Mathematics and the Empirical Sciences'. *Synthese* 12, no. 2/3 (1960): 287–301.

Sutter, Herb. 'The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software'. *Dr. Dobb's Journal* 30, no. 3 (2005): 202–210.

Taylor, Charles. *Philosophical Arguments*. Cambridge Mass.: Harvard University Press, 1997.

Teller, Paul. 'Twilight of the Perfect Model Model'. *Erkenntnis* 55, no. 3 (2001): 393–415.

Thrift, Nigel. *Non-Representational Theory: Space, Politics, Affect*. London: Routledge, 2008.

Tiles, Mary. *Bachelard: Science and Objectivity*. Cambridge: Cambridge University Press, 1984.

———. 'Technology and the Possibility of Global Environmental Science'. *Synthese* 168, no. 3 (2009): 443–452.

Tjiattas, Mary. 'Bachelard and Scientific Realism'. *The Philosophical Forum* XXIII, no. 3 (1991): 203–210.

Toon, Adam. 'The Ontology of Theoretical Modelling: Models as Make-believe'. *Synthese* 172, no. 2 (2010): 301–315. doi:10.1007/s11229-009-9508-x.

Turkle, Sherry. *Simulation and Its Discontents*. Cambridge: MIT Press, 2009.

Turner, Stephen. *The Social Theory of Practices: Tradition, Tacit Knowledge, and Presuppositions*. Chicago: University Of Chicago Press, 1994.

Urry, John. 'Complexity'. *Theory, Culture & Society* 23, no. 2–3 (2006): 111–115. doi:10.1177/0263276406062818.

Virilio, Paul. *The Information Bomb*. London: Verso, 2006.

Wagner, Roy. 'Figure-Ground Reversal Among the Barok'. *HAU: Journal of Ethnographic Theory* 2, no. 1 (2012): 535–542.

———. *The Invention of Culture*. Revised and expanded edition. Chicago: University of Chicago Press, 1981.

Weisberg, Michael. 'Who Is a Modeller?' *British Journal of Philosophy of Science* 58 (2007): 207–233.

Winsberg, Eric. *Science in the Age of Computer Simulation*. Chicago: University of Chicago Press, 2010.

Wittgenstein, Ludwig. *Philosophische Untersuchungen/Philosophical Investigations*. Translated by G. E. M. Anscombe, P. M. S. Hacker, and Joachim Schulte. Rev. 4th ed. Chichester: Wiley-Blackwell, 2009.

Zammito, John H. *A Nice Derangement of Epistemes: Post-Positivism in the Study of Science from Quine to Latour*. Chicago: University of Chicago Press, 2004.