Numerical instability and dynamical systems

Vincent Ardourel [*]

*IHPST,
CNRS/Université Paris 1-Panthéon-Sorbonne,
France*

Julie Jebeile[†]

*Institute of Philosophy & Oeschger Center for Climate Change
Research,
University of Bern, Switzerland*

Abstract

In philosophical studies regarding mathematical models of dynamical systems, instability due to sensitive dependence on initial conditions, on the one side, and instability due to sensitive dependence on model structure, on the other, have by now been extensively discussed. Yet there is a third kind of instability, which by contrast has thus far been rather overlooked, that is also a challenge for model predictions about dynamical systems. This is the *numerical instability* due to the employment of numerical methods involving a discretization process, where discretization is required to solve the differential equations of dynamical systems on a computer. We argue that the criteria for numerical *stability*, as usually provided by numerical analysis textbooks, are insufficient, and, after mentioning the promising development of backward analysis, we discuss to what extent, in practice, numerical instability can be controlled or avoided.

Keywords. Instability – dynamical systems – discretization – computer simulations – initial conditions – model structure – numerical solution

Contents

[*]Electronic address: `vincent.ardourel@univ-paris1.fr`; Corresponding author
[†]Electronic address: `julie.jebeile@philo.unibe.ch`

1 Introduction

Mathematical models of dynamical systems are sometimes prone to instability. Instability is a serious issue in applied mathematics in that it poses problems for model predictions about dynamical systems. Informally, a model of a dynamical system is said to be unstable with respect to a given variable or property if small changes in the specification of the system entail significant changes in the predicted dynamical behavior with respect to that variable or property. Conversely, a model of a dynamical system is stable with respect to a given variable or property if small changes in the specification of the system entail similarly small changes in the predicted dynamical behavior with respect to that variable or property.[1]

Two kinds of instability have tended to be the focus of investigation in the philosophy of science. On the one hand, *sensitive dependence on initial conditions* makes dynamical systems unstable with respect to small changes in the *initial conditions*. This generates instability in some non-linear mathematical models, and is what leads to the famous "Butterfly effect" with its prejudicial impacts on model predictions, extensively studied (e.g. Batterman 1993; Kellert 1993; Smith 1998; Werndl 2009). On the other hand, philosophers of science interested in climate science modeling have more recently focused on *sensitive dependence on model structure*. In this case, *structural instability* is generated

---

[1]A model of a dynamical system can be stable under some respects, and unstable under others. For example, as Corless and Fillion (2013, 554) show, the Lorenz attractor is unstable with respect to the position, but stable with respect to the dimensions of the attractor. In the paper, the (in)stability in models of dynamical systems we discuss mainly concerns the systems' trajectories, and, for the sake of readability, this specification will remain implicit.

by an error in the *model structure*, and the resulting uncertainty, i.e. structural uncertainty, is the discrepancy between the model's predicted trajectories and the observations. The "Hawkmoth effect", so named to indicate it is the cousin of the "Butterfly effect", results from the amplification of this structural uncertainty over time; see Mayo-Wilson (2015), for example, for insights into the notion of *structural chaos*. In particular, this effect lies at the heart of a philosophical debate between, on one side, Frigg and co-authors who argue that structural instability in regional climate models places in-principle limits on our confidence in model predictions (e.g. Frigg, Bradley, Du, and Smith 2014; Frigg, Smith, and Stainforth 2015), and, on the other side, Winsberg and co-authors who put forward objections (e.g. Winsberg and Goodwin 2016).

Yet there is a third kind of instability which also presents challenges for model predictions about dynamical systems. This *numerical instability*, on which we will focus here, stems from the use of numerical methods involving a discretization process, where such discretization is required to solve the differential equations of dynamical systems on a computer. Numerical instability is a frequent and important matter of concern in scientific practice, since numerical methods are ubiquitous in scientific modeling and are required to make predictions in computer-assisted sciences; yet this form of instability has not yet been subject to anywhere near the same level of philosophical investigation as the Butterfly and Hawkmoth effects. The time is now ripe, however, for the philosophy of applied mathematics to develop an analysis of numerical instability of dynamical systems for its own sake; it is this project that this paper encourages and to which it contributes.

The state-of-the-art philosophy of applied mathematics indeed provides ample resources for such an analysis. For instance, it investigates the various kinds of numerical errors affecting computer simulations (e.g. Humphreys 2004; Lenhard 2019); proposes to interpret numerical errors as modeling errors through the perspective of backward numerical analysis (Fillion and Corless 2014; Fillion and Moir 2018); puts forward definitions of the concepts of exact solutions (Fillion and Bangu 2015; Fillion and Corless 2019); explains why approximate numerical solutions are sometimes preferred to exact analytical solutions (Ardourel and Jebeile 2017); and discusses the available methods – e.g. perturbation methods, backward error analysis (Corless and Fillion 2019) or effective validity for inferences (Moir 2019) – used to produce satisfactory approximate solutions.[2]

Our aim here is to investigate numerical instability and the way in which it can hamper model predictions. We begin by offering a brief overview of the two above-mentioned kinds of instability affecting models of dynamical systems and their predictive capacity, and then provide a definition of numerical instability (Section 2). We go on to investigate cases of numerical instability occurring in a wide range of non-linear mathematical models (Section 3), and then argue that the criteria for stability of numerical methods, as usually provided by numerical analysis textbooks, are not always sufficient (Section 4). We finally discuss

---

[2]See also Moir (2013) for discussions on errors and stability in mathematical modeling.

to what extent, in practice, numerical instability can be controlled or avoided (Section 5).

Numerical analysis is an important method by which to tackle numerical instability, and hence in this paper numerical analysis plays an important role. As a field of applied mathematics, numerical analysis has been brought to bear on the study of numerical (in)stability in relation to floating-point arithmetic, numerical linear algebra, and differential equations (e.g., Corless and Fillion 2013; Hairer and Wanner 1991; Higham 1996; Wilkinson 1971). While numerical analysis considers forms of numerical instability generated by the broad class of numerical methods, in this paper we focus on the numerical instability that occurs more particularly in time-evolving trajectories of dynamical systems, described with differential equations and solved with discretization-based methods. Consequently, in this paper we are strictly interested in that part of numerical analysis which deals with differential equations, including the mathematical discussions on *stiff problems*, corresponding to instability in non-linear differential equations (e.g. Hairer, Nørsett, and Wanner 1987).

2 Three sources of instability

In this section we provide a brief overview of the three main kinds of instability that can affect model projections about dynamical systems. Sensitive dependence on initial conditions (SDIC) and sensitive dependence on model structure (SDMS) are at the origin of two kinds of instability that have been extensively discussed in the philosophy of science. We first briefly define these kinds of instability, before turning to the notion of numerical instability on which we focus in this paper.

The three cases have in common the general way instability manifests. Indeed, in all three cases instability is characterized by significant changes in the predicted dynamical behavior induced by only small changes in the specification of the system. The three cases differ in what is changed, namely: (i) initial conditions, (ii) model structure, and (iii) model by discretization.

In the three cases, we consider a dynamical system with time evolution $x(t) = \varphi_t(x(0))$ and initial condition $x(0)$.[3] We suppose that this flow is the solution to the following differential equation with initial condition $x(0)$:

$$\frac{dx}{dt} = f(x) \tag{1}$$

2.1 Initial conditions

The dynamical system is *sensitive to initial conditions* if, for any initial condition $x(0) = a$ and every small arbitrary distance ϵ, there exists an initial condition b close to a within ϵ, and an instant t, such that the difference between the two corresponding solutions of the dynamical system becomes greater than a

---

[3]We can define the dynamical system as $< \mathbb{R}, S, \varphi >$ with S the state space.

quantity Δ, i.e. that $d(\varphi_t(a), \varphi_t(b)) > \Delta$, with d the distance between the solutions (e.g., Mayo-Wilson 2015, 1238; Nabergall, Navas, and Winsberg 2019, 6).

Informally, the dynamical system is said to be unstable if small changes in the initial conditions entail significant changes in the dynamical solution over time. Of course, *how significant* is a vague notion, and in fact depends on the magnitude of Δ.[4] Thus, a system will be more or less sensitive to initial conditions depending on the rate of change. An exponential rate is often considered as a typical rate of change in philosophical discussions, although it is not considered part of the definition of sensitivity on initial conditions. Not only the rate of change but also the time scale are important considerations in the philosophical discussions which seek to assess the consequences of such instability on predictions.

2.2 Model structure

Let us now keep the initial conditions fixed, and consider a change in the model structure, i.e. a change in the formulation of the equations. Roughly speaking, a dynamical system would be said to be *sensitive to model structure* or *structurally unstable* if small changes in the function f involve significant changes in the dynamical evolution.

More precisely, let us borrow the definition provided by Nabergall et al. (2019, 9). They define a space of time-evolution functions (or flows) Φ with a metric δ, the flow $\varphi \in \Phi$, and they introduce a metric d in state space. In that case, Φ is *sensitively dependent on model structure* to degree Δ at φ if for any initial state, there is a time t such that for almost all $\tilde{\varphi} \in \Phi$ with $\delta(\varphi, \tilde{\varphi}) < \epsilon$, one has $d(\varphi_t, \tilde{\varphi}_t) > \Delta$.[5]

In the example developed in (Bradley, Frigg, Du, and Smith 2014; Frigg et al. 2014), not only do the nearby trajectories diverge under model error, but the whole distribution of trajectories – whose initial conditions are within a certain distance – also diverge over time. The authors thus define a probability distribution $p_0(x)$ around x supposed to represent uncertainty on initial conditions, and examine the time evolution of the probability distribution $p_t(x) = \varphi_t[p_0(x)]$ where φ_t is the dynamics of the model.

2.3 Discretization

Let us now keep the initial conditions and the model structure fixed, and focus on equation 1 that needs to be numerically solved on a computer. One numerical method is chosen, among several options, to discretize the differential equation.

---

[4]Practitioners may find the idea of relative error more useful than absolute error. Although the definition of sensitivity to initial conditions usually involves absolute error, it could certainly be revised.

[5]The definition of the structural instability is a question in itself, which is discussed in Frigg et al. (2014), Nabergall et al. (2019), Winsberg (2018, 232). It is out of the scope of this paper to address this question.

Like initial conditions and model structure, *discretization*, we argue, is itself a source of instability.

In this context, a model of a dynamical system – implemented and solved on a computer – will be said to be *numerically unstable* with respect to a given numerical method if the small differences between the original differential equation and its discretized version to be implemented on the computer entail significant changes in the numerical solution over time with respect to the exact solution.[6] For example, let us consider the basic numerical method, the *forward Euler numerical method*. This numerical method transforms the differential equation 1 into the following discrete equation :

$$(x_{k+1} - x_k)/h = f(x_k) \tag{2}$$

where h is the time step. This discrete equation can be written as $x_{k+1} = hf(x_k) + x_k$, which is an algorithm enabling one to successively calculate the value x_1 from x_0, x_2 from x_1, x_3 from x_2, and so on, x_n from x_{n-1}. This leads to a discrete sequence (x_k) for every integers k from 0 to n, which is the numerical solution of the differential equation 1 with the forward Euler discretizing method.

In this case, numerical instability is present when, however small the discretization step h is, a significant discrepancy between the exact solution $x(t)$ and the numerical solution (x_k) appears at some point. More precisely, let $x(t_k)$ be the value of the exact flow (or solution) at the time t_k. The model of a dynamical system is said to be *numerically unstable with respect to numerical method m* and a quantity Δ, if, after discretization with the numerical scheme m, for any arbitrarily small discretization step h, there exists a time t_k such that distance $d(x(t_k), x_k) > \Delta$.

Unlike sensitive dependence on initial conditions and sensitive dependence on model structure, numerical (in)stability does not stem directly and only from certain properties of the dynamical system. It also stems partly from the discretization method being used to solve the equation numerically. Numerical (in)stability comes from the transformation of a continuous flow into a discrete time evolution. Usually, in numerical analysis, numerical (in)stability is deemed to be a property of the *algorithms* or *numerical methods* used to solve the equations (Corless and Fillion 2013, 29, Fillion and Corless 2014, 1464); for example, the forward Euler method is qualified as numerically stable or unstable without any reference to a particular dynamical system. However, while the use of an unstable numerical method likely leads to unstable trajectories of the dynamical system, a stable numerical method can, under some criteria, also lead to uncontrolled errors, as we will see (Section 3). There is therefore an important sense of (in)stability, on which we focus, as being a property of the models of dynamical systems – implemented and solved on computer – *and* relative to the chosen numerical method. In what follows, we further investigate cases of the numerical instability of dynamical systems.

---

[6]The small differences are non-eliminable since the discretization process changes the nature of the model, in that the *continuous* differential equation is replaced by a *discrete* algebraic equation.

3   Cases of numerical instability

Numerical instability due to discretization manifests in various cases of the numerical resolution of differential equations, including ordinary differential equations and partial differential equations.

A notable historical example was reported by meteorologist Norman A. Phillips (1959) who called it non-linear instability. When developing one of the very first general circulation models (an atmospheric model over a large geographical space), he encountered this instability in a numerical solution of the nonlinear barotropic vorticity equation. As illustrated in Figure 1, the numerical computation breaks down abruptly after several days of simulation run.
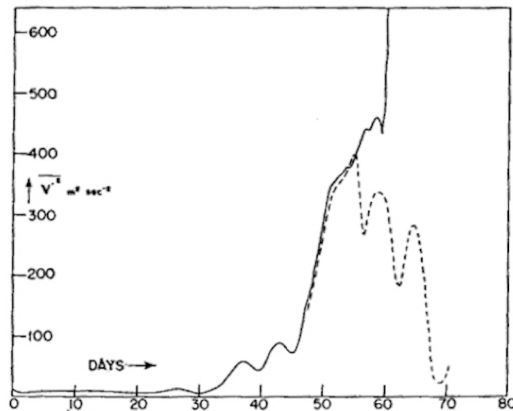

Figure 1: Disturbance kinetic energy as a function of time. The solid curve was obtained without smoothing, the computations breaking down at about 56 days. The dashed curve was obtained by periodically introducing a filtering procedure. (Phillips 1959, 503)

In order to explain this instability, let us consider the following partial differential equation which contains a nonlinear term of type "$u \cdot \frac{\partial u}{\partial x}$", such as $\frac{\partial u}{\partial t} + u \cdot \frac{\partial u}{\partial x} = 0$. This describes a simple configuration of one-dimensional advection, where $u(x,t)$ is the fluid velocity.[7] Suppose that the solution u is such that $u = \sin(kx)$. The non-linear term then becomes $u \cdot \frac{\partial u}{\partial x} = \frac{1}{2}k \sin(2kx)$. This corresponds to a wave whose length is half the length of the original wave. By iteration, the non-linear term "generates" new waves of ever-smaller lengths; this is called the *aliasing effect*. This process mirrors the natural phenomenon of transfer of energy to the smaller scales. The transfer of energy to smaller

---

[7]Advection is the horizontal transport of properties such as heat, temperature, salinity or polluting material, by displacement of a fluid such as water or air. $u \cdot \frac{\partial u}{\partial t}$ is the advected quantity over time; in the context of fluid dynamics, it is the fluid velocity. $\frac{\partial u}{\partial x}$ is its partial derivative against the spatial coordinate x. The fluid velocity, like any quantity in climate models, is decomposed into a Fourier series, i.e. into a sum of cosines and sines. In other words, it is represented as a *wave packet*, each wave being a perturbation.

scales goes on until the smallest wavelengths are produced and then eliminated by viscous dissipation.

The instability occurs when the size of the wave becomes shorter than twice the grid length, because now they can no longer be resolved in the model; this is "the smallest resolvable wave". As Phillips argued, "this instability arises because the grid system cannot resolve wave lengths shorter than about 2 grid intervals; when such wave lengths are formed by the non-linear interaction of longer waves, the grid system interprets them incorrectly as long waves" (Phillips 1959, 501). Yet even if the size of the discretization is taken to be arbitrarily small, there will always be new waves which could not be resolved, since the non-linear term always generates new waves of ever-smaller length. As Phillips emphasizes, "*This "instability" cannot be eliminated by reducing Δt* [the discrete time step]"(1959, 502). This example epitomizes the definition of numerical instability we offered in Section 2.3): however small the discretization step is, a significant discrepancy between the exact solution and the numerical solution appears at some point. In Section 5 we will discuss how it is, nevertheless, possible to overcome such numerical instability.

This numerical instability is connected to the class of *stiff problems*, extensively studied in numerical analysis. Roughly speaking, stiff problems are differential equations for which the typical time scale of the differential equation cannot be "resolved" by the numerical method. In other words, the size of the discretization step is not sufficiently small to resolve the time evolution of the differential equations. If the time-scale variation of the differential equation is unknown, is very small or, even worse, decreases over time, then numerical instability can occur at some point over the numerical resolution. There are many examples of such issues that cannot be discussed in this paper (see e.g. Bui and Bui 1970; Hairer and Wanner 1991; Krivine, Lesne, and Treiner 2007); an example of stiff problem is given in Appendix 7.1.

Moreover, numerical instability can occur even for problems which are not identified as stiff. Even very simple and apparently unproblematic dynamical systems can lead to numerical instability if the computer simulation is sufficiently long. An example comes with the use of another basic numerical method, namely the *backward Euler method*. This method transforms the differential equation 1 into the discrete equation:

$$(x_{k+1} - x_k)/h = f(x_{k+1}) \tag{3}$$

Figure 2 represents the numerical resolutions of the *simple pendulum* by using this method, for different initial conditions. The differential equation is $d^2\theta/dt^2 = -g/l \sin(\theta)$, where θ is the angle. Stern and Desbrun (2008) show that numerical instability occurs in this one-dimensional dynamical system. The exact solutions of the dynamical system are known: they are ellipsoids in the phase space. Yet unlike the exact solutions, the numerical resolutions of the dynamical system with the numerical method significantly deviate from the exact trajectories; we will come back to this problem in Section 5. The explanation of this numerical instability is very basic: discretization errors accumulate at each
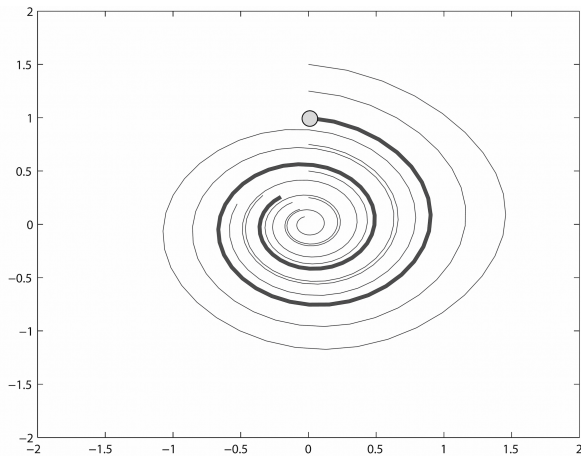
Figure 2: Numerical computations of the simple pendulum equation with backward Euler method. The trajectories are decreasing spirals because of numerical instabilities. (Stern and Desbrun 2008).

step of the numerical resolutions. It is noteworthy that numerical instability can occur in dynamical systems that are not sensitive to initial conditions: as the example illustrates, the simple pendulum is a non-linear problem, but is *not* sensitive to initial conditions since it has periodic solutions. Furthermore the example shows that numerical instability may occur due to the use of numerical methods, even if the methods are supposed to be stable – under some criteria that we will discuss in Section 4.

Given that, usually, the exact solutions to the models of dynamical systems are unavailable, how can scientists know that the models are numerically (un)stable? Criteria for stability are provided by numerical analysis textbooks, and it is to these we now turn.

4 Criteria for stability in numerical analysis

Numerical stability of differential equations is a crucial property for predictions requiring numerical computation. This notion attracts much attention in applied mathematics. We now review the main criteria for numerical stability usually provided by numerical analysis textbooks. We argue that, despite their importance, they are not sufficient to avoid the numerical instability of dynamical systems defined in Section 2.3 and illustrated in Section 3.

4.1 Zero-stability

In numerical analysis, a first important criterion for stability is *zero-stability*.[8] Roughly speaking, a numerical method is *zero-stable* if a small difference between two initial conditions, does not entail significant change to the numerical solution. More precisely, let us consider a numerical method (such as the forward Euler method introduced in Section 2.3) with two initial conditions x_0 and $\tilde{x}_0$. The method is said to be *zero-stable* if there exists a quantity S (named the constant of stability) such that, for any integer k of the two numerical solutions $(\tilde{x}_k)$ and (x_k) built from the two initial conditions satisfy:

$$|\tilde{x}_k - x_k| \le S|\tilde{x}_0 - x_0| \tag{4}$$

when time step h tends to zero. In other words, the distance between the two numerical solutions based on two distinct initial conditions remains bounded by a certain quantity. This first notion of stability refers to "the 'stability' of the numerical method with respect to 'small perturbations' in the starting conditions" (Süli and Mayers 2003, 332).[9]

Zero-stability is a crucial property that a numerical method has to meet. An example illustrating the importance of zero-stability is given in Appendix 7.2. However, there are numerical methods, which are zero-stable, and yet lead to numerical instabilities in the sense defined in Section 2.3. Our point is that a zero-stable numerical method can lead to numerical instabilities depending on the dynamical system to which it is applied. For instance, whereas both cases discussed in Section 3 use zero-stable numerical methods, they exhibit numerical instabilities. In particular, numerical instabilities can occur with a periodic dynamical system, such as the one-dimension simple pendulum system. Such examples reveal that this first criterion is not sufficient to avoid numerical instabilities. Thus, given a dynamical system, it is hard to know in advance whether numerical instabilities will occur despite the use of a zero-stable method. Therefore, a case-by-case analysis is needed.

4.2 Absolute stability

A second important criterion for stability is *absolute stability*. In order to define it, let us consider a numerical method, with a finite time step h, which provides the discrete sequence (x_k) for the numerical solution (such as the forward Euler method introduced in Section 2.3). The numerical method is said to be *absolute stable* if there exists a quantity $C > 0$ such that, for any value of x_k:

$$|x_k| \le C \tag{5}$$

---

[8]So-named because, for the multi-step methods of ordinary differential equations, "zero-stability" can be determined by considering only the differential equation $y' = 0$ (Süli and Mayers 2003, 332). It is also sometimes called *D-stability* in honor of Dalquist (Hairer et al. 1987, 380), or just *stability*.

[9]This presentation is simplified. In particular, it leaves aside small roundoff errors. For more details, see e.g., (Bréhier n.d., 14; Palais and Palais 2009, 137). Moreover, for linear multi-step methods, one needs to take into account the initial values at the different steps (see Süli and Mayers 2003, 332).

In other words, the numerical method is said to be *absolute stable* if the numerical solution (x_k) is asymptotically bounded (Palais and Palais 2009, 137; see also Hairer and Wanner 1991, 40; Süli and Mayers 2003, 348). Unlike zero-stability, absolute stability concerns the behavior of a numerical method as time t tends to infinity, with a fixed step size h. Absolute stability assesses to which extent the numerical solution to a stable problem can diverge, i.e. tends to infinity. Absolute stability does *not* concern small changes in the initial conditions. It only concerns the asymptotic behavior of the numerical solution, specifically whether this solution is likely to tend to infinity.

The condition of absolute stability is particularly important for stiff problems. We discuss this notion in Appendix 7.3, which requires some mathematical details. Here, we contend that, despite its importance, absolute stability is not a sufficient condition for numerical stability based on an example. The simple pendulum problem, as presented in Section 3, can be affected by numerical instability even while it makes use of an absolute stable method, namely the backward Euler method that we defined in Section 3. In long-run numerical computations, the use of absolute stable methods can even lead to discrete solutions that significantly depart from the exact solution. The reason is that absolute stability guarantees that numerical solutions will not diverge. However, numerical solutions can still deviate to exact solutions without diverging, such as in the case of the simple pendulum where the numerical solutions are decreasing spirals (see Figure 2).

In a nutshell, the main criteria traditionally provided by numerical analysis textbooks, namely zero-stability and absolute stability, are insufficient conditions for numerical stability. However, a promising alternative approach called backward error analysis has recently attracted philosophical attention, and will be briefly presented in what follows.

4.3   Backward error analysis

We have so far discussed the predominant approach in numerical analysis textbooks, which can be called "forward error analysis". However, another approach, developed notably by Wilkinson (1971) and later by Corless and Fillion (2013), called "backward error analysis", provides the conditions for an *a posteriori* assessment of the numerical stability of the model of a dynamical system. Two concepts play an important in this approach: the backward error and the conditioning. An assessment of both is required to study whether the trajectories of the model of a dynamical system remain sufficiently close to the exact solution.

In the predominant forward error analysis, the main error to estimate – the forward error – is the discrepancy between the exact solution and the numerical (approximate) solution. By contrast, backward error analysis focusses on the backward error, which corresponds to the forward error being reflected back into a perturbation error in the initial conditions, and can thus be estimated more easily (see Fig. 3). In other words, within the backward perspective, the numerical solution is viewed as "the exact solution of a nearby problem"

(Corless and Fillion 2013, 52). Accordingly, the approach aims to answer the question: "*how much error in the input would be required to explain all output error?*" (Fillion and Corless 2014, 1462).
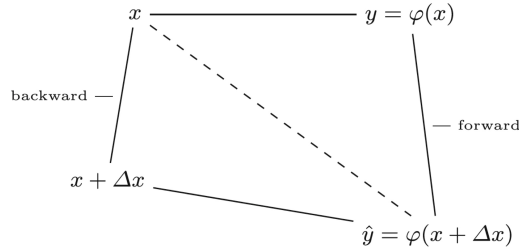

Figure 3: Backward error and forward error. (Fillion and Corless 2014, 1462).

Conditioning is the other important notion in this approach. According to Corless and Fillion, "a problem is well-conditioned if small changes in the problem lead to small changes in the solution" (2013, 507). Thus, the *conditioning* of an initial value problem is related to the sensitive dependence of differential equations (see Sections 2.1 and 2.2): it is even "called 'stability' in the differential equations literature" (*Ibid.*).

In the approach, the backward error and the forward error are linked by a quantity termed the *condition number* κ. This number measures the sensitive dependence of the dynamical system in changes of its specification.[10] In practice, if one can estimate the condition number, one can also work out the relationship between the forward error and the backward error. This turns out to be possible, for example, using Matlab software (Corless and Fillion 2013, chap. 12). In the same way, the backward error can be evaluated *a posteriori* through the computation of the *residual*, a key quantity. In the case of dynamical systems, the residual corresponds to the error in the velocity of the trajectories (Corless and Fillion 2013, 519). One can thus in turn calculate the forward error, and evaluate how close to the exact solution the computed trajectories of a dynamical system are (e.g., see (Corless and Fillion 2013, 533).

While a detailed discussion of this approach is out of the scope of this paper, we should nonetheless mention that it comes with a number of advantages (see e.g. Moir 2010 for ordinary differential equations). In particular, it is a general approach which is not restricted to dynamical systems (Fillion and Corless 2014, 2019; Fillion and Moir 2018). But, despite its incentives, backward analysis, to our knowledge, is yet not widely used in practice. It is therefore not easy to

---

[10]Roughly speaking, the relative forward error ($\delta y = \Delta y/y$) is bounded by the product of condition number with the relative backward error ($\delta x = \Delta x/x$), so that:

$$\delta y \leq \kappa \cdot \delta x \tag{6}$$

A problem, or a dynamical system, is called *ill-conditioned* if $\kappa > 1$, and *well-conditioned* if $\kappa \leq 1$.

assess the exact extent to which backward analysis is and can be implemented to overcome numerical instability in dynamical systems. That said, in the scientific practice, besides these mentioned methods, other strategies are currently being developed by practitioners and are more often employed. We now turn to some of these strategies.

5 Dealing with the risk of numerical instability

There is a risk of numerical instability in non-linear mathematical models of dynamical systems that are being solved by computer. In some cases, like the simple pendulum, the instability can be easily identified because the exact behavior of the dynamical system is well known. But, more often, the exact solutions for dynamical systems are not known.[11] In non-linear mathematical models, even the mere detection of numerical instability is sometimes not an easy task. How then to deal with the risk of numerical instability?

It seems reasonable, given the risk of numerical instability, to ask scientists to gather evidence supporting the reliability of their non-linear models. However, the criteria for numerical stability, discussed in Sections 4.1 and 4.2 (i.e. zero-stability and absolute stability), are insufficient to guarantee *a priori* the absence of numerical instability. Backward analysis, presented in Section 4.3, constitutes a new and promising *a posteriori* method to overcome numerical instability, however. Besides the methods mentioned above, scientists have in practice also developed certain specific *strategies*. In this section, we will outline some of the strategies that scientists may adopt on a case-by-case basis, depending on the problem they are facing.

5.1 Developing new numerical methods

In order to avoid instability occurring in long numerical computations, a first important strategy is *to develop new numerical methods*. Several options are usually available. One option is to use variable discretization steps: instead of choosing a constant time step parameter h, one can choose a variable time step h_k with respect to computation step k; or one can choose a technique to distribute mesh points (e.g., see Söderlind 2006), such as spectral methods for partial differential equations (e.g., Trefethen 1994, chap. 7). We also mention the defect-controlled methods, which are another dynamic approach to managing numerical stability (see, e.g., Corless 1992).[12]

A second option, on which we will focus in the remainder of this section, is to turn to *geometric integration* and *symplectic methods*. Such methods have been proven to be efficient for long-run computations, and for some other kinds

---

[11]There are, however, cases where the use of numerical solutions is preferred although scientists know the exact solutions (Ardourel and Jebeile 2017).

[12]We are in debt to an anonymous reviewer for providing such insights.

of problems, and in particular for dynamical systems.[13] This is a lively topic within the field of numerical analysis. For example, although the two first textbooks by Hairer et al. are dedicated to numerical methods for non-stiff and stiff problems, the third is entirely devoted to *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. The authors emphasize the importance and the specific character of these methods:

> In the last few decades, the theory of numerical methods for general (non-stiff and stiff) ordinary differential equations has reached a certain maturity, and excellent general-purpose codes, mainly based on Runge–Kutta methods or linear multistep methods, have become available. [...] It turned out that the preservation of geometric properties of the flow not only produces an improved qualitative behaviour, but also allows for a more accurate long-time integration than with general-purpose methods.

> An important shift of view-point came about by ceasing to concentrate on the numerical approximation of a single solution trajectory and instead to consider a numerical method as a *discrete dynamical system* which approximates the flow of the differential equation – and so the geometry of phase space comes back again through the window. (Hairer, Lubich, and Wanner 2002, v)

The main idea of this approach is the following: for some kinds of dynamical systems, such as Hamiltonian systems, numerical methods, which preserve – in a discrete sense – some invariants, e.g. momenta or symplectic forms, will lead to more stable numerical resolutions.

Let us take an example discussed by Hairer et al. (2002). This is the numerical resolution of the equations of a reduced solar system with the five outer planets relative to the Sun. Different numerical resolutions performed over a time period of 200,000 days are carried out with four numerical methods (Figure 4). Unlike the two first methods, the last two methods are symplectic numerical methods and lead to very stable long-run numerical computations. Moreover, according to Hairer et al. (2002):

> An integration over a much longer time of say several million years does not deteriorate this behaviour. (Hairer et al. 2002, 15)

This long-time numerical stability is explained by the preservation of the symplectic form of the Hamiltonian system when the symplectic Euler method or the Störmer–Verlet scheme are used (see Ardourel and Barberousse (2017)). This strategy can, in principle, be applied to any Hamiltonian system. For example, we have seen previously that the numerical resolution of the simple pendulum is not stable with the backward Euler method (see Figure 2). By contrast, Hairer et al. show that the numerical resolutions become stable if a symplectic Euler method is used (2002, 5; see also Stern and Desbrun (2008)).

---

[13]Having recourse to adaptive time steps is compatible with resorting to geometric integration (e.g., see Ardourel and Barberousse 2017, 180).
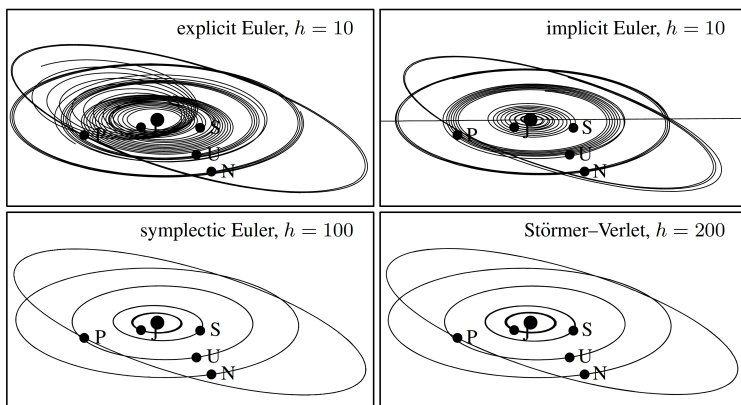
Figure 4: The trajectories of the planets of a reduced solar system computed with four numerical methods: forward Euler scheme, backward Euler scheme, symplectic Euler scheme, and Störmer–Verlet scheme. A 10-days time step is used with the forward and backward Euler integrators while a 100-days step is used with the symplectic Euler and a 200-days step with the Störmer–Verlet scheme. For each numerical computation, the initial conditions correspond to the positions of planets on September 5, 1994 at 0h00, and numerical resolutions are performed over a time period of 200,000 days (Hairer et al. 2002, 14).

5.2   Tuning, parameterizing and tinkering

Another approach is to consider models as autonomous mediators standing between experimentation and theoretical activity. Understood in this sense, models are not entirely grounded on theories but should also be shaped by data. Hence, matching the data becomes as high a priority as solving the equations in the most exact way, and may even be of higher priority, as claimed by Lenhard.

Lenhard and his co-author (Lenhard 2007; Lenhard and Küster 2019) highlight that the discretization process is a part of simulation modeling. The steps of simulation modeling encompass the design of an initial continuous mathematical model, the transformation of this model into a discrete version by a discretization process, the implementation of this discrete version on computer, the simulation, and finally the evaluation, which can require iterative loops of the transformation of the continuous model into the discrete version (see Lenhard and Küster 2019, Fig. 2). Each step – the different operations and the resulting transformations – aims to fix specific possible problems at the different stages of the model build.

Importantly, in his 2007, Lenhard argues that the discretization process, which he considers to be an important task in scientific modeling, is not just a mathematical problem concerning the resolution of the equations and the obtaining of acceptable approximate solutions to the initial equations. For him, the discretization process should also be oriented toward fitting the data. In

this context, fitting the data requires tuning the model parameters accordingly, making adequate parameterizations, and/or choosing an appropriate numerical scheme.

In other terms, the results of the discrete version should not be taken strictly as the "numerical solutions" to the continuous version, or "in any case, [they] correspond to these with an arbitrary precision" (Lenhard 2007, 189). On the contrary, taking the example of Arakawa's simulation of the atmosphere, he claims that "imitating the phenomena of the atmosphere is more important than solving the primitive equations" (Lenhard 2007, 184).

In this respect, the problem of instability has a different degree of epistemological difficulty: "The problem of the instability of long-term integrations could only be overcome when the construction of a discrete simulation model was seen as a task in its own right, oriented more at the imitation of phenomena than at solving the equations of the theoretical model" (Lenhard 2007, 177). As Lenhard highlights, Arakawa in his approach to the atmosphere makes an assumption that is inconsistent with our best knowledge in physics, i.e. he posits the conservation of kinetic energy in the atmosphere. Yet by doing so he is able to overcome numerical instability. "In terms of physics, one can say that Arakawa artificially limited the growth of instability by conserving kinetic energy" (Lenhard 2007, 185). In the same vein, Hollingsworth, Kållberg, Renner, and Burridge (1983) suggest a modification of the numerical scheme in order to overcome a problem of instability due to linear terms of definite difference equations not conserving momentum.

As Lenhard finally concludes:

> The key point is that the consequences of a model, its properties, can be observed in experiments, and through this, by implementing different discretization schemes or by introducing, tuning, and adjusting parameters, one can obtain, in turn, a further change in the model that is driven directly by the phenomena or the data. The model becomes the object of a recursive process that runs repeatedly through numerical experiments, observations, and parameter adjustments. (Lenhard 2007, 191)

We want to highlight that tinkering with the numerical scheme, i.e. developing ad hoc solutions, can indeed be a strategy to overcome numerical instability. This is the case when dealing with non-linear instability and the aliasing effect (see Section 3). Solutions to aliasing due to nonlinear terms have been developed to eliminate aliasing errors and are available to scientists. Examples include filtering wave components whose wavelengths are less than four grid intervals, as suggested by Phillips (1959) himself but also by Platzman (1961) and Baer (1961); filtering out all waves whose wavelengths are between $2\Delta x$ and $3\Delta x$ as shown by Orszag (1971); and using a different difference scheme, e.g. a quadratic conserving scheme or a staggered-grids-based scheme (cf. Arakawa 1977, 48-51).

We agree with Lenhard that matching the data, possibly via tuning the model parameters, parameterizations, and tinkering with the numerical scheme,

is an important aspect of dealing with numerical instability. That said, we are also aware that there is a risk of overfitting the data, which makes the model incapable of providing genuine novel predictions outside the range of the benchmark data. We believe that more work needs to be done to explore the tension between the capacity of the tuning of model parameters to fix numerical instability, and the risk of overfitting the data.[14]

6 Conclusion

In philosophical studies regarding mathematical models of dynamical systems, instability due to sensitive dependence on initial conditions, on the one side, and instability due to sensitive dependence on model structure, on the other, have by now been extensively discussed. Yet there is a third kind of instability, which by contrast has so far been overlooked, which also poses challenges for model predictions about dynamical systems: this is *numerical instability* due to numerical methods involving a discretization process, where discretization is required to solve the equations of dynamical systems on a computer.

After providing some definitions, our discussion in this paper turned to the question of how scientists can know that their models are numerically (un)stable. To this end, we first considered the criteria for numerical *stability*, as usually provided by numerical analysis textbooks, and we argued that they are insufficient. After having briefly presented backward analysis, an approach which shows promise, we then outlined practical strategies that scientists might adopt to avoid numerical instability in practice.

We hope this contribution might encourage further investigation by philosophers of this important kind of instability, rooted in applied mathematics, which has so far been rather overlooked compared to the Butterfly effect and the Hawkmoth effect.

7 Appendix

This Appendix provides examples and mathematical details on some notions discussed in the paper, viz. stiff problems, zero-stability and absolute stability.

7.1 Stiff problems

Here is an one-dimensional example of stiff problem provided by Hairer and Wanner (1991):[15]

$$\frac{dy}{dx} = -50(y - \cos x) \tag{7}$$

On the left of Figure 5 are the solution curves of the equation. They show a smooth solution, close to $y = \cos x$, as well as a bunch of solutions that reach

---

[14]For a recent attempt at showing how tuning the parameters can fix structural instability, see Baldissera Pacchetti (2020).

[15]They borrow this example from Curtiss and Hirschfelder (1952).

this smooth solution after a transient phase. However, some other numerical solutions do not behave the same way. On the right of Figure 5 are two solutions with initial value $y(0) = 1$, and respectively step sizes $h = 1.974/50$ and $1.875/50$. These solutions have no transient phase, and enable us to infer that, as soon as the step size is larger than $2/50$, the numerical solution goes too far beyond the equilibrium and violent oscillations occur (Hairer and Wanner 1991, 2). Specific numerical methods have been developed in order to solve stiff
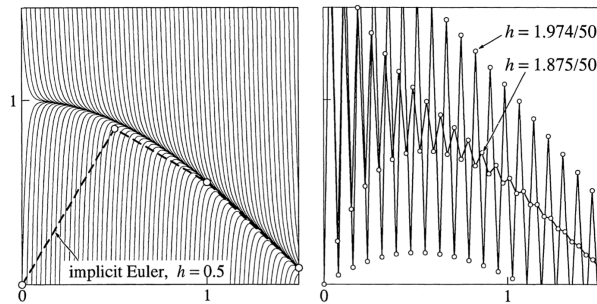

Figure 5: On the left, solution curves with implicit Euler solution. On the right, explicit Euler solution for $y(0) = 0$, $h = 1.974/50$ and $1.875/50$. (Hairer and Wanner 1991, 2)

problems (see Section 7.3).

7.2 Zero-stability

In order to understand the importance of a numerical method being zero-stable, let us take the example of a dynamical system defined by the following partial differential equation:

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \tag{8}$$

with $u(x, 0) = \cos^2 x$ if $|x| \leq \pi/2$, $u(x, 0) = 0$ otherwise.

Let us take the leap frog numerical method, viz. the discretization $u_j^{n+1} = u_j^{n-1} + \lambda(u_{j+1}^n - u_{j-1}^n)$, with the space step $h = 0.04\pi$ and time step $k = \lambda h$, where λ is the mesh ratio (see Trefethen 1994, 148). Figure 6 shows the two numerical solutions with respectively $\lambda = 0.9$ and $\lambda = 1.1$. As we can see, for $\lambda < 1$ there is a propagation of the wave as predicted by the theory. This is the regime where the numerical method is zero-stable. On the contrary, for $\lambda > 1$ numerical artefacts appear in the propagation of the wave. This is the regime where the numerical method is *not* zero-stable.

As emphasized by Trefethen (1994), in the case where $\lambda > 1$: "The errors introduced at each step are not much bigger than before [i.e. for $\lambda < 1$] but they grow exponentially in subsequent time steps [...]. This rapid blow up of a sawtooth mode is typical of unstable finite difference formulas" (1994, chap 4, 148).
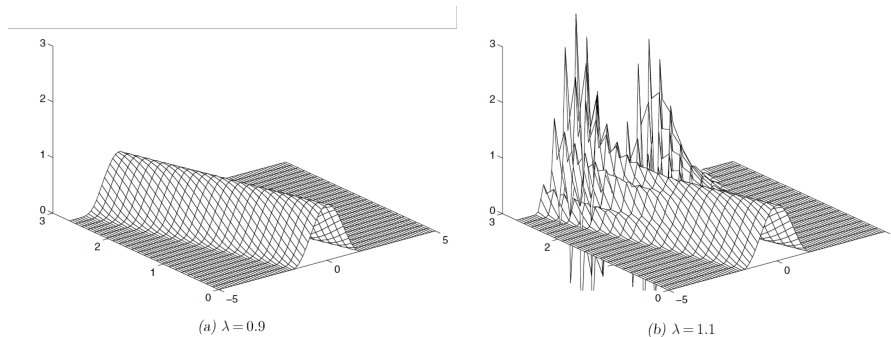
18

Figure 6: Stable and unstable leap frog approximations to $\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}$. (Trefethen 1994, 149, Fig. 4.1.1)

It is important to emphasize that the instability is not generated by round-off errors; it is generated by discretization errors which accumulate in the regime where the numerical method is not zero-stable. In both cases ($\lambda < 1$ and $\lambda > 1$) the numerical method is consistent. Nevertheless, the numerical solution to equation 8 would converge to the exact solution when h and k tend to zero in the regime of zero-stability, i.e. when $\lambda < 1$.[16] Thus, as we have just seen, in order to avoid numerical instability, it can be crucial to investigate whether a numerical method is zero-stable.

Zero-stability can be applied to numerical methods used for ordinary differential equations. In those cases, it concerns the behavior of the numerical method for a fixed time value t, as time step h tends to zero (see Hairer et al. 1987, 378, and Fasshauer 2007, chap. 4). This is a particularly important condition for multi-step methods (Süli and Mayers 2003, 332; Hairer et al. 1987, 398; Trefethen 1994, chap. 1, 40). Zero-stability also concerns the numerical methods used for partial differential equations. In this case, it also concerns the behavior of the numerical method for a fixed time value t, as time step k tends to zero, but also as *spatial* step h tends to zero.

Consistency is a property required for a numerical method. A numerical method is said to be *consistent* if the local errors introduced at each step tend to zero when the discretization steps tend to zero. However, consistency is a necessary but not sufficient condition for solution *convergence*, which means that the discrete solution tends to the exact continuous one when the time step tends to zero. Zero-stability is also required: if a numerical method is consistent and zero-stable, then it is convergent. In other words, being zero-stable allows

---

[16]There are mathematical results that state the conditions for which a finite difference formula can be zero-stable. This is for instance the case of the Courant-Friedrichs-Lewy (CFL) condition. Informally, the CFL condition states that a finite difference formula can be stable only if its numerical domain of dependence is at least as large as the mathematical domain of dependence (Trefethen 1994, 162).

a numerical method to be convergent.[17]

7.3 Absolute stability and stiffness

In order to better understand what absolute stability is, and how it is connected with stiff problems, let us take the example of the initial value problem:

$$\frac{dx}{dt} = -\mu x \quad x(0) = a \tag{9}$$

The solution is an exponential function such as $x(t) = ae^{-\mu t}$ with $\mu > 0$.

The resolution of the initial value problem with the *Euler forward* method leads to the following discrete equation:

$$x_{k+1} = x_k - h\mu x_k, \quad x_0 = a \tag{10}$$

The exact discrete solution is: $x_k = a(1 - \mu h)^k$. If $h < \frac{2}{\mu}$, both continuous and discrete solutions tend to zero when t or k tends to infinity. Nevertheless, when $h > \frac{2}{\mu}$, the discrete solution increases with k. This illustrates that the Euler method is not absolutely stable for all values of h. It is a *conditionally stable* numerical method.

Let us now use the *Euler backward* method in order to solve this same problem. In that case, the discrete equation becomes:

$$x_{k+1} = x_k - h\mu x_{k+1}, \quad x_0 = a \tag{11}$$

The exact discrete solution is: $x_k = x_0/(1 + \mu h)^k$. Here, both continuous and discrete equations tend to zero when t and k tend to infinity, whatever the value of h is. This illustrates that the Euler backward method is an *unconditionally stable* numerical method.

The Euler forward method is an *explicit* method: one just needs to know the value x_k to directly infer the value x_{k+1} at the next time step by applying the discrete time evolution. On the contrary, the Euler backward method is *implicit*: one cannot directly derive the value of x_{k+1} with x_k, instead one generally has to solve a non-linear equation with x_{k+1}. Generally, explicit methods are not suitable to solve stiff problems – something which is sometimes part of the definition of stiffness itself! Thus, Hairer et al. (1987) assert that "stiff equations are problems for which explicit methods don't work" (1987, 2). Implicit methods, on the contrary, are indeed generally suitable.[18]

---

[17]In the context of partial differential equations, this is linked to the main theorem of numerical analysis, called the *Lax Equivalence Theorem*. Informally, the Lax Equivalence Theorem says that, as soon as the numerical scheme is consistent and based on a *linear* finite difference model, stability is a necessary and sufficient condition for convergence (Trefethen 1994, 157). The notion of convergence is connected to *consistency* (e.g. Trefethen 1994, chap. 1, 53) but also to *conditionning* (e.g. Corless and Fillion 2013, 598; see Section 4.3).

[18]There are mathematical results that state under which conditions a linear multistep numerical method is absolute stable (see Fasshauer 2007, chap. 4, Hairer et al. 1987). The notion of *L-stability* can be introduced to further elaborate the idea of absolute stability (Bui and Bui 1970, 356).

References

Arakawa, A. (1977). Computational aspects of numerical models for weather prediction and climate simulation. *Methods in Computational Physics: Advances in Research and Applications*, *17*.

Ardourel, V., & Barberousse, A. (2017). The representation of time in discrete mechanics. In C. Bouton & P. Huneman (Eds.), *Time of nature, nature of time. philosophical perspectives of time in natural sciences* (Vol. 326, p. 173-208). Boston Studies in the Philosophy and History of Science, Dordrecht: Springer.

Ardourel, V., & Jebeile, J. (2017). On the presumed superiority of analytical solutions over numerical methods. *European Journal for Philosophy of Science*, *7*(201-220).

Baer, F. (1961). The extended numerical integration of a simple barotropic model. *Journal of Meteorology*, *18*, 319-339.

Baldissera Pacchetti, M. (2020). Structural uncertainty through the lens of model building. *Synthese*, *forthcoming*.

Batterman, R. W. (1993). Defining chaos. *Philosophy of Science*, *60*(1), 43-66.

Bradley, S., Frigg, R., Du, H., & Smith, L. A. (2014). Model error and ensemble forecasting: A cautionary tale. In C. G. Guichun & L. Chuang (Eds.), *Scientific explanation and methodology of science* (p. 58-66). Singapore: World Scientific.

Bréhier, C.-E. (n.d.). *Introduction to numerical methods for ordinary differential equations.* (Lecture notes. `http://mmi-lyon.fr/wp-content/uploads/2017/03/brehier_numerical_ode.pdf`)

Bui, T. D., & Bui, T. R. (1970, October). Numerical methods for extremely stiff systems of ordinary differential equations. *Appl. Math. Modelling*, *3*, 355-358.

Corless, R. M. (1992). Defect-controlled numerical methods and. shadowing for chaotic differential equations. *Physica D: Nonlinear Phenomena*, *60*(1-4), 323-334.

Corless, R. M., & Fillion, N. (2013). *A graduate introduction to numerical methods. from the viewpoint of backward error analysis.* Springer.

Corless, R. M., & Fillion, N. (2019). Backward error analysis for perturbation methods. In N. Fillion, R. M. Corless, & I. S. Kotsireas (Eds.), (p. 35-79). Proceedings of 2015 and 2016 ACMES conferences, Fields Institute Communications, vol. 82, New York: Springer.

Curtiss, C., & Hirschfelder, J. (1952). Integration of stiff equations. *Proceedings of the National Academy of Sciences*, *38*(235-243).

Fasshauer, G. (2007). *Numerical methods for differential equations/computational mathematics ii (spring 07). handouts and worksheets.* *http://www.math.iit.edu/~fass/478_578_handouts.html.*

Fillion, N., & Bangu, S. (2015). Numerical methods, complexity and epistemic hierarchies. *Philosophy of Science*.

Fillion, N., & Corless, R. (2014). On the epistemological analysis of modeling and computational error in the mathematical sciences. *Synthese*, *191*,

1451-1467.

Fillion, N., & Corless, R. M. (2019). Concepts of solution and the finite element method: a philosophical take on variational crimes. *Philosophy & Technology*.

Fillion, N., Corless, R. M., & Kotsireas, I. S. (Eds.). (2019). *Algorithms and complexity in mathematics, epistemology, and science*. Proceedings of 2015 and 2016 ACMES conferences, Fields Institute Communications, vol. 82, New York: Springer.

Fillion, N., & Moir, R. H. C. (2018). Explanation and abstraction from a backward-error analytic perspective. *European Journal for Philosophy of Science*, *8*, 735-759.

Frigg, R., Bradley, S., Du, H., & Smith, L. A. (2014). Laplace's demon and the adventures of his apprentices. *Philosophy of Science*, *81*, 31-59.

Frigg, R., Smith, L. A., & Stainforth, D. A. (2015). An assessment of the foundational assumptions in high-resolution climate projections: the case of ukcp09. *Synthese*, *192*, 3979–4008.

Hairer, E., Lubich, C., & Wanner, G. (2002). *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations*. Springer Series in Computational Mathematics 31, first ed. 2002, second edition 2006.

Hairer, E., Nørsett, S., & Wanner, G. (1987). *Solving ordinary differential equations i : Nonstiff problems*. Springer Series in Computational Mathematics 8, second ed. 1993, third ed. 2008.

Hairer, E., & Wanner, G. (1991). *Solving ordinary differential equations ii : Stiff and differential-algebraic problems*. Springer Series in Computational Mathematics 14, first ed. 1991, revised edition 2010.

Higham, N. J. (1996). *Accuracy and stability of numerical algorithms*. Philadelphia: Siam, Society for Industrial and Applied Mathematics.

Hollingsworth, A., Kållberg, P., Renner, V., & Burridge, D. M. (1983). An internal symmetric computational instability. *Quaterly journal of the Royal Meteorological Society*, *109*, 417-428.

Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*. New-York : Oxford University Press.

Kellert, S. H. (1993). *In the wake of chaos*. Chicago: University of Chicago Press.

Krivine, H., Lesne, A., & Treiner, J. (2007). Discrete-time and continuous-time modelling: some bridges and gaps. *Math. Struct. in Comp. Science*, *17*, 1-16.

Lenhard, J. (2007). Computer simulation: The cooperation between experimenting and modeling. *Philosophy of Science*, *74*(2), 176-194.

Lenhard, J. (2019). *Calculated surprises: A philosophy of computer simulation*. Oxford Studies in Philosophy of Science.

Lenhard, J., & Küster, U. (2019). Reproducibility and the concept of numerical solution. *Minds and Machines*.

Mayo-Wilson, C. (2015). Structural chaos. *Philosophy of Science*, *82*, 1236–1247.

Moir, R. H. C. (2010). *Reconsidering backward error analysis for ordinary differential equations* (Unpublished master's thesis). Faculty of Science, Department of Applied Mathematics, The University of Western Ontario, London, Ontario, Canada.

Moir, R. H. C. (2013). *Structures in real theory application: A study in feasible epistemology* (Unpublished doctoral dissertation). Faculty of Science, Department of Applied Mathematics, The University of Western Ontario, London, Ontario, Canada, `https://ir.lib.uwo.ca/etd/1578`.

Moir, R. H. C. (2019). Effective validity: a generalized logic for stable approximate inference. In N. Fillion, R. M. Corless, & I. S. Kotsireas (Eds.), (p. 225-268). Proceedings of 2015 and 2016 ACMES conferences, Fields Institute Communications, vol. 82, New York: Springer.

Nabergall, L., Navas, A., & Winsberg, E. (2019). An antidote for hawkmoths: on the prevalence of structural chaos in non-linear modeling. *European Journal for Philosophy of Science*, *9*, 21.

Orszag, S. A. (1971). On the elimination of aliasing in finite difference schemes by filtering high-wavenumber components. *Journal of Atmospheric Science*, *28*, 1074.

Palais, R. S., & Palais, R. A. (2009). *Differential equations, mechanics, and computation.* Student Mathematical Library, 51. IAS/Park City Mathematical Subseries. American Mathematical Society, Providence, RI; Institute for Advanced Study (IAS), Princeton, NJ.

Phillips, N. A. (1959). An example of non-linear computational instability. In B. Bolin (Ed.), *The atmosphere and the sea in motion* (p. 501-504). Oxford University Press.

Platzman, G. W. (1961). An approximation to the product of discrete functions. *Journal of Meteorology*, *18*, 31-37.

Smith, P. (1998). *Explaining chaos.* Cambridge: Cambridge University Press.

Söderlind, G. (2006). Time-step selection algorithms: Adaptivity, control, and signal processing. *Applied Numerical Mathematics*, *56*(3), 488-502.

Stern, A., & Desbrun, M. (2008). Discrete geometric mechanics for variational time integrators. In *Discrete differential geometry : An applied introduction* (chap. 15). Siggraph Course Note.

Süli, E., & Mayers, D. (2003). *An introduction to numerical analysis.* Cambridge: Cambridge University Press.

Trefethen, L. N. (1994). *Finite difference and spectral methods for ordinary and partial differential equations.* (available at http://people.maths.ox.ac.uk/trefethen/pdetext.html)

Werndl, C. (2009). What are the new implications of chaos for unpredictability? *British Journal for the Philosophy of Science*, *60*(1), 195–220.

Wilkinson, J. H. (1971, October). Modern error analysis. *SIAM Review*, *13*(4), 548-568.

Winsberg, E. (2018). *Philosophy and climate science.* Cambridge: Cambridge University Press.

Winsberg, E., & Goodwin, W. M. (2016). The adventures of climate science in the sweet land of idle arguments. *Studies in History and Philosophy of*

Modern Physics, *54*, 9–17.