Explorations in the indeterminacy of computation: an interview with M. Beatrice Fazi

Article (Accepted Version)

This article has been accepted for publication in *Theory, Culture & Society* (SAGE).

If quoting, please refer to the published version.

Official URL: https://doi.org/10.1177/0263276420957054

Explorations in the Indeterminacy of Computation: An Interview with M. Beatrice Fazi

David Beer and M. Beatrice Fazi

Abstract
This interview with M. Beatrice Fazi explores in detail her work on computation. Focusing in particular upon her recent publications, it covers the themes of contingency and indeterminacy. The questions explore Fazi's perspectives on computational aesthetics, abstraction and experience. Through an interrogation of the conceptual insights that Fazi's recent work offers, the interview outlines an agenda for future work in the philosophy of computation and sets forward a series of conceptual policies for seeing the digital, software and data in a fresh light.

Keywords
abstraction, aesthetics, computation, indeterminacy, media theory

M. Beatrice Fazi has recently published a wide-ranging and penetrating philosophy of computing. The book *Contingent Computation: Abstraction, Experience, and Indeterminacy in Computational Aesthetics* (Rowman & Littlefield International, 2018) explores indeterminacy in computing and how it can be theorised. The book aims to break with common conceptions of computer technologies and replace these with perspectives that capture the contingency present within these systems. This interview explores some of the key themes from the book and examines the central arguments that Fazi's text develops. Thinking of computation beyond the digital, it provides a series of conceptual angles into a range of features of computation.

David Beer: It seems that your book seeks to unsettle some of the dominant ideas about computation. From the outset you bring in the concepts of indeterminacy and contingency. Perhaps you could start by explaining why you bring in these concepts and what you are using them to do in framing your argument.

M. Beatrice Fazi: The book offers a philosophical study of computation, which I address as a method of organising reality through logico-quantitative means. The systematisations of computation are generally considered to be simple reductions that capture the complexity of the world. From this perspective, computation is assumed to merely appropriate reality; in consequence, there is said to be no novelty in computation, but only the repetition of the pre-programmed. *Contingent Computation* challenges that view. I propose that computation is not a mere epistemic reduction. By engaging with the formal, logico-quantitative and axiomatic character of computing, I argue that computational processes are dynamic and generative because they have the potential to actualise themselves.

You are right to say that the book wants to unsettle dominant ideas about computational technologies. I introduce the expression 'metacomputation' to describe the enduring belief that rational calculation can a priori represent and explain reality. I address this metacomputational view by considering the long-standing philosophical pursuit of a *mathesis universalis*, i.e. a

universal science of reasoning. I stress the isomorphism between this ancient quest and modern attempts to automate thought, originally proposed by mathematical logic and now instantiated in computing machines. Post-industrial informational economies and societies that push for enlarging the domain of what can be quantified, calculated and hence commodified, or cognitive science's computationalist theories that equate minds and computers – these are just some of the ways in which metacomputation operates today. If 'software is eating the world', as it is often commented, this feast is arranged with the metacomputational confidence that deductive calculation can encompass and account for all that can be thought of and acted upon. It should be noted, however, that calls to resist such metacomputational 'enframing' (to use a well-known Heideggerian term) are equally based on the assumption that to compute is to reduce or approximate a reality that extends much further than what can be formally rendered. This is true within philosophy: let us consider, for instance, philosophical oppositions to the automation of sensation and thought, which stress the extra-calculatory idiosyncrasies of both feeling and thinking. However, this is an assumption that is shared by science too whenever it works to circumvent formalism's limitations by making computational systems more akin to the many variabilities of life, culture and nature. So, whether the technocultural project of metacomputation is endorsed or rejected, there seems to be a bilateral consensus on an equation that understands computing as an instrumental reduction of life's complexity through the use of finite algorithmic determinism.

With *Contingent Computation*, I wanted to move beyond this metacomputational understanding of computing: I addressed computation beyond the instrumental, representational and cognitivist purposes that have been assigned to it. While wanting to supersede metacomputation, however, I also challenged traditional critiques of it, insofar as these accept that computation is what the metacomputational project made it to be. My main philosophical concern was demonstrating that computation's processing should be understood not as a matrix of total determinism but in terms of discrete processes of self-determination. I argued that computation is a process of determining indeterminacy. This indeterminacy is logical, quantitative and internal to computational processing. It is the indeterminacy that is inscribed into the formal and mathematical definition of an algorithmic procedure and that, as such, does not have to simulate the indeterminacies of life or lived experience. *Contingent Computation* then offers an alternative to metacomputation not by discarding the logico-quantitative character of computing. On the contrary, the book engages with computational axiomatics through a novel philosophical reading of the ontological significance of computability theory's foundational papers (specifically, those of Kurt Gödel and Alan Turing). While the metacomputational view implies that computational systems compose a pattern of 'closed' logical necessity, the book philosophically mobilises Gödel's theorisation of the incompleteness of axiomatic systems and Turing's notion of incomputability to argue that this pattern is in fact 'open' to contingency qua indeterminacy. Incompleteness and incomputability show that indeterminacy pertains to the formal too. In this sense, they reveal to us computation's own contingency. This proposed conceptualisation of contingency in computation provides a new perspective from which to engage with the calculative infrastructure that underpins 21st-century society, while reworking the ways in which it is also possible to challenge that infrastructure.

DB: To explore this, you focus on three areas across the book: aesthetics, abstraction and experience. Could you explain a little more about how these three work together, and why you felt it necessary to supplement the exploration of aesthetics, which is a central focal point in your book, with discussions of both abstraction and experience? Also, am I right that you place aesthetics as central to your analysis of computation?

MBF: Yes, an explicitly philosophical understanding of aesthetics is central to the analysis of computation that the book proposes, and this investigation of aesthetics is a thread throughout the

book that also allows me to achieve other speculative goals. *Contingent Computation* responds to some of the theoretical challenges that contemporary debates in and on digital aesthetics are confronted with. In my work, I often talk of these challenges in terms of an impasse: that between the qualitative features that belong to the perceptual, sensuous receptivity that aesthetics cares for, and the quantitative modes of formal and logical organisation that computing instead relies on. This is an impasse that I describe as a deadlock between the continuity of sensation and of lived experience, on the one hand, and the discreteness of digital technologies, on the other.

In order to overcome this impasse, a reconceptualisation of aesthetics in computation is necessary. This reconceptualisation must be contextualised vis-à-vis current debates in digital culture that interpret aesthetics not in terms of a theory of art, but in a manner that is closer to the etymological roots of the word, which lie in the ancient Greek *aisthesis*. To put it simply, aisthesis is a theory of sensory knowledge. 20th-century philosophy has developed the prospect of that sensory knowledge by stressing an affinity between aesthetics and ontology. That aesthetics concerns ontology and ontology concerns aesthetics is something I also argue for. More specifically, I draw on Gilles Deleuze's proposal for an aesthetic ontology. Deleuze's proposition is original in the history of thought, insofar as it constructs aesthetic investigations as openly metaphysical inquiries into the possibility of the new. Aesthetics, in this sense, concerns creation. Since I wished to demonstrate the possibility of ontological production in computation, my guiding question for *Contingent Computation* has thus been: if we want to address computation according to this ontological understanding of aesthetics, what should we be doing?

The book engages with this question by reworking the relation between aisthesis and logos in computational systems. I do not deny the importance of sensation (and of related notions such as matter and embodiment) for digital culture and digital aesthetics. However, I insist that sensibility is not enough. Insofar as digital technologies employ strategies of formal abstraction, digital computation has a relation with the intelligible (i.e. what is apprehensible only through forms of abstractive activity). This relationship is formal, extensive, informational, and does not directly resemble an energetic, intensive plane of affects. An aesthetics of the digital must be able to speak of the materialities of computing while also accounting for such formalised and formalising relations with the intelligible. The book then stresses the limits of what I call an 'aisthesis of the digital' in attending to this formal, abstractive and logical dimension. Largely, these are also the limits of Deleuze-inspired engagements with digitality, which I address by considering the 'affective turn' in media studies. I insist on stressing why, for Deleuze, a 'digital aisthesis' would be a contradiction in terms. Positions that look at digitality through a Deleuzian lens want to construct said digital aisthesis, but they bypass a fundamental aspect of Deleuze's critique of computational logos. Deleuze's antagonism towards computers must be understood vis-à-vis his argument for the ontological superiority of sensation over formalisation. Implicitly or explicitly, what the affective turn in media studies ends up doing is endorsing such an ontological superiority of the analogue. This endorsement is inherent to affect theory's tendency to transduce the quantitative into the qualitative, or the extensive into the intensive. Ultimately, an aisthesis of the digital engulfs logos by 'virtualising' it. In my view, however, it is a mistake to virtualise the ontology of digital computation in this way.

Focusing on abstraction and experience is central to my study of the relation between sensibility and intelligibility in computation, and to my effort to mend the fracture between aisthesis and logos in digital culture. According to an aisthetic perspective on computational aesthetics (which can be seen to correspond to the affective turn in media studies but that is also maintained by some phenomenological approaches), aesthetics concerns ontological indeterminacy, for ontological indeterminacy is what brings about the new. Aesthetics, as creation, concerns the rapport between determination and indetermination. Yet indeterminacy, according to an affective view, cannot be grasped or rendered by techno-informational mediations; rather it can only be lived through sensation, and via the immanence of thought to the latter. According to this view,

indeterminacy is thus abstract because it is beyond any possible technological capture of the generative approximations and uncertainties of the living and the lived.

Elaborating this onto-aesthetic relation between abstraction and indeterminacy in computing is a crucial step in my investigation of computational aesthetics. According to an affective approach, abstract indeterminacy can only be felt: this is the indeterminacy of Deleuze's virtual. In my work, however, I resist such a virtualisation of computation because virtualising the digital involves sidestepping the logico-quantitative specificity of computational systems. I thus engage with abstraction and indeterminacy in a different manner. I consider how formal, logical, quantitative abstraction (and not the Deleuzian virtual abstract) is fundamental to what the computational is and does. In other words, I reconceptualise and reassess precisely the instrumental techno-enframing of abstractive formalisation, which has been traditionally the object of much philosophical critique. Formal abstraction in computing does not dislocate and deterritorialise but orders and organises, and it does so through what I call 'formalisation-as-discretisation'. 20th-century formalisations of the informal notion of computation have made abstractive discretisation a specific method of calculation. Resolving the fracture between aisthesis and logos involves recuperating this discretisation and placing it at the core of the aesthetic investigation of digital technologies.

The relevance of notions of experience for the kind of onto-aesthetic study of computation that I propose is directly connected to this issue. Drawing again on Deleuze, I take aesthetics to pertain to the conditions of real experience. However, since I am looking at the aesthetic legitimacy of computation's calculative operations (something that, it is worth repeating, Deleuze would never have considered as aesthetic), I advance a related conceptualisation of computation's own experience. In other words, not the experience of a user or programmer, and not the experiential grounds of culture, society, art. Instead, in the book I investigate the possibility for computation to hold a level of experienceability that is specific to its logico-quantitative character. In this sense, one of the central speculative operations that I carry out involves looking at how computation's abstractive discretisations might construct the experience of computation (i.e. computation's own experience) beyond or before the possibility of an associated milieu between not only machines and humans, but also between ontologies of the mechanical on the one hand, and ontologies of the lived, on the other.

DB: The different conceptual angles that you point at raise questions about empiricism. Can you explain how you have built an empiricism that is able to account for hard-to-capture things like indeterminacy – or that is able to encapsulate the very different aspects of aesthetics and experience? Also, what do you mean by the term 'computational empiricism'?

MBF: Empiricism ties knowledge to experience. In my view, that is something to which it is difficult to object. However, it is crucial to assess what one takes experience to be, and define what one wants empiricism to do. My empiricism (like all empiricisms) is bound to the way in which experience is approached and understood. I consider experience not in terms of a collection of sensory contents or a stream of phenomenal data. I also take seriously Kantian questions about the categories of experience and the search for its conditions of possibility, together with the way in which post-Kantian philosophy has elaborated these questions and this search. You are right to say that the empiricism that I had to build is one that would be able to account for the different aspects of aesthetics and experience that I consider in the book. I needed an empiricism that could address computation's operations of abstractive discretisation in their logico-quantitative specificity; that is, an empiricism that could engage with the formalisms of computing. Obviously, computation is a reality in the world; I object, however, to a sense-data-centred mode of empirical enquiry that overlooks computation's conceptual capacity. Or, to put this in the terms that I used earlier, computation's relation with intelligibility. The empiricism that I built had to be able to

account for the formal indeterminacy of computation vis-à-vis computation's capacity for abstraction (and hence, discretisation).

Although he is critical of sense-data empiricism, in this respect too Deleuze (and his 'transcendental empiricism') could bring me only so far when assessing computational systems. This is still because of the way in which indeterminacy, in his philosophy, hinges on the immanence of thought and sensation. I thus turned to the 'radical empiricism' of Alfred North Whitehead. While many have stressed the affinities between Whitehead and Deleuze, I underlined the philosophical differences between them, and showed how the onto-aesthetics of Whitehead can be used to 'radicalise' that of Deleuze. Perhaps it will not come as a surprise, at this point in our conversation, that this operation concerns the relation between abstraction and experience. In Deleuze, abstraction (as a method of generalisation and simplification) is understood as an onto-epistemic capture that obstructs and burdens experience because it separates thinking from the intensive indeterminacy of feeling. In Whitehead, by contrast, abstraction is a method of construction that organises past actualities and idealities, since these are both fundamental to what experience is. Deleuze's and Whitehead's different views on abstraction reveal one of the core disparities between the two philosophers: for Deleuze, to experience is to move away from determination; for Whitehead, to experience is to move towards it. Both Deleuze's transcendental empiricism and Whitehead's radical empiricism propose an understanding of experience that does not limit the latter to the psychological and intentional domains of human experiencing, yet the key point to consider here is that the conditions of experience are dissimilar in Deleuze's and Whitehead's respective philosophies. For Deleuze, the differential relations in intensity expressed through virtual sensation endow reality with its transcendental conditions. Whitehead instead described very actual operations of determination that constitute what experience is. This Whiteheadian approach to the conditions of experience is more suited to develop my hypothesis that computation might be said to experience in its own terms, namely via quantitative processes that are the actual – and yet formal – operations of computation itself.

The empiricism that I propose thus detaches experience from both the early-modern empiricism of the senses and the transcendental empiricism of virtual sensation. My conceptualisation of computation's own experience, moreover, differs greatly from what I call 'computational empiricism'. I propose this expression to denote a technoscientific view on computational systems that makes these systems as close as possible to an empirical reality. As a way of exemplifying this attitude, I focus on computational techniques that are said to be 'unconventional', 'natural' and 'non-classical' because of the non-standard reworkings of the determinism of computing that they offer. These techniques propose alternative understandings of the computing machine, which however still assume a quite orthodox equivalence between computation and onto-epistemic reduction. Because traditional computing is understood to be limited and limiting in dealing with the many empirical variabilities that are external to their pre-programming, what are looked for are opportunities for that pre-programming to deviate from pre-determination, and thus to be more dynamic or more complex, just as nature and life are. Genetic algorithms and evolutionary computation are good examples of such changing grounds for computing, and so are the biologically-inspired views that inform many present feats in AI, and which also underpin the current interactive paradigm epitomised by Alexa, Siri and the Internet of Things more broadly.

In *Contingent Computation*, I take David Hume as the philosophical frame of reference for computational empiricism, insofar as these computing practices recall the associationism between abstraction and experience (and the externality between thought and sense-data) that he promoted. In order to problematise computational empiricism, I consider Whitehead's critique of Hume and of the 'sensationalist principle' of early-modern empiricism. I tackle computational empiricism's comparable reduction of experience to sense experience, together with its elaboration of change as accidental variability. On top of that critique, however, and in order to open up a novel commitment to empiricism in and for the philosophical study computation, I also divorce the

concept of the contingent from that of the empirical. I do not reject the evidence that computational systems might include, and also be modelled upon, the empirical mutability of the real world, and through interactive and situated operations, these computational systems might calculate better, faster and stronger. We are witness to this every day, no matter whether these operations are Google searches on a smartphone or quantum calculations in a state-of-the-art research lab. Equally, I recognise that it is possible to develop a successful theoretical approach that would look at the implementations, performances and applications of algorithmics into codes and programmes, and to how these instances are contingently embedded within culture and society. Many of these approaches have or are being developed, for instance, in the fields of software studies and science and technology studies (STS). While these positions usefully stress computation's existence in its situated environmental complexity, with *Contingent Computation* I open another road towards the experience of the computational. I needed an empiricism that could speak of the contingency of computation in its logico-formal character, before turning to the chance events, glitches and accidents of its implementations, or to the computational simulations of the empirical world with the consequent replications and appropriations of the mutability of such empirical world's behaviours.

DB: Self-actualisation is another concept that takes on a crucial role in your analysis. You talk about using aesthetic investigation in order to 'uncover computation's potential for self-actualisation'. I was wondering how computation can be part of self-actualisation and what the potential politics are for forms of actualisation that are computed?

MBF: I understand computation as 'in act' and actual. I am keen to stress this, for such a claim qualifies my position in relation to the many forms of idealism and materialism that are possible when considering digital technology. In the book, I talk of 'computational actual occasions'. This is an expression that I adapt from Whitehead's definition of an actual event. Whitehead died in 1947. In his work, there is no account of computing machines. Nonetheless, his cosmology of quantitative events and discrete processes of abstraction is remarkably relevant to describe the digital ontologically. Central to Whitehead's cosmology are what he called 'actual occasions'. An actual occasion is an atomic occurrence; it is an episode, an event, something that happens in, but also extends through, space and time. An actual occasion is a unit of reality that begins and ends. Most importantly, an actual occasion is a process of self-determination that is ingressed by indeterminacy. This self-determination corresponds to the occasion's self-actualisation, carried out both at the level of the sensible and at the level of the intelligible. An actual occasion involves both physical and conceptual operations, conveying distinct yet related ontological determinations. Significantly, these operations are aesthetic insofar as they are 'prehensions': actuality's grasping, seizing or holding of other actualities (in 'physical prehensions'), as well as of idealities (in 'conceptual prehensions').

By considering computational processes in terms of actual occasions, I assign to them the same self-actualising capacity and prehensive character. I propose that 'computational actual occasions' determine themselves via the physical manipulation of data (in other words, by physically prehending other computational actualities), but also because they address logically – and not affectively or strictly empirically – their own logical indeterminacy. By drawing on Whitehead's ontological schema, I have thus developed a theoretical framework that allows me to extend the aesthetic investigation of computation from the sensible to the intelligible and to understand these two dimensions as related and yet not immanent to each other. Via this computational aesthetics at two levels, I can focus on the generative potential of the logico-quantitative operations of computational systems. Here is then the relevance of the notion of self-actualisation for the aesthetic investigation of computation: in addition to aesthetics concerning creation (as I already mentioned), I understand creation and creativity precisely in the ontological

terms of self-determination. The production of novelty in computation corresponds to the process of self-actualisation of the computational procedure itself.

While a Deleuzian reading of Whitehead downplays the extra-affective character of conceptual prehensions, I have stressed it. Via the Whiteheadian notion of conceptual prehension, the book demonstrates that formal abstraction in computation does not stop its generative processuality. Quite the opposite, it is a constitutive condition of it. Self-determination starts from a physical reception of data, but it is only concluded via the 'ingression' (this is a very Whiteheadian term) of a conceptual determination. Such a conceptual determination is a very actual capacity, not codetermined by virtual sensibility and not ontologically inferior to it. My reading of Whitehead, moreover, builds on an implicit definition of experience that is to be drawn from the ontological position of the philosopher. I address experience in terms of self-actualisation. To experience is to produce oneself as a new actuality. Arguably, engaging with Whitehead's philosophy of actuality involves supporting his panexperientialism. I have done that too, yet with a significant proviso: I accept that, metaphysically speaking, all actuality experiences, but I can do that because I address this experiencing in terms of the self-determination of actuality itself. My speculative approach then highlights how, if we take actuality to be dipolar because brought to realisation via physical and conceptual determinations, then we must also develop an understanding of experience that equally involves processes of actualisation ingressed by physical and conceptual indeterminacy. This central point meant that I could develop my claim that computation possesses its own mode of experiencing by basing such experiencing on the abstractive and quantitative nature of the computational method, without the need to resort to a source of indeterminacy that would be external to the computational processing.

What is the potential politics for forms of actualisations that are computed? Undoubtedly, much contemporary social, political and economic power is generated and consolidated through the pervasive mathematisation and datafication of the world, giving rise to forms of what some have called 'algorithmic governmentality', or to extractivist practices geared towards the universalising calculatory capture of lived and living particularities. This tendency towards totalisation via computation, as a mode of commodification too, is predicated on, and also develops, the assumption that computation is that all-enframing *mathesis universalis* that I discussed in your first question. This is an assumption, we said, that a project like mine wants to tackle. My ontological proposition requires not only to consider forms of actualisation that can be computed, but also to address the possibility that computation is or has a mode of actualisation is its own right. That is to say, to consider not only that which can be brought into the realm of the computational via axiomatisation, but also forms of axiomatic computation that are already modes of actualisation. This proposition builds on a metaphysical schema in which there is unity but not totality. What I am proposing is an ontology in which discreteness is the only way in which continuity can exist; an ontology in which the very old (the oldest, perhaps) political question about the relation between the universal and the particular is reworked through that Whiteheadian suggestion for togetherness without a whole.

Obviously, this is a different type of self-actualisation than that of a traditional 'political subject'. And yet, it is still important to see this potential politics in the context of a theory of subjectivity. The latter has to be constructed while rethinking the possibility of a political alliance between the human and the technological. The many 'posts' of modernity have done a lot to reassess precisely that; philosophical posthumanism, for instance, has conceptualised the manner in which it is possible to conceive subjectivity vis-à-vis the hybrid techno-human assemblages of our time. Notably, however, these efforts are often sustained by a metaphysics of vital matter in continuous becoming. My ontology of discrete processes and self-actualisation offers something else in this respect too: when asked to look at a possible coalition politics between the human and the non-human, I start from the prospect of not reaching an ontological conciliation, due to the specific configurations between abstraction and experience that these diverse subjectivities give rise to. It is then not a question of assembling or of coproducing. My proposed understanding of

self-actualisation shows that lived experience is not the only means to engage with process. Strikingly, becoming and computing can go together, even without a vitalist transduction, if we allow for the possibility of a relationality without a whole, or without that common denominator that the virtual is believed to be.

DB: Your discussion of 'computational idealism' was really striking. You build this in a focused chapter in the book, but it then crops up throughout the discussion of abstraction. This got me wondering about where these ideals emerge from, who uses them and to what ends these ideals are geared. I see that determinism is one of the features you associate with it.

MBF: *Contingent Computation* focuses on the logico-quantitative character of computing, yet this attention does not imply looking at computation as if it was disengaged or detached from reality. The book refuses digital Platonism, which makes abstraction a means of onto-epistemological transcendence. I construct an aesthetics of discreteness that aims to speak of the abstractive operations of computational systems. Of course, other aesthetics of discreteness are possible; an aesthetics of 'algorithmic necessity', underpinned by what I call a 'computational idealism' that focuses precisely on such onto-epistemological transcendence, is one of these other possibilities. I address this in a dedicated chapter where I assess positions in technoculture and technoscience that I characterise as idealist because they approach computational structures and procedures as if these were a priori forms of intelligibility. I further that analysis of computational idealism throughout the part of the book that considers directly abstraction because that idealist approach to digital technology puts forth a specific understanding of the abstract that I intend to dismantle. According to this computationally idealist view that I object to, the abstract amounts to the eternal and universal status of logico-mathematical perfection, which in turn signposts a more fundamental type of reality than that which can be accessed through sensation.

Abstraction, then, is believed to be central to formalisation precisely because the formal can direct the empirical while remaining indifferent to phenomenal change. This is a top-down schema that freezes abstraction and experience into a relation of transcendence. It is transcendence that allows for the generality and comprehensiveness of logico-mathematical sciences, and it is transcendence that is trusted whenever society puts its faith in an algorithmic rationality that is believed to never be wrong. Aesthetically speaking (and following a more conventional elaboration of aesthetics than the one proposed in the book), the same transcendence underpins the alleged beauty, simplicity, harmony and elegance of those logico-mathematical sciences to which computing is said to be ancillary. Transcendence, then, highlights that equivalence between logical truth and aesthetic value that programming cultures, for instance, celebrate whenever they look for the shortest and most elegant script, or which the formal and natural sciences pursue when praising the possibility of a 'theory of everything' as revealing a fundamentally 'beautiful' universe. These are attitudes that draw on a long tradition of aestheticisation of mathematics. And these are approaches that I understand as putting logic at the heart of aesthetics. *Contingent Computation*, however, starts from the opposite position: it puts aesthetics at the heart of logic. The book shows that computational logos is already aesthetic. Aesthetics concerns the conditions of real experience, and these conditions are about sensibility and intelligibility.

To continue answering your question, yes, determinism is one of the features that I associate with computational idealism, although this association is not exclusive and needs to be explained. My critique of computational idealism is developed in tandem with a reconceptualisation of the determinist character of computing. I want to think processuality and axiomatic formalism together; this brings me to criticise those 'forms without process' that underwrite computational idealism. However, I can demonstrate that computational axiomatics is complex and dynamic only if I have also shown that the systematisation of reality via a simple method of abstraction (an endeavour that, I said this while addressing your first question, I call 'metacomputation') cannot be

accomplished. My dismantling of ideas of 'aesthetic computing' that promote the aestheticisation of idealised formal simplicities over the complexity of actual processes of formal discretisation is geared toward providing such evidence. This also explains why I address computational idealism together with the onto-epistemological principle that underpins it, which I call 'Universal Computation'. I consider a philosophical lineage that marries idealism with the universalising Leibnizian project of *mathesis universalis*. Universal Computation is a 20th-century reworking of such a rationalist endeavour, in which the apriority of logico-mathematical formalisation remains key, for it is meant to account for a horizon of total (universal, indeed) operativity. Philosophically, idealism of the Platonic kind can again be seen as a reference point here, insofar as this is a realist yet transcendent ontology of the universal. Wanting to dismantle or surpass traditional aesthetic ideals of beauty, simplicity, elegance and harmony in computing means, then, on the one hand, acknowledging that the pre-programmed and determinist character of computation fits with the idealist scenario that aesthetic computing thrives upon. On the other hand, however, the proposed critique of computational idealism must also engage with such determinism directly, to see whether this can be understood outside of the orthodox terms of a project for universalising totality.

DB: Your analysis moves in its later stages toward limits and potential. This seemed like an important step for capturing the liminality of computation. How do limits and potential work together? Do limits create potential or just constrain it? How does potential in computation facilitate the breaching of limits?

MBF: Yes, the book argues that computation's limits should be considered alongside its potential. *Contingent Computation* engages with the logico-quantitative specificity of digital computers and with the axiomatic character of these technologies. I place the formalisation of the notion of computability within the larger debate about the limits of formal reasoning so as to study the constraints of computational axiomatics. That formal axiomatic systems are always limited because incomplete is something that has been famously demonstrated by the logician Kurt Gödel. In 1931, Gödel's incompleteness theorems showed the existence of statements that are true when viewed from outside a given system, but the provability of which cannot be deduced inside the formal axiomatic system in question. This moment in the history of formal reasoning points to another foundational moment in the history of computing, when the mathematician Alan Turing demonstrated that some functions are incomputable. Drawing on Gödel's incompleteness theorems, Turing wanted to find a mechanical and formally expressed process that could set the standard for deductive procedures. This attempt, published in the 1936 paper 'On Computable Numbers, with an Application to the *Entscheidungsproblem*', phrased the computational method in terms of an algorithm capable of solving a problem in a finite number of sequential steps. What Turing discovered, however, is the existence of problems that cannot be solved via finite algorithmic means.

Traditionally, incompleteness and incomputability have been used as evidence of the pitfalls and failures of formal axiomatic systems. On my philosophical reading, however, the incomplete and the incomputable become a mark of computation's ontological potential for self-actualisation. In proposing this, I build on the onto-aesthetic view that takes indeterminacy to be central to the production of novelty. Having found another indeterminacy within computational formalisation, I can also argue for another potentiality in the ontology of computing. Just like the indeterminacy of computation is different from the indetermination of life, of living systems and of lived experience, so the potential of the computational procedure is to be found in the latter's logico-mathematical specificity. Incompleteness and incomputability are negative results that can be mobilised to dismantle the metacomputational project. The setback of metacomputation becomes the positive outcome upon which the book constructs another ontology for the actuality of computation. The limits of computation thus do not strictly create or constrain computation's potential: they are its potential. This potential, in turn, is not about breaching limits. What I am

describing is not the computing machine doing something that it is not supposed to do. This is not about accidents or glitches; equally, this is not about chance. So, although we are talking about the limits of the computational, I do not characterise the latter as 'liminal' in the sense in which this term is largely used. The incompleteness of axiomatics and the existence of the incomputable do not push computing at the threshold, margins or in the in-between of more or less stable phases. This indeterminacy is not a mid-phase for computer science, and these limits are not temporary precarities either.

Contingent Computation understands computational operations as processes of self-determination. This computational self-determination calls for another type of ontology: a generative space outside of the energetic and the intensive, and in fact internal to what is informational and extensive. I can return here to a point we have already touched upon in this interview: how I object to ontological virtualisations of digital computation. It is important to repeat that vis-à-vis the book's examination of a potentiality that is specific to the capacity of discrete computational processes to actualise themselves. The task of establishing an aesthetics of contingent computation is that of investigating computation's potential for self-actualisation. Such potential of the discrete computational procedure can vouch for computation's aesthetic validity because it confirms its generative capacity. This generative capacity, I want to add, can be described with that beautiful Whiteheadian expression that calls for the 'internal adventure of becoming' – where becoming belongs not to a continuous whole but to discrete parts. This is not the becoming of the virtual continuum then, but neither is the becoming described by computational empiricism, for which computational 'facts with no thought' are connected and endowed with the possibility of variation only thanks to an external source of reception that collects and represents them as phenomena. Via this philosophical study of the potentiality of computation, I am addressing change as an internal principle of dynamic eventuation. Deleuze's virtuality is arguably such an internal principle in and for affective ontologies; since the virtual excludes computation's logico-quantitative specificity from the conditions of real experience, however, virtuality cannot be the potentiality of computing. Whitehead's ontological schema is, once again, more suitable than Deleuze's for the way Whitehead distinguishes, and yet relates, the sensible and the intelligible, assigning to them different potentialities to describe how actuality is constituted by and via its internal relations.

Whitehead calls actuality a decision amid potentiality. An actual occasion's prehensive activity is, for Whitehead, this deciding, and just as there are two types of prehensions (physical and conceptual), so there are two types of potentiality, 'real' and 'pure'. Real potentiality is actuality's potential to create new actualities beyond what already exists. Through physical prehensions, actual occasions from the past become the data that enter and inform the constitution of a future actual occasion. Self-actualisation, however, is never concluded without a conceptual determination too. We discussed conceptual prehensions already. I need to add now that actuality also determines itself in relation to the pure indeterminacy of unknown idealities (what Whitehead calls 'eternal objects') that are to be conceptually prehended. In a parallel way, the book argues that, alongside the 'real potentiality' of what I describe as computational actual occasions, the computational process includes the 'pure potential' of the incomputable. *Contingent Computation* thus stresses computation's logical relation with an indeterminacy that corresponds to the evaluation of a pure possibility – that of incomputability. This philosophical reading of the incomputable as the pure potential of computation is a speculative reworking of Turing's result, and is meant to prompt a philosophical reconsideration of the pre-programmed character of computing: the computational process has this internal real and pure potential for self-actualisation precisely by virtue of the fact that it does what it is supposed to do. To say this in the vocabulary of the book, a computational actual occasion has an 'end', both in terms of an aim and a completion. This end is its self-production. My proposed philosophical investigation of the limits and potential of computation thus affords one of the book's key ontological contentions: the reliance of

computational systems on pre-programmed rules does not prevent, but in fact enables, the generation of novelty.

DB: As you raise it in a dedicated chapter, and given the current circumstances, I have to ask about 'factuality'. Are these forms of computation somehow changing or redefining conceptions of objectivity and subjectivity? What does this mean for how computation and changes in computation impact on what is seen to be a fact?

MBF: Actuality is not factuality – not, at least, in the sense of 'brute' and 'bare' factuality that Whitehead criticised. This critique should be read together with Whitehead's suspicion towards what he called 'scientific materialism' (i.e. positivism), and it links back to his negative assessment of Hume's empiricism. Just like the latter, the positivist epistemologies of scientific materialism associate a stream of perceived facts, external to the subject who perceives them, with those impressions and ideas that the same subject has when interpreting those facts. Positivism celebrates data as actualities. For Whitehead, however, data do not give us the actual, but only quantifiable records of it. Whitehead warned against the 'fallacy of misplaced concreteness': data are an abstraction from the immediacy of experience. To use a contemporary vocabulary: there is no such thing as 'raw data'. Yet Whitehead's point is not a socio-constructivist argument about data as artefacts; his is rather an onto-epistemological constructivism that stresses how experience always abstracts. There is no such thing as non-abstractive access to facts.

Contingent Computation addresses the manner in which factuality is conceived within computing, especially in those life-oriented, matter-oriented alternative paradigms underpinning computational empiricism, which understands actuality in terms of a concatenation of value-free sense-data. Building on Whitehead's argument, I emphasise how these data entering unconventional, natural and non-classical computation are not actuality per se but measurable records of it. I then address emergentism (that is, the belief in, or the theory of, emergence) as the ontology that subtends these approaches. Alternative computing's fascination with emergentism has to do with the behavioural unpredictability that the notion of emergence describes. Significantly, emergentism focuses on explanatory frameworks that arise from the observation and simulation of those real-world systems from which alternative computing induces new methodologies of information processing. Contingency, however, is understood in terms of the chance occurrences of the physical world, and possibilities for computational complexity and dynamism are consequently modelled upon those of the empirical plane of factuality.

In the book, I argue that emergentism ontologises computation as a 'fact with no thought'. This is an expression that I formulate to denote the naturalisation of computation into an empirical phenomenon. You ask me whether computation influences how we can comprehend factuality. It is on this understanding of computation as a 'fact with no thought' that I need to linger in order to answer that question. What I describe as computational empiricism demotes experience to a collection of empty facts in search of an external principle of interpretation, thus situating the mental valuation of this factuality outside of the constitution of the fact itself. This has consequences. Let us consider, for example, how we live in 'hyper-positivist' times, with many contemporary techno-practices supporting an understanding of science as a quantifiably objective process of numerical standardisation. The humanities, as a set of disciplines but also as a socio-political project, are asked to comply with this view. Digital methods are then often acritically invoked to provide a fictional 'view from nowhere'. Of course, the pitfalls of this a-perspectival understanding of objectivity have been highlighted abundantly. In this respect, it is important to note that the 'facts with no thought' of 21st-century data societies threaten not only humanities disciplines but everything that falls under the remit of theory and speculation (and thus part of the hard sciences too), for instance in the name of computational instruments that could replace hermeneutics, or even the scientific method, with pattern recognition.

Speaking still in terms of disciplines, it is worth noting how critical theory has done much to resist positivism's focus on bare factuality, dismantling assumptions that data can speak for themselves. *Contingent Computation* endorses those critiques of instrumentality; my conceptualisation of contingency in computation, for example, shows that axioms cannot explain away everything, as they cannot even explain themselves. And yet, my approach also differs from traditional critical theory's evaluations of algorithmic rationality. This is because I do not accept its equating computation's logico-quantitative specificity with that reductionist enterprise that much cognitivism – and cognitive capitalism too – wants computation to be. Determining the role of factuality in relation to the actuality of computation is thus part of my speculative effort to rethink the limits of computation alongside its potentiality. The Whiteheadian analysis of the interrelation of facts and forms is crucial precisely when considering such a potentiality. By 'forms' here I do not mean the a priori transcendent entities of computational idealism. Forms, in my proposed philosophy of computational actuality, are rather 'forms of process': again, this has to be interpreted in a Whiteheadian sense, which highlights the internal constitution of actuality via a conceptual labour with the indeterminate, thus pointing to a structure of actualisation that is specific to the process of determination itself.

Addressing such interrelation of factual existence and schemas of abstraction helps me to overcome the impasse between the qualitative and the quantitative in computational culture, but also that between a symbolic order and a lived one upon which postmodernity has been stuck. Highlighting such interrelation, however, is also important in relation to those 'current circumstances' that you mention – and I believe that you are referring there to urgent debates about fake news and a post-truth society. A philosophy of actuality of the kind that I propose never tires of affirming the importance of theoretical work against declarations of its death or inutility. In my proposed view, abstracting is not leaving the world for ivory towers; on the contrary, abstraction is how we enter the world and construct it in terms of its actuality. Finally, refusing to assign to abstraction the status of mind-dependent representations of a mind-independent world, as Whitehead teaches us to do, is a means of defence against many cheap relativisms that aliment current practices of disinformation.

DB: To close, perhaps you could say what you think are the next crucial developments in theorising computation. And also, what are you looking to focus your work on now that the book is complete?

MBF: Future theorisations of computation must address the latest developments in automation and respond to the upsurge of interest in artificial intelligence. This renewed attention to the computational invests society, culture and the economy, just like the applications and implications of such technologies do. We are at a crucial moment in the conceptualisation of computational systems. Until relatively recently, a large part of 'new media' events and situations were still addressed through the tropes of communications studies; today, however, words such as 'algorithms' and 'AI' have much currency. It is crucial to see what responses philosophical inquiries can offer to our present and future techno-condition; even more critical is to see whether the humanities, at large, decide to engage at all. Finally, this is a key moment for media theory too – a field of study that is still in the making and whose historical and conceptual rapport to computing is complicated but rewarding.

In this evolving scenario, the issue of computational indeterminacy is more and more relevant. Let us consider a trending topic such as machine learning. This expression denotes a set of computer programs that are said to 'learn' because they can 'teach' themselves to change their own instructions if exposed to large amounts of data. When reflecting on the current hype around machine learning technologies, I am inclined to frame these techniques as part of that computational empiricism that I assess critically in the book. This supposedly 'new way of doing AI' draws from old cybernetic ideas, such as those developed under the connectionist approach –

neural networks, for instance, being the prototypical technology of connectionism, and aiming to build computational systems that are inspired by the brain and the way in which this can adapt and act vis-à-vis empirical change. Having said this, I am nonetheless keen not to dismiss this growing interest in the computational elaboration of real-world indeterminacy. We should obviously resist a promotional rhetoric that neglects the histories of these technologies. However, there remains an important sense in which these AI techniques are effectively proposing something different, not least in their further destabilisation of the already moving ground between autonomy and automation.

At present, I am investigating what I call the 'autonomy of automation' to develop a theoretical approach that considers computational agents no longer in terms of extensions or enhancements of human cognitive faculties. The speculative hypothesis that guides my latest research pertains to the possibility of challenging what I define as the 'simulative paradigm', which has been looming over computational culture since Alan Turing's 1950 proposal of an 'imitation game' as the benchmark of success for thinking machines. I approach this possibility from an ontological and epistemological standpoint. I consider questions about 'being' together with questions about 'knowledge', and vice versa. I talk of onto-epistemology, however, also because I want to stress that, while we are investigating what computation is and what computations does (that is, its ontology) we are also addressing thought, what thinking is and what thinking does. The question of thought is an issue that the computational turn in the 20th century, and the amplification of that turn in the current millennium, have brought to the fore again. For instance, asking whether computation can create or generate anything new, as I did in *Contingent Computation*, involves defining creation and creativity in relation to the eventuation of a new mode of thought into the world; this eventuation implies the actualisation of systems of knowledge that in turn concern structural relations with a reality that happens to be other from (or alien to) previously established epistemic processes, and their related ontologies.

Contingent Computation aimed to rework the relation between contemporary automated systems of calculation and intelligibility. Deleuze, I argued, opened up the prospect of resolving the question of what thought is through aesthetics. My focus on computational systems, and on the explicit axiomatic character thereof, however, brought me to become unconvinced with the Deleuzian flattening down, via immanence, of thought onto sensation. By drawing on Whitehead's ontological schema, *Contingent Computation* radicalised Deleuze's aestheticisation of thought so as to examine that which Deleuze would not have considered to be either aesthetic or a legitimate form of thought, i.e. computational processing, and thus theorise computation's own conceptual capacity. Aesthetics has been the speculative means through which I proved that the undeniable abstractive activity of computation can be addressed beyond the old, yet still powerful, cognitivism that permeates the field of computing, while still considering the formal and quantitative specificity of computation. I am now developing these lines of inquiry into new projects that further my philosophical investigation of what computation and thought respectively are, or together can be.

While studying the aesthetic in computation, I learned that it is necessary to maintain a theoretical focus on intelligibility. I am now expanding this insight into the study of epistemic mechanisations. *Contingent Computation* has challenged conceptions of the computational as a calculatory depiction of the world. My critique of the representationalism of computing, however, is not an attack on representation per se; as we saw, it has to be understood as a critique of metacomputational assumptions. The speculative challenge becomes now that of thinking representation qua abstraction and conceptualisation, and to assess how these operate within the epistemic remit of computation. If *Contingent Computation* showed that computational logos is already aesthetic, I am now focusing on how computational logos is not only aesthetic. In my newest work, the question becomes explicitly epistemological (or onto-epistemological): I am addressing algorithmic procedures and structures that break with old 'images of thought' (to use, one last time, a Deleuzian expression) but also produce new images of what thought processes can be – images that can be independent from human access and only significant to those

machines which have created them. Much of the research I am producing lately – for example, I am writing on the incommensurability of algorithmic thought – focuses on the profound onto-epistemological alterity that the computational can stand for, and addresses the difficulty of any rendition, mediation or even communication between two modes of abstracting (that of humans and that of machines) for which there is no common phenomenological and existential ground. I am thus engaging, for instance, with the critical prospect of understanding what explanation and interpretation might be within the formalising space of computation, at a time that asks us to acknowledge epistemic representations that might arise within calculation but also surpass the boundaries of human cognition.

Biographical Notes

David Beer is Professor of Sociology at the University of York. His books include *Metric Power* (2016) and *The Data Gaze* (2018).

M. Beatrice Fazi is Lecturer in Digital Humanities in the School of Media, Arts and Humanities at the University of Sussex. Her publications include the book *Contingent Computation: Abstraction, Experience, and Indeterminacy in Computational Aesthetics* (2018).