Forthcoming in Distinktion: Journal of social theory

The difference a method makes: Methods as epistemic objects in computational science

Matt Spencer

Centre for Interdisciplinary Methodologies, University of Warwick, Coventry, United Kingdom

m.spencer.1@warwick.ac.uk

For special issue on Abstraction, edited by Celia Lury, Ana Gross and Mike Michael

The Difference a Method Makes: Methods as epistemic objects in computational science

Computational science is intrinsically interdisciplinary; the methods of one scientist may be the objects of study for another. This essay is an attempt to develop an interdisciplinary framework that can analyse research into methods as a distinctive kind of epistemic orientation in science, drawing on two examples from fieldwork with a group of specialists in computer modelling. Where methods for simulation are objects of research in their own right, they are distinct in kind to the objects of simulation, and raise a different set of sociological and philosophical questions. Drawing on the historian Hans-Jorg Rheinberger's theory of epistemic objects, I ask: what kind of epistemic object does a method make, and how is research organised around it? I argue that methods become objects of research as purposeful things, in terms of their enrolment in the intentional structure of the experimental system. And, as methods research tends to be interventionary, in the sense that its mode of study creates and modifies its objects, we therefore observe a practical recursion, a dynamic of scientific reinvention, a "tuning" of experimental systems that sheds light on the form of these systems' historicity, their differential self-reproduction.

Keywords: epistemic things; science and technology studies; methods; scientific practice; abstraction; radical empiricism; recursion, computational science

Introduction

Methods are not just things that scientists use. In some situations, methods themselves become the focus, object and inspiration of research. This paper tries to develop an understanding of intentionality in scientific practice that can account for the efficacy of research into methods, a form of research that does more than produce knowledge; it produces new forms of science. The essay is motivated by experience conducting fieldwork with a group of computational scientists between 2010 and 2012, and I draw on two small examples from that time to illuminate the discussion.

Method has long served a "meta" role for philosophical and sociological commentators on science, notably as a means to debate and demarcate what counts as science and what does not (for example, Feyerabend 1993, Lakatos 1999). For science and technology studies in turn, a great deal rides on the foundational discovery that science is not quite as methodical as "official accounts" represent it to be (Collins 1974, Law 2004: 18-19), opening up the analytical relevance of direct examination of what happens in practice. However, where methods are *situated* in particular knowledge practices (Suchman 1987, Haraway 1988), embedded in scientific cultures (Knorr-Cetina 1999) and materially embodied in objects and apparatuses (Pickering 1995, Radder 2003), they also tend to become less visible within analysis. In contrast, a recent set of approaches in sociology and anthropology focus on methods as worldly "devices" in their own right, situated among their objects, blurring the line between how and what is studied (Lury and Wakeford (2012); see also Riles (2000, 4)). It is in this turn from Method (with a capital "M") to method (with a lowercase "m") that we may examine the analytical stakes of exploring method as the object of scientists' concern, instead of taking it as *my* object. There are three key implications from this that could be picked out:

The first is to draw attention to epistemic pluralism within research settings. Not all scientists working in the field of numerical simulation are focussed on studying things with computer models. A converse orientation of research aims to discover things about the various "how"s intrinsic to modelling. This extends the key insights of Merz (1999) that computational science is epistemically "multiplex" in nature.

Secondly, my two empirical examples both show methods research reworking the mathematical and computational abstractions that furnish the laboratory. The second implication is to contribute to the discussion animating this special issue as a whole, demonstrating that a study of methods research is one place we might get to grips with a somewhat paradoxically "less abstract" approach to abstraction, grounding it as a concrete practical achievement.

The third implication of the analysis is to appreciate that research "turned in" towards methods drives the "tuning" of experimental systems more broadly[1]. If Method as the implementation of standardised procedures implies the repetition of the same, *methods research* in contrast consists of modes of *invention* of scientific practice, a form of recursive looping in which scientists act upon the purposeful structures of their practices, a loop that forms at the level of the laboratory an intrinsic motor of historicity. We are familiar with the idea that methods are performative; they create objects and worlds. But when methods are themselves the object, this performativity loops back to the wider structure of research itself.

Methods as objects of science

Where computational science receives philosophical attention, computer modelling has been the main focus of interest, for example in debates over whether simulation is a new kind of science (Frigg and Reiss 2009, Humphreys 2009, Winsberg 2010), or the role of modelling in practice (Morgan and Morrison 1999, Magnani and Nersessian 2002) and the relationship between models and theories (Suppes 1960, French and Ladyman 1999, Hesse 1966). More generally, simulation has come to stand in a relation of synecdoche to computational science, the part that stands for the whole. But these kinds of shorthand are precisely what an examination of practice can call into question. In my fieldwork with Imperial College's Applied Modelling and Computation Group (AMCG) in London, a significant part of the research going on in this context did not fit into the category of studying external objects by simulating them; on the contrary these scientists were studying methodological unknowns that are embedded in the infrastructures of modelling. That we might examine a "multiplex" configuration of differently oriented epistemic activities is an approach inspired by Martina Merz's study of research at CERN. Merz noted that software components, in some contexts relatively unproblematic instruments, become in other contexts the "question generators" in their own right (Merz 1999, 296).

The "unknowns" that generate questions often serve as indices of an external Nature, and indeed in computer modelling (in contrast to methods research), the phenomena under investigation, whether they are natural, ideal, or engineered, are all external to the laboratory in their own various ways. Methods research in contrast is characterised by a "how" relation acting as a seed around which a project crystallises, an index of a reflexive relation to practice rather than an outside.

These questions, of how research orients itself towards its objects, how "unknowns" emerge as the focus of study, have themselves been prime inspirations for the study of scientific practice. Michael Polanyi developed his theory of tacit knowledge to account for the unformalizable "act of commitment" in which scientists anticipate that something is there to be found (Polanyi 1983, 25). But Polanyi's emphasis on the personal level of explanation risks falling back on a behaviourist inscrutability. The idea that "tacit foreknowledge of as yet undiscovered things" (Polanyi 1983, 23) is something an individual *has*, risks making this issue one of individual psychology, simply displacing the mystery into learned personal dispositions. The power of Hans-Jorg Rheinberger's approach to epistemic objects[2], is that does much the same work as tacit knowledge, but situates these emerging unknown objects of scientific investigation at a super-personal level, within historically unfolding experimental systems amenable to empirical analysis (Rheinberger 1997b).

To understand Rheinberger's contribution we need to appreciate that the objects of science are not objects in the everyday sense, things that can be picked up and brought into a laboratory, placed on the bench, and the techniques of science applied. Nor are we to understand epistemic objects as the stabilised results or referents of science. In contrast to the objects of linguistic reference or ostention, epistemic objects are deferred; "we *cannot yet* point at them" (Rheinberger 2005, 407). Emerging within and driving forward research in the making, epistemic objects invite study and inspire research. Experimental systems form around them. They are material, but tentative things characterised by "irreducible vagueness… [which] is inevitable because, paradoxically, epistemic things embody what one does not yet know" (Rheinberger 1997b, 28). In Karin Knorr-Cetina's words, these objects of science exhibit "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" (Knorr-Cetina 2001, 190). If they are objects, they are objects, to use Latour's pun, that can object (Latour 2000).

Rheinberger's account goes beyond classic studies of scientific practice that showed the material-practical codings and "externalised retinas" involved in making things visible (Lynch 1988, Goodwin 1994), by turning instead to the "displacing dynamics" by which an epistemic object is "absent in its experimental presence" (Rheinberger 1992, 310). Rather than relying on the metaphor of vision, Rheinberger deploys cybernetic and evolutionary metaphors (322). Research happens in differentially reproducing (or, we might add, persisting) experimental systems, which are evolving, "future-generating machines", operating in conditions of noise (312). But what account of the dynamics of experimental systems is capable of accounting for this evolution, without collapsing it into predetermination or aimless drift? Methods research, it seems, is a really important part of the story.

To see methods as epistemic requires a twist on Rheinberger's account, which portrays a primarily successional relationship between epistemic and technical objects: what was once vague and ambiguous becomes later stabilised and enrolled in experimental systems as an instrument. Electrons, in Hacking's example, eventually becoming things you "can spray" (Hacking 1983, 23; Rheinberger 2005, 407). Merz observed that in computational science, the trade goes in both directions. The object, she says, "keeps oscillating within a space that is delimited by its physics content ("epistemic object") and its black-box features ("technological object")" (1999, 314).

We need to be cautious about black boxes. The form of multiplex at stake here, where techniques, instruments or infrastructures become epistemic in computational science, is as embodiments of method, rather than in terms of their content (physics or otherwise). The black-box metaphor, can easily become an epistemological obstacle, to use Bachelard's term (2002), when allied with the idea of "content", and the resultant "container schema" applied to significance. What is significant when techniques are subject to epistemic scrutiny and intervention, *what it is* about these techniques that draws in an investment of attention, the "unknown" that they mark or embody, is not in any simple sense "inside" them, and certainly not guaranteed by metaphorical boundaries of containment, but emerges "contextually" or situationally from their enrolment in an experimental system. Another way to put this is that the way methods matter as objects of study is not confined to the efficient causes of their composition but extends to the final causes of their purposeful enrolment in the experimental system. A better topological intuition for methods research, where the "how" of science becomes object *and* practice, a mode of practical recursion, may be a Mobius strip (see Lury (2013)).

Some methods made epistemic in this way are mathematical, and my first example concerns numerical methods research, the study of making the continuous discrete[3], a field in applied mathematics that studies the mathematical "how" that underlies a first sense of "simulation" as the solution to otherwise intractable problems of physical theory. Other methods made epistemic are computational, and I present a second example in which my informants were engaged in re-configuring the abstractions of a simulation code, the infrastructural "how" implied in a second sense of "simulation" as a collaborative scientific activity.

Numerical Methods

The Applied Modelling and Computation Group (AMCG) is a computer simulation research group with primary specialisms in modelling computational fluid dynamics (CFD) and radiation transport. Most of the simulations created at AMCG are applications of the Finite Element Method (FEM), a method for numerically solving problems described in calculus. Fluids problems, to which the research in both of the examples below relates, are described by equations (the Navier-Stokes equations and their kin) that are well established but which cannot be analytically solved except in a small number of idealised cases. AMCG's *Fluidity* framework uses FEM to compute approximate solutions by "discretising" the problems. Typically, the spatial domain is represented in the model by a tetrahedral mesh of elements, the variables (such as pressure and velocity) are represented by a set of polynomial functions, and their evolution over time is represented by calculations over a series of discrete timesteps. Simulating the dynamics of a system thus involves computing approximate values across this mesh as they change over time, some of which are saved in memory to be analysed and visualised as the simulation output.

One area of methods research that is prominent at AMCG is the study of the various discretisation schemes that can be devised for the finite element method. The choice of polynomial functions (for example, linear or quadratic) to represent variables, and whether they are continuous or discontinuous on the grid, can make a large difference to the characteristics of the resulting simulation, particularly its speed as a computation and its accuracy as a solution, in ways that vary according to the characteristics of the phenomena being modelled. One of my informants related a story about the study of the mathematical properties of discretisation schemes that provides us with a good starting point for considering how a method gathers attention as an object of research.

> "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 configuration: "In this case [a colleague - 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". 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, this scientist (CE) collaborated with a colleague (WS) in order to investigate the method further. "[We] did some numerical investigations... and it was working kind of unaccountably well" (CE).

The most obvious epistemic objects in these scientific settings are the phenomena explored through computer modelling: oceanographic phenomena, reactor designs, turbulent flows, flooding, tidal phenomena, atmospheric phenomena. But here, the *working well of this discretisation scheme* becomes an epistemic object, equal in harbouring lack, potential futures of research, and inspiration of investigation, yet as a methodological epistemic object, it is distinctive for its lack of external referent. It is not just that the scheme works, but that it works "kind of unaccountably well": the "how" embodied in the scheme is not merely a technical ground for research, but an emergent unknown, driving the formation of research around the future it invites. A vast array of tools, substances and apparatuses, work well without inspiring further research; this not because they have been exhaustively studied and exhaustively known, but because the unknowns they may embody haven't (yet) mattered. It is thus not just how they function but how they matter that is their situational character. As one of the senior scientists put it:

> "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 it's 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)

The relation to the future, to what will "push you forward", is a relation, ultimately, to the purposeful structure of the research, to what will make a difference to the experimental system's web of concern. Mattering in science is *more than instrumental* (in the sense of serving a role in a synchronic functional configuration), it is future-bound (serving a role in the future realisation of the experimental system). In the case of CE's discretisation, it happened that he stumbled on a surprising phenomenon that led him towards a proof, without requiring this "year of working out".

> "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 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." (CE)

A surprise such as this steadiness of the result, is routinely treated as a sign of equipment malfunction (Spencer 2012), but once reasons are found to take it as an event pertaining to the object, it opened the door to a surprisingly neat mathematical treatment. Surprising because as CE put it "all the most famous proofs of this kind of thing are very difficult and involve a sequence of about fourteen inequalities," but in this case, via a back and forth movement between computational testing and paper proof, he discovered a much neater proof of the general properties of the discretisation scheme, showing its properties make a difference precisely in the kinds of problem that crop up in geophysical applications, thus concretizing the new scheme in the toolbox of Fluidity, FEM and CFD more generally. The result from this study of method came to matter to the technology but also to the discipline, intervening in how this kind of research is done.

Method as object

The relations between the discretization scheme, the problem it solves and the computation, are eventually wrapped up in proof, tried and tested, and embedded into the experimental system, becoming part of the array of techniques upon which future research depends. It is tempting to see these "conditions of possibility" as having been the epistemic object all along, just in uncertain, yet-to-be-established form. But this would read the situation backwards and collapse the futurity of the object into a synchronic structure. Epistemic objects are only analytically visible through a lens of practice, which in my reading requires a radical empiricist sensibility.

Treating science as practice has often been primarily a negative move, serving a deflationary or "irreductive" function with respect to accounts that preselect only certain kinds of actor (nature, for example, or social interests) as epistemologically relevant (Latour 1993). But positive implication of practice-centric analysis is its commitment to find a way of accounting that gives a central role to actuality, the *taking place* of research. The practice turn in this sense can be read as part of a general "radical empiricist" inversion of the critical terms of explanation. Instead of the actual being explained in terms of what is possible and what is necessary, the actual is what explains the possible and the necessary. While often regarded as a legacy of pragmatism, this inversion is similarly apparent in some strains of phenomenology (Todes 2001, xvii) and process philosophy (Auxier and Herstein 2017), and also in the tenet of STS that in order to explain order we should not assume order (Law 1992).

A focus on actuality helps illuminate the performative effects of prospective and retrospective explanation. Once discovered, the mathematical relationship established by CE and colleagues provides a retrospective way of accounting for the "working well" of the method. The necessity embodied in logical implication explains the actual properties of simulations that used this scheme. But this framing cannot explain how the scheme became epistemic in the first place. The retrospective framing *requires* that the vague futurity of objects be bracketed out: a bracketing that is a basic technique of objectivity[4]. The recognition of this has been the pivot of the practice turn in social theory. Objectivity requires "the time to totalize", in Bourdieu's formulation (Bourdieu 1977, 9). The retrospective view of a "transcendental vantage point" in science (Lynch 1990, 482), cannot see its taking place. The eventual mathematical proof explains properties of the method from its necessity as logical implication when represented symbolically, or causal functioning when implemented in a model, but on the other hand, it becomes epistemic insofar as it is more than a mechanism, insofar as it embodies an excess, embedded in a purposeful, concernful endeavour, insofar as the "how" of its working *matters*. The working of the discretisation scheme only becomes an attractor of inspiration to the extent that it matters, that it, to use the famous Batesonism, *makes a difference*.

The falling of functionalism out of favour in social theory during the latter part of the Twentieth Century makes us wary of talking about functions, purposes, teleologies. Indeed, a methodological anti-functionalism is a valuable empirical tool: we must avoid the temptation to see a practice or institution as necessary because it seems to serve a particular end, so that we start to unpick the contingencies, associations, contestations and so on, that make it *our kind of object*.

However, recent moves towards the study and use of "lively" methods introduce new ways of thinking about the purposeful in culture: relations of testing, trialling, transforming, sorting, solving, ordering (Lury 2012, Back and Puwar 2012). If our interest, following these authors, is in the live(li)ness of methods, this is quite different to the "dead" sense of function. What makes these devices lively is their material instantiation of purposes, as emergent micro-teleologies of practice. That they are "for" certain ends does not imply necessity, or that the wider whole "works", but rather speaks of their specific ways of being actual. In this respect it is helpful to consider Ruth Millikan's naturalist theory of "proper functions" (organs, instincts, artefacts *being for* some purpose). Millikan relies on a rejection of the conceptual analysis of "in principle" causal mechanisms, emphasising instead the observation that all actual purposes arise out of specific (and indeed overwhelmingly specific) histories of (re)production (Millikan 1989). If methods become epistemic as ways of working that matter, these micro-teleologies, what Pickering called "vectors of cultural extension" (Pickering 1995, 20), are not effects of their internal causal structure alone. Nor are they effects of being represented as such by an agent, or of being designed from scratch as such. They are an effect of historicity, the actuality of their emergence within historical sociotechnical assemblages, conditioned in their various ways by the relative persistence of that laboratory, wider experimental systems, their unique trajectories, and the modes of transformation in science. Recursion is thus at the heart of the becoming epistemic of methods: they inspire research (and thus are motors of historicity) insofar as their working matters (and thus is enmeshed in the historicity of their situations).

If we therefore need to embrace a kind of functionalism to understand methods research, this is not in the sense that *we* as social scientists would want to presuppose that social systems work, nor that we would want to attempt to map out *how* they work. It would instead be in the sense of a respecified functionalism, how these sciences take their own methods, as methods *for* research, embodied teleologies, as their own objects of study and intervention (cf Garfinkel (1991)). We don't need, therefore, to position ourselves on opposite sides of a fence, between *our* practice-centred account, and *their* objective retrospect, but can observe this difference as an inner fold and dynamic, between the prospective making-epistemic of methods, and the objectivity of their results.

Infrastructuring methods

> "So the idea is that we want to create the right abstractions in order to get an optimum performance for this kind of application on different platforms and we want to drive the whole thing from a very high level of abstraction for the user so that your problem specification is very close the mathematical problem formulation, and our abstraction layers basically take care of transforming that into an implementation that makes use of characteristics of the platform that you want to run on" (PY)

A group at AMCG were developing a new software stack for finite element simulations, a project described by its members in terms of the reconfiguration of abstractions that it would involve. Because it modified the interfaces that are "built in" to the technical infrastructures used by members of the group, it also had "social" effects, by reconfiguring how different kinds of specialists work with their objects, and how they interface with one another (this sociality is commonly observed in infrastructure studies—see for example Star and Ruhleder (1996)).

As much as it was the development of a technical system, this was also a study of how to do computational science, reworking along the way fundamental aspects of how computer modelling is practiced. While the numerical methods discussed above became epistemic as a more or less direct result of a difference introduced in the discretisation scheme, in this case the difference emerged from a more indirect shift in the technical milieu, making apparent and subject to questioning what was previously implicit. In this case, the prime driver was the growing challenge of sufficiently optimising code to make use of increasingly parallel and more exotic supercomputer architectures, and thus to continue to gain payoffs from Moore's Law[5]. This made the ways of working that are built into the modelling framework open to questioning: how best to do this kind of science. The group had in previous years reworked software architecture to deal with pressures of collaboration with growing numbers of participants and growing variety, so the posing of questions about how software should be organised was fresh and recent history (see Spencer (2015)).

When I did my fieldwork, it was not yet obvious whether the result of the group's project to reconfigure abstractions was going to be a proof of concept, a set of techniques to be transferred to other frameworks, or whether it would become a new framework for running simulations in its own right. As time went on, it has become closer to the latter, and the resulting package has been given a name: Firedrake. I refer to the project here as the Firedrake project, but note that when most of this research was conducted, it hadn't yet gained that name. I give an overview, drawing on early discussions and later publication.

The Firedrake project was an effort to develop a new set of tools that together redefine how simulations are crafted. That this implies social effects by reconfiguring relations of expertise continued to drive the high-level formulation of the project. "The development of [CFD] software," explains one of the write-ups of this work, "is… increasingly a multidisciplinary effort, and its design must enable scientists with different specializations to collaborate effectively without requiring each one of them to understand every aspect of the system in full detail" (Rathgeber et al. 2017, 2). Firedrake's abstractions were to facilitate an effective distribution of ignorance and interdependence, enabling collaborators of different kinds "not to have to know" about each other's' work.

How do abstractions organise expertise? Abstraction used in this context is very close to the conventional software engineering sense, which owes a lot to modular programming (for example Parnas (1972)). Abstractions are technically achieved separations of concern. The usual separation that matters in software engineering is that between (1) what a component does from the point of view of other interacting components or users, and (2) how it does this at the level of its implementation. The term has semantic flexibility, however. As a thing, *an* abstraction, it is identified with (1) directly: it is a high level, multiply realisable, specification[6] for a component (Degueule, Combemale, and Jézéquel 2017, 66). As an activity, on the other hand, *of* abstraction, it is identified with the relation *between* (1) and (2), abstraction being the "art" of arranging and working with such separations (McDonough 2017, 5). But there is a further inflection specific to the setting of computational science, relating to the way in which software encodes physical theory.

The kinds of simulations crafted by scientists at AMCG can be represented as solutions to mathematically defined physical problems, and this makes it possible to treat mathematical theories of physics as abstractions in a sense analogous to high level specifications of software programs. We usually think of mathematical theories as abstract in a different sense, that of their idealisation: representations in mathematical formalisms are "abstracted" from the real-world phenomena that they describe. However, when computer models of physical processes are built, physical theory can serve a more pragmatic role, enrolled in the design process as a high-level specification of functionality for the simulation (though generally a *partial* specification, and one which may be knitted together with others – see for example Winsberg (2009)). Another way to state this difference is between questions of how physical theory represents reality (see for example Cartwright (1983), Morgan and Morrison (1999)), and on the other hand how physical theory is *represented in* computer models[7].

This helps us to appreciate what is meant when traditional CFD code is described by computational scientists as being written at a "low level". The hierarchy of imagination works "upwards" from the physical device, to the machine code, and higher-level languages and processes that run on it, nested layers or levels of abstraction that define different ways of working. Fluidity is written in FORTRAN, which is usually considered a relatively low-level language, just one up from assembly code, but in this context it is not the language itself but what is expressed in it that is low level: what is written in code is the algorithm to solve the equations. The (algorithm for the) solution is written in code, rather than the problem itself. The latter, defined in the continuous mathematics of calculus, remains on paper.

While there is a separation between problem and solution in terms of their media, working with the model requires moving back-and-forth between them, as the algorithm written in code must be handled in terms of being a solution to a specific problem. This situation creates cognitive and social challenges. They put it like this: "Although hand coding algorithms at a low level can produce efficient code, that approach suffers from several serious drawbacks. The key among these is a premature loss of mathematical abstraction: the symbolic structure of differential equations, function spaces, and integrals is replaced by loops over arrays of coefficient values and individual floating-point operations. Interspersed among these are parallel communication calls, threading and vectorization directives, and so forth" (Rathgeber et al. 2017, 3). It is challenging to think back and forth between the calculus and the algorithm, the paper and code. It can be hard to debug, and difficult to optimise. It raises problems of interdisciplinarity because some forms of optimisation do not just require effective ways of handling the software; they need differently specialised collaborators, who may well have little knowledge of fluids or indeed of calculus, and who may, in the words of one informant, inadvertently "break the abstractions" when working with a traditional low level CFD code. The goal with Firedrake, he said, was to enable computer scientists, and particularly those specialised in advanced compiler techniques for parallel or exotic architectures, "not to have to know" about the problem being solved. Likewise, the specialist in the domain being modelled using Firedrake does "not have to know" the details of how the solution to their problem is computed.

Firedrake incorporates several key components to refashion the practice of modelling, but one which stands out as particularly relevant here is the use of a "high level" software language for Finite Element simulations, Unified Form Language (UFL). UFL provides a medium of expression, in code, that is close to that of the calculus. "Rather than writing directly executable code utilizing library calls to access functionality, the numerics of the finite element method are specified purely symbolically in a special-purpose language. A specialized compiler or interpreter then uses this input to generate low-level, efficient code" (Rathgeber et al. 2017, 4). At one side, UFL provides a new language for the modellers to work in. At the other, Firedrake takes this UFL formulation as an input and from it (and from other inputs such as the geometric mesh for the spatial domain, boundary conditions, and so on) generates lower level code and data structures in formats amenable to manipulation by computer scientists. UFL thus enacts a shift from specification of the problem in code rather than on paper, enabling scientists to write calculus (or something very close to calculus) as code.

That these boundaries of abstraction and implementation are negotiable is important. Not only are they negotiable, but they are subject to continual reworking, as methods research spans not only how software should be developed for science, extending to the relations between the media of code and theory themselves. This is a deeper architecture of relations that we see if we unpack the monolithic catch-all of "practice", and if we ask of infrastructures not what is their content, but what by what modes of transformation they become "live".

The "working that matters" that becomes epistemic in this, more infrastructural, form of methods research, is more expansive and diffuse than what we saw in the numerical methods example, where the difference made by the way discretisation schemes work is somewhat more local. This infrastructure embodies ways of working that are not just how the technology works, but how the research group works, and even the broader relations of disciplines. One could say that what matters in this form of methods research is the distribution of cognition itself; not the static distribution of cognition, but how it may become, the future-bound possibility embodied in the experimental system as a distribution working upon itself. The epistemic object in this Firedrake research, is not just how to create a modelling package as a functional infrastructure, but yet-to be realised potential for the (re)organisation of science.

We are now in a position to reflect on why methods research seems conventionally secondary to modelling. Their objects matter because they are for the ends of other epistemic activities. Methods as objects do not make indices of an outside "nature". And because methods research is interventionary and reflexive rather than representational, like other forms of "synthetic" research it doesn't fit the central stereotype of science (cf Hoffmann (2007), Barry (2005)). Finally, the packaged-up results of methods research are routinely enrolled in epistemic strategies for "sanctioning models" (Winsberg 1999), whether this be in terms of inferences about the efficacy of a software package, or a proof about the properties of a discretisation scheme, so in conventional representations they play a supporting role to modelling.

As a result a specific form of demarcation is an issue within research groups like AMCG, but not between science and non-science. Some of my informants would complain about the fact that research councils seemed to favour giving grants to projects with goals aligned with modelling at the expense of methods research. They would also note the converse symmetries: "It is a classic chicken and egg situation with the mathematical/numerical/computational methods and applications: which drives which" (GT). Modelling and methods are "chicken and egg" however, not in the direct sense of mutual causation or precession, but in the sense of their converse symmetry of conditions and inspirations, the trading places of technical and epistemic objects. While modelling relies on the application of stable methods, so the study of methods relies on the use of a set of stabilised representational relationships, validated "test case" simulations, that are used as instruments for the exploration of their practicalities and efficacy.

Tuning practice

A concept of intentionality is crucial to an adequate account of scientific practice. "Research" is research about something. But we needn't collapse this intentionality into linguistic terms, based on how propositions are about things (Rheinberger 2005, Bloor 2005), nor is it helpful to think solely in terms of personal level intentions. Following Rheinberger, our account of the various materials, technologies, practices, bodies, concepts, persons, and so forth, that take part in the taking place of research, must account for how certain *things* become foci towards which the whole assemblage is arranged. These epistemic objects provide a practice with its orientation: it is them it is about.

But methods aren't material things; they aren't like substances in test tubes. As epistemic objects they are the enrolment of techniques in intentional endeavours. The recursive objects of methods research are the micro-teleologies of research itself, the workings that matter.

Following Rheinberger, epistemic objects emerge as focal points of research in the unique pathways of unfolding historical research ensembles. Historicity, this emergence out from a unique trajectory, is *how* the experimental system is actual. But this uniqueness does not imply arbitrariness, for these systems are oriented towards their objects and the happening of those objects' emergence. It is telling that Rheinberger uses normative language to gesture to experimental systems' ways of being good, in the sense of "good for" this emergent objectuality, "It is in the fabric of properly "tuned" experimental systems," he says, "that scientific events materialize" (1997a, 426-427). How, then, does this "being properly tuned" come about? What are the modes of self-making, of self-transformation, of "tuning", for science? What are the "topological invariants" of practice in science (Lury 2012, 248)?

We have seen two examples here: numerical methods unpicking the "how" of an algorithm's working, and infrastructural methods intervening in the "how" of the wider research practice's working, in both cases this "working" matters as more than efficient cause; they matter in terms of their final cause, the working for, the being for, the entanglement of techniques in research with research into techniques, and both with research's being about its objects. In both cases we can detect a prospective attention towards ways that methods make a difference, which sits alongside, or converse to, retrospective accounting practices that wrap up numerical methods in proof and infrastructural methods in testing and benchmarking. But, to the extent that the "product" of research is the (re)production of the laboratory, and not just its formal paper-based outputs, this is not an elision or eclipse behind a dominant objectivity. If we think about "the" scientific method as a universal protocol, there is indeed nothing particularly interesting about the (re)production of the laboratory. There is no phenomenon, nothing to explain, because that multiply-realisable specification (Method) could be implemented at will. But once we move towards an "anarcho-rationalism" of multiple situated methods (cf Hacking 1982), then both persistence and change within this "universe of drifting, merging, and bifurcating [experimental] systems" (Rheinberger 1997b, 181) raise the question of how the accomplishment of science becomes an issue for it, how in everyday work it folds upon itself in self-making contortions.

Footnotes

Included below references ↓

References

Auxier, Randall E, and Gary L Herstein. 2017. *The Quantum of Explanation: Whitehead's Radical Empiricism*. Vol. 9: Taylor & Francis.

Bachelard, Gaston. 2002. *The Formation of the Scientific Mind*. Translated by Mary McAllester Jones. Manchester: Clinamen Press.

Back, Les, and Nirmal Puwar. 2012. "A manifesto for live methods: provocations and capacities." *The Sociological Review* 60 (S1):6-17.

Badiou, Alain. 2007. *Being and Event*. Translated by Oliver Feltham. London: Continuum.

Barry, Andrew. 2005. "Pharmaceutical matters: The invention of informed materials." *Theory, Culture & Society* 22 (1):51-69.

Bloor, David. 2005. "Toward a sociology of epistemic things." *Perspectives on Science* 13 (3):285-312.

Bourdieu, Pierre. 1977. *Outline of a Theory of Practice*. Translated by Richard Nice. Cambridge: Cambridge University Press.

Cartwright, Nancy. 1983. *How the Laws of Physics Lie*. Oxford: Oxford University Press.

Collins, Harry M. 1974. "The TEA set: Tacit knowledge and scientific networks." *Science studies* 4 (2):165-185.

Degueule, Thomas, Benoit Combemale, and Jean-Marc Jézéquel. 2017. "On language interfaces." In *Present and Ulterior Software Engineering*, 65-75. Springer.

Feyerabend, Paul. 1993. *Against method: Outline of an anarchist theory of knowledge*. London: Verso.

French, Steven, and James Ladyman. 1999. "Reinflating the semantic approach." *International Studies in the Philosophy of Science* 13 (2):103-121. doi: 10.1080/02698599908573612.

Frigg, Roman, and Julian Reiss. 2009. "The Philosophy of Simulation: Hot new issues or same old stew?" *Synthese* 169:593-613.

Garfinkel, Harold. 1991. "Respecification: Evidence for locally produced, naturally accountable phenomena of order, logic, reason, meaning, method, etc. in and as of the essential haecceity of immortal ordinary society (I)—an announcement of studies." *Ethnomethodology and the human sciences*:10-19.

Goodwin, Charles. 1994. "Professional vision." *American anthropologist* 96 (3):606-633.

Hacking, Ian. 1982. "Language, Truth and Reason." In *Rationality and Relativism*, edited by Martin Hollis and Steven Lukes, 48-66. Oxford: Blackwell.

Hacking, Ian. 1983. *Representing and Intervening: Introductory topics in the philosophy of natural science*. Cambridge: Cambridge University Press.

Haraway, Donna. 1988. "Situated knowledges: The science question in feminism and the privilege of partial perspective." *Feminist studies* 14 (3):575-599.

Hesse, Mary B. 1966. *Models and Analogies in Science*. Notre Dame: University of Notre Dame Press.

Hoffmann, Roald. 2007. "What might philosophy of science look like if chemists built it?" *Synthese* 155 (3):321-336.

Humphreys, Paul. 2009. "The Philosophical Novelty of Computer Simulation Methods." *Synthese* 169:615-626.

Knorr-Cetina, Karin. 1999. *Epistemic Cultures: How the sciences make knowledge*: Harvard Univ Pr.

Knorr-Cetina, Karin. 2001. "Objectual Practice." In *The Practice Turn in Contemporary Theory*, 184-197. London: Routledge.

Lakatos, Imre. 1999. "Lectures on scientific method." *For and against method*:19-109.

Latour, Bruno. 1993. *The Pasteurization of France*. Translated by Alan Sheridan and John Law. Boston: Harvard University Press.

Latour, Bruno. 2000. "When things strike back: a possible contribution of 'science studies' to the social sciences." *The British Journal of Sociology* 51 (1):107-123.

Latour, Bruno, and Steve Woolgar. 1986. *Laboratory Life: The Construction of Scientific Facts*. Princeton: Princeton University Press.

Law, John. 1992. "Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity." *Systems practice* 5 (4):379-393.

Law, John. 2004. *After method: Mess in social science research*: Routledge.

Lury, Celia. 2012. "'Bringing the world into the world': the material semiotics of contemporary culture." *Distinktion: Scandinavian Journal of Social Theory* 13 (3):247-260.

Lury, Celia. 2013. "Topological sense-making: Walking the Mobius strip from cultural topology to topological culture." *Space and Culture* 16 (2):128-132.

Lury, Celia, and Nina Wakeford. 2012. *Inventive methods: The happening of the social*: Routledge.

Lynch, Michael. 1988. "The externalized retina: Selection and mathematization in the visual documentation of objects in the life sciences." *Human studies* 11 (2-3):201-234.

Lynch, Michael. 1990. "Allan Franklin's Transcendental Physics." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 2:471-485.

Magnani, Lorenzo, and Nancy J. Nersessian, eds. 2002. *Model-Based Reasoning: Science, technology, values*. New York: Kluwer Academic.

McDonough, James E. 2017. *Object-Oriented Design with ABAP: A Practical Approach*. Pennington: Apress.

Merz, Martina. 1999. "Multiplex and Unfolding: Computer Simulation in Particle Physics." *Science in Context* 12 (02):293-316. doi: 10.1017/S0269889700003434.

Millikan, Ruth Garrett. 1989. "In defense of proper functions." *Philosophy of science* 56 (2):288-302.

Morgan, Mary, and Margaret Morrison. 1999. *Models as Mediators: Perspectives on natural and social sciences*. Cambridge: Cambridge University Press.

Parnas, David Lorge. 1972. "On the criteria to be used in decomposing systems into modules." *Communications of the ACM* 15 (12):1053-1058.

Pickering, Andrew. 1995. *The Mangle of Practice: Time, agency, and science*. Chicago: University of Chicago Press.

Polanyi, Michael. 1983. *The Tacit Dimension*. Gloucester: Peter Smith.

Radder, Hans. 2003. "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.

Rathgeber, Florian, David A Ham, Lawrence Mitchell, Michael Lange, Fabio Luporini, Andrew TT McRae, Gheorghe-Teodor Bercea, Graham R Markall, and Paul HJ Kelly. 2017. "Firedrake: automating the finite element method by composing abstractions." *ACM Transactions on Mathematical Software (TOMS)* 43 (3):24.

Rheinberger, Hans-Jörg. 1992. "Experiment, difference, and writing: I. Tracing protein synthesis." *Studies in History and Philosophy of Science Part A* 23 (2):305-331.

Rheinberger, Hans-Jörg. 1997a. "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:S245-S254.

Rheinberger, Hans-Jörg. 1997b. *Toward a History of Epistemic Things: Synthesizing proteins in the test tube*. Stanford: Stanford University Press.

Rheinberger, Hans-Jörg. 2005. "A Reply to David Bloor: "Toward a Sociology of Epistemic Things"." *Perspectives on Science* 13 (3):406-410.

Riles, Annelise. 2000. *The Network Inside Out.* Ann Arbor: University of Michigan Press.

Spencer, Matt. 2012. "Image and Practice: Visualisation in Computational Fluid Dynamics Research." *Interdisciplinary Science Reviews* 37 (1):86-100.

Spencer, Matt. 2015. "Brittleness and Bureaucracy: Software as a Material for Science." *Perspectives on Science* 23 (4):466-484.

Star, Susan Leigh, and Karen Ruhleder. 1996. "Steps toward an ecology of infrastructure: Design and access for large information spaces." *Information systems research* 7 (1):111-134.

Suchman, Lucy A. 1987. *Plans and situated actions: The problem of human-machine communication*: Cambridge university press.

Suppes, Patrick. 1960. "A Comparison of the Meaning and Uses of Models in Mathematics and the Empirical Sciences." *Synthese* 12 (2/3):287-301.

Sutter, Herb. 2005. "The Free Lunch is Over: A fundamental turn toward concurrency in software." *Dr. Dobb's Journal* 30 (3):202-210.

Todes, Samuel. 2001. *Body and world.* Cambridge, Mass.: MIT Press.

Winsberg, Eric. 1999. "Sanctioning models: The epistemology of simulation." *Science in context* 12 (2):275-292.

Winsberg, Eric. 2009. "A Tale of Two Methods." *Synthese* 169 (3):575-592. doi: 10.1007/s11229-008-9437-0.

Winsberg, Eric. 2010. *Science in the Age of Computer Simulation*. Chicago: University of Chicago Press.

[1] The term "experimental system" is taken from Rheinberger (1997b), on whose theoretical framework I rely in this paper. Methods are commonly taken as objects of sociologists' or philosophers' study: the power of Rheinberger' theory lies in enabling the analysis of methods as scientists' own objects of study.

[2] Rheinberger uses both "epistemic thing" and "epistemic object" more or less interchangeably. To avoid multiplication of terms, I follow Knorr-Cetina and primarily use "object" for the purposes of the present discussion (Knorr-Cetina, 2001).

[3] One of the fundamental issues in numerical methods is finding ways to solve problems that are defined in calculus (i.e. in terms of the continuum) in the discrete mathematics that describes the operation of digital computers.

[4] One of the central observations of Latour and Woolgar's classic study of endocrinology research practice was this relation between becoming a fact and shedding of modalities that refer it to its conditions of production (Latour and Woolgar 1986). "[T]here is an essential difficulty writing the history of a fact; it has, by definition, lost all historical reference." (1986, 108)

[5] Although Moore's Law continues to drive increasing density of transistors on a chip, the power of individual processors has stabilised, so Moore's Law translates to increasingly parallel computing rather than faster individual processors (Sutter 2005).

[6] "Specification" in this context refers to the "high level" design of a component, what it does.

[7] When physical systems are modelled in simulation work, simulations are not simply created in their full complexity. They are, instead, built up iteratively, developing a set of verified, validated and benchmarked "precursor" simulations of increasing complexity, that compare against other data models, encoding some of these into automated tests for new functionality, and developing visualisation and diagnostic techniques to provide means of working with the output data. These processes test the relationship between the simulation and the "target" system it represents. There is therefore a threefold relation of representation in any theory-derived simulation, triangulating the physical theory with the target system and the computational system. I focus here on just one side of the triangle: that between physical theory and the computational system, because it is the way that this relationship is organised by the abstractions of the modelling framework, that was at stake in the Firedrake project. Other infrastructural projects could unpick the other sides. For example, the development of automated testing frameworks organise verification and validation of simulations, by automating judgement against comparator datasets.