On the foundations of computational mathematics

B.J.C. Baxter & A. Iserles

Contents

"...there is no permanent place in the world for ugly mathematics."

G.H. Hardy, [33]

1 On the place of numerical analysis in the mathematical universe

A long time ago, when younger and rasher mathematicians, we both momentarily harboured the ambition that one day, older and wiser, we might write a multivolume treatise titled "On the Mathematical Foundations of Numerical Analysis". And then it dawned that such a creation already exists: it is called "a mathematics library". Indeed, it is almost impossible to identify a mathematical theme, no matter how 'pure', that has never influenced numerical reasoning: analysis of all kinds, of course, but also algebra, graph theory, number theory, probability theory, differential geometry, combinatorial topology, category theory.... The mainspring of numerical analysis is, indeed, the entire aquifer of mathematics, pure and applied, and this is an enduring attraction of the discipline. Good luck to those content to spend their entire mathematical existence toiling in a narrow specialism; our choice is an activity that borrows eclectically, indeed with abandon, from all across the mathematical landscape.

This is emphatically *not* the popular mathematical perception of numerical analysis. The conventional view is captured in a memorable footnote of [38], bravely defining what numerical analysis was *circa* 1945 (from the standpoint of a pure mathematician) and then adding "As a branch of mathematics, however, numerical analysis probably ranked the lowest, even below the theory of statistics, in terms of what most university mathematicians found interesting". (To which many contemporary numerical analysts will add "so, not much has changed".) Numerical analysis as a subject has had a bad name and it makes sense to tackle this perception head on, before any further elaboration of the discipline, its future and its interaction with other areas of scholarly endeavour.

Numerical analysis lies at the meeting point of pure mathematics, computer sciences and application areas. It often attracts some degree of hostility from all three. Yet it is important to note that the complaints of pure mathematicians, computer scientists and the broad constituency of scientists and engineers are very different and often contradictory.

To pure mathematicians, numerical analysis fails the Hardy test: it is (supposedly) ugly. The beauty of mathematics, described so lovingly by Hardy [33], is in the quest for pattern and rigour: precise definition, well-formulated theorems with perfect proofs.... Falling back on numerical calculation is, almost axiomatically, the moment mathematics fails: the end of rigour and precision, the abandonment of traditional theorem-and-proof, merely tedious discretization and number crunching. This is nonsense. A mathematical problem does not cease to be a *mathematical* problem once we resort to discretization: in principle, we replace a differential equation with a difference equation. The mathematical problem is, if anything, more demanding and intricate. Numerical formulation allows a computer – a mindless contraption – to reproduce strings of numbers and figures that might (or might not) bear some resemblance to the true solution of the underlying problem. The work of a theoretical numerical analyst is precisely to verify the provenance and precision of the above numbers and figures, using exactly the same rules of mathematical engagement as, say, a topologist or an algebraic geometer.

The opposition of computer scientists and of the applications' community is different in kind. Computer scientists are historically wedded to the discrete and display disregard, and even disdain, for computations involving real numbers or floating point arithmetic [56]; it is particularly illuminating that `C`, the *lingua franca* of programming languages, was not originally provided with complex numbers. Applied mathematicians, scientists and engineers dislike numerical analysis because they view numerical computation as a necessary evil, distracting them from their true focus . For example, suppose that you are an applied mathematician. Your aim is to further the understanding of a natural or technological phenomenon. You have obtained experimental data, formulated a mathematical model (almost certainly in terms of partial differential equations), analysed it with perturbation techniques, ever dreading the moment of that awful trigraph "DNS": direct numerical simulation. For, sooner or later, you will be forced to spend weeks of your (or your graduate students') time programming, debugging, and fine-tuning algorithms, rather than absorbing yourself in your genuine area of interest.

The focus of this essay is on the interaction of numerical with pure mathematics.

Firstly, we will dispense briefly with the other protagonists, not because they are less important or less interesting but because they are a matter for another discourse. As far as computer scientists are concerned, their genuine 'interface of interaction' with numerical thinking is in two distinct areas: complexity theory and high-performance computing. While complexity theory has always been a glamourous activity in theoretical computer science, it has only recently emerged as a focus of concerted activity in numerical circles [5, 70], occasionally leading to a measure of acrimony [51, 64]. It is to be hoped that, sooner or later, computer scientists will find complexity issues involving real-number computations to be challenging, worthwhile and central to the understanding of theoretical computation. Likewise, the ongoing development of parallel computer architectures and the computational grid is likely to lead to considerably better numerical/computational interaction at the more practical, engineering-oriented end. The applied flank of numerical analysis is likely to be secured by the emergence of *scientific computing,* also known as *computational science:* the subject of scientific endeavour centred on computation and on a computational model as an alternative to experiment and expensive testing [73]. This is bound to lead to the emergence of a common agenda for scientists, engineers and numerical analysts.

Returning to the interaction with pure mathematics, we note that the latter was never as 'pure' and as detached from the 'real world' as its most fundamentalist adherents, or their fiercest critics, might choose to believe; cf. N. Young's careful deconstruction in [76]. It is difficult to imagine the invention of calculus, say, without the intertwined interests in navigation and astronomy. To move the time frame to the more recent past, imagine differential geometry, functional analysis and algebraic topology without the influence of relativity or quantum theory, or probability theory without statistics. We are in the midst of a new industrial revolution, despite the ending of the 'dot-com' boom. In place of mechanical contraptions machining physical goods, the 'new economy' is increasingly concerned with information, its derivation, transmission, storage and analysis. The computer is the information-processing engine of this emergence (the French word *ordinateur* is much more appropriate) and its influence on the themes and priorities of mathematics will be comparable to the role of physics in earlier centuries.

Note the absence of "numerical analysis" in the last paragraph. This was deliberate. The joint border of mathematics and computing contains considerably more than numerical analysis alone (or, indeed, computer science) and grows daily. It is crucial to recognise that there is a two-way traffic across the border. On the one hand, pure mathematicians increasingly realise that many of their problems can be elucidated by computation: numerical analysis, numerical algebra, computational number theory, computational topology, computational dynamics, symbolic algebra and analysis.... Arguably, the day is nigh when every mathematical discipline will have its computational counterpart, a convenient means to play with ideas and investigate mathematical phenomena in a 'laboratory' setting. The idea is most certainly *not* to dispense with mathematical proof and rigour, but to use the computer to help with the necessary guesswork that leads to clear formulation and proof. On the other hand, seeing mathematical issues through the prism of computation inevitably leads to new *mathematical* problems, in need of traditional mathematical treatment. As we have already mentioned, such problems are often considerably more intricate than the mathematical phenomena that they approximate: discretizations in place of differential equations, iterative schemes in place of algebraic equations, maps in place of flows.

It is misleading to appropriate the name 'numerical analysis' for this activity. Firstly, we have already mentioned that the interaction of mathematics and computing is much wider and richer. Secondly, not all numerical analysts are concerned with mathematical issues: the design, programming and implementation of algorithms, i.e. their software engineering, are just as important and, arguably, more crucial for future applications than theorem proving. One should never confuse mathematics (or beauty, for that matter) with virtue. Thus, it makes sense to resurrect the old designation of *computational mathematics* to embrace the range of activities at the mathematics/computation interface. From this moment onward we will thus often abandon the dreaded words "numerical analysis" altogether and explore the foundations of computational mathematics.

Mathematics is not some sort of an optional extra, bolted onto computational algorithms to allow us the satisfaction of sharing in the glory of Newton, Euler and Gauss. Mathematical understanding is ultimately crucial to the design of more powerful and useful algorithms. Indeed, part of the enduring profundity and beauty of mathematics is how often 'useless' mathematical concepts and constructs turn out to be very useful indeed. Hardy was right: mathematics reveals the structure and the underlying pattern of a problem. This is just as true with regard to mathematical computation as, say, in regard to algebraic geometry or analytic number theory.

2 The 'mathematics' in computational mathematics

The minimalist definition of 'the mathematics behind computational mathematics' is little more than linear algebra and Taylor expansions. The maximalist definition is that all mathematics is either relevant or potentially relevant to computation. The first definition is wrong but the second, while arguably correct, is neither useful nor informative. We need to refocus and explore a number of broad mathematical themes in greater detail, as they pertain to computation but always *from a mathematical standpoint*. This exercise, by necessity, is tentative, inexact, incomplete and coloured by our ignorance and prejudices. It lays no claims for any universal truth, being simply an attempt to foment debate.

2.1 Approximation theory

Suppose that we investigate a mathematical construct A, say, which, while probably amenable to some sort of mathematical analysis, is not amenable to computation. In the computational context it often makes sense to replace it by a different construct, A_ϵ, say, where ϵ denotes one or more built-in parameters. The virtue of A_ϵ is that it can be computed, but, to belabour the obvious, it is distinct from the original problem A. Thus some obvious pertinent questions follow: How closely does A_ϵ approximate A? Can we estimate, given ϵ, the error incurred by replacing A with A_ϵ? What is an appropriate measure for the error? How are the parameters and the error related?

These are classical approximation problems in numerical analysis and, whether in university courses or the research community, there is no simple dividing line between "proper" numerical analysis and approximation theory. We all take it for granted that the basics of approximation theory should be acquired by any budding numerical analysis, and these fundamentals are usually taken to include interpolation, least-squares and Chebyshev approximation by polynomials, some orthogonal-polynomial theory, elements of the theory of splines, and possibly simple rational approximation. These topics are still important, but it is time to add to the list.

In the last two decades, approximation theory has undergone a revolution on several different fronts. In the long run, this revolution underscores, indeed deepens, its relevance to computational mathematics. The most significant developments are probably new approximating tools, nonlinear approximation and multivariate approximation.

The most familiar of these emerging approximation tools are *box splines* [23], *radial basis functions* [11] and, perhaps the best known and understood, *wavelets* [22]. Each is based on a different premise and each confers a different advantage. It should also be noted that both box splines and radial basis functions are based on the fundamental work of I. J. Schoenberg.

The box spline M_X associated with the $d \times n$ matrix X with columns in $\mathbb{R}^d \setminus \{\mathbf{0}\}$ is the distribution defined by

$$\langle M_X, \phi \rangle := \int_{[0,1)^n} \phi(Xt) \, \mathrm{d}t,$$

where $\phi : \mathbb{R}^d \to \mathbb{R}$ can be any compactly supported smooth function. When $n \geq d$ and X is of full rank, then M_X can be identified with a function on the image of X. For example, when $X = (\, 1 \quad 1 \,)$, we obtain

$$\begin{aligned} \langle M_X, \phi \rangle &= \int_{[0,1)^2} \phi(t_1 + t_2) \, \mathrm{d}t \\ &= \int_{\mathbb{R}} M_X(x) \phi(x) \, \mathrm{d}x, \end{aligned}$$

where, slightly abusing notation, $M_X : \mathbb{R} \to \mathbb{R}$ also denotes the continuous function, supported by the interval $[0, 2]$, given by

$$M_X(x) = 1 - |1 - x|, \qquad x \in [0, 2].$$

In this case, we have obtained a *univariate B-spline*, and, in fact, all B-splines arise in this way. However, when X is *not* a $1 \times n$ matrix, we obtain the greater generality of box splines. Such functions are piecewise polynomial, although this is not completely obvious from the geometric definition given here. Their mathematical properties are beautiful, but, until relatively recently, they were solutions looking for problems. However, their fundamental relevance to subdivision, a rapidly developing field of computer-aided geometric design, is now evident.

How did such objects arise? The characterization of B-splines as volumes of slices of polyhedra was discovered by Curry and Schoenberg [20], but their generalization, and the beautiful mathematics thereby generated, did not arise until de Boor's work in the late 1970s. In the 1980s, they became a central topic of interest in approximation theory. Micchelli, one of the most active mathematicians in this field, provided a fascinating historical survey in Chapter 4 of his lecture notes on computer-aided geometric design (CAGD) [48].

The development of radial basis functions has been less coherent than that of box splines. It is well known that *natural spline* interpolants provide univariate interpolants minimizing a certain integral, and this variational characterization was the origin of the term "spline": the integral for cubic splines is an approximation to the bending energy of a thin rod, so that the "draughtsman's spline", an archaic tool for drawing smooth curves before CAGD, is approximated by cubic spline interpolants. Duchon [25] modified this variational characterization to construct *thin plate splines* (although these functions had previously been studied in [32], where they were termed "surface splines"). Duchon found that the function $s : \mathbb{R}^2 \to \mathbb{R}$ that interpolates the data $s(z_j) = f_j$, $j = 1, 2, \ldots, n$, at non-collinear distinct points $z_1, \ldots, z_n$ in $\mathbb{R}^2$, and minimizes the integral

$$\int_{\mathbb{R}^2} \left(s_{xx}^2 + 2s_{xy}^2 + s_{yy}^2\right) \, \mathrm{d}x\mathrm{d}y$$

is given by

$$s(z) = \sum_{k=1}^{n} c_k \phi(\|z - z_k\|) + p(z), \qquad z \in \mathbb{R}^2,$$

where $\phi(r) = r^2 \log r$, for $r \geq 0$ ($\phi(0) = 0$), p is a linear polynomial, and the coefficients $c_1, \ldots, c_n$ are defined by the interpolation equations and the moment conditions

$$\sum_{k=1}^{n} c_k = \sum_{k=1}^{n} c_k z_k = 0.$$

Thus $s - p$ is a linear combination of translates of the radially symmetric function $\Phi(z) = \phi(\|z\|)$, for $z \in \mathbb{R}^2$, and Duchon's analysis established the nonsingularity of the underlying linear system. This construction can be generalized to any dimension, and the possibly surprising functional form of Φ becomes more plausible once the reader knows that Φ is a multiple of the fundamental function for the biharmonic operator Δ^2 in $\mathbb{R}^2$.

Shortly before Duchon's analysis, a geostatistician, Rolland Hardy had constructed an approximant with a similar functional form [34]. He took

$$s(z) = \sum_{k=1}^{n} c_k \phi(\|z - z_k\|), \qquad z \in \mathbb{R}^d,$$

but chose $\phi(r) = (r^2 + c^2)^{1/2}$, calling the resultant surfaces *multiquadrics*. Several workers found Hardy's multiquadrics to be useful interpolants, but no theoretical basis was then available; it was not even known if the linear system defining the interpolant was invertible. Nevertheless, their practical performance was too compelling for them to be abandoned, and a careful survey of Richard Franke found thin plate splines and multiquadrics to be excellent methods for scattered data interpolation [27]. By this time, several variants had been suggested, so that Franke termed them "global basis function type methods". All of these methods used interpolants that were linear combinations of translates of a radially symmetric function, and the term "radial basis function" seems to have entered mathematical conversation shortly thereafter, although we cannot pinpoint its genesis.

Franke had conjectured that the multiquadric interpolation matrix A, defined by

$$A_{jk} = \phi(\|x_j - x_k\|), \qquad 1 \leq j, k \leq n,$$

for $\phi(r) = (r^2 + c^2)^{\pm 1/2}$, was nonsingular given distinct $x_1, \ldots, x_n \in \mathbb{R}^d$, whatever the dimension d of the ambient space. This conjecture was established by Charles Micchelli in his seminal paper [47], using mathematics originally developed by I. J. Schoenberg some forty years earlier, and this history is an excellent example of the path from pure mathematics to computational mathematics. In the 1930s, Fréchet, and others, had axiomatised metric spaces and were beginning to consider the following deep question: which metric spaces can be isometrically embedded in Hilbert spaces? This question was completely settled in a series of remarkable papers [53], [54], [55] and [67], and is too involved to present here in detail; a modern treatment may be found in [13]. To whet the reader's appetite, we remark that the interpolation matrix is positive definite, for any dimension d, if the function $f(t) := \phi(\sqrt{t})$ is *completely monotonic* (and non-constant), that is, $(-1)^m f^{(m)}(t) \geq 0$, for every $t > 0$ and each non-negative integer m (a new geometric proof of this fact may be found in [3]). This theory of positive definite functions on Hilbert space is equally fundamental to *learning theory*, where radial basis functions and data fitting are finding new applications.

One may ask what is so fundamental about box splines or radial basis functions? Our answer is based not on the specific definition of a box spline or of a radial basis function, but on the overall mathematical process underlying their construction. They are both a multivariate generalisation of the same familiar univariate construct, the spline function. Yet, they are not 'obvious' generalisations. After all, responding to a need for piecewise-polynomial approximation bases in a multivariate setting, in particular originating in the finite-element community, approximation theorists have been extending splines to $\mathbb{R}^m$, $m \geq 2$, for decades using tensor products. Yet, this procedure automatically restricts the underlying tessellation to parallelepipeds, and this imposes an unwelcome restriction on applications. An alternative, which has attracted impressive research effort, is an *ad hoc* construction of piecewise-smooth functions in more complicated geometries, one at a time. The twin approaches of box splines and of radial basis functions offer a multivariate generalisation of splines which allows an automatic generation of interpolating or approximating bases which are supported by fairly general polyhedra or, in the case of radial basis functions, which trade global support off for their ability to adjust to arbitrarily scattered data. The underlying trend is common to other foundational themes of computational mathematics: rather than extending existing constructs in a naive or *ad hoc* manner, dare to mix-and-match different mathematical concepts (in the present case, geometry and convexity theory, harmonic analysis, theory of distributions) to underly a more general computational construct.

2.2 Harmonic analysis

This is, of course, an enormous subject, whose *idée fixe* has been the construction of orthogonal decompositions of Hilbert spaces of square-integrable functions, but we should remark that there has again been a remarkable split between theoreticians and computational harmonic analysts. For example, the excellent books of Körner [42] and Terras [62] describe both the Fast Fourier Transform algorithm and the Poisson summation formula, but neither observes that the Danielson–Lanczos lemma, which is the recursion at the heart of every FFT code, is *precisely* the Poisson summation formula relating Fourier coefficients of functions defined on $\ell^2(\mathbb{Z}_{2n})$ to those defined on $\ell^2(\mathbb{Z}_n)$. As another example, there are many deep theoretical works describing the Radon transform from the perspective of a pure mathematician studying the harmonic analysis of homogeneous spaces [35], but almost none mention its fundamental importance to computational tomography [50]. Perhaps some *rapprochement* has occurred in the last decade with the emergence of wavelets and fast multipole methods, and it is to be hoped that this will flourish.

2.3 Functional analysis

For the traditionally minded, the phrase "the mathematics of numerical analysis" is virtually synonymous with functional analysis. Even bearing in mind the observation that computational mathematics is substantially wider than just numerical analysis, there is no escaping the centrality of functional analysis in any description of the foundations of computing.

While rightfully scornful of the "just discretise everything in sight and throw it on the nearest computer" approach, which sadly prevails among many practitioners of scientific computing, we should remember that this used to be the prevailing attitude of mathematicians with an interest in solving partial differential equations. The magic wand that turned mathematical wishful thinking into a coherent theory was functional analysis, and it was wielded in the main by research groups inspired by two towering individuals, Richard Courant in New York and Jacques-Louis Lions in Paris. The traditional view of the discretization procedure is that continuous function values are approximated by discrete values at grid points. Although amenable to mathematical analysis, this approach rests on the obvious infelicity of comparing unlike with unlike. The alternative paradigm approximates an *infinite-dimensional* function space by a *finite-dimensional* subspace. Thus, both the exact solution of a differential equation and its approximation inhibit the same space and the discretization error can be measured in the natural topology of the ambient function space.

A natural bridge between the infinite-dimensional and its approximation is provided by one of two complementary approaches. The *Ritz method* replaces a boundary-value differential problem by a variational problem, thereby traversing the "variational to differential" route, with a pedigree stretching to Newton, Euler and Lagrange, in the opposite direction. In a variational setting the original differential system reduces to global optimization in the underlying (infinite-dimensional) Hilbert space $\mathcal{H}$ and it is approximated by optimizing in a subspace $\mathcal{H}_m \subseteq \mathcal{H}$, where $\dim \mathcal{H}_m = m$. The norm $\|\cdot\|_{\mathcal{H}}$ provides the natural measuring rod to calculate the approximation error. The *Galerkin method* seeks a finite-dimensional approximation to the differential equation in the *weak* sense. Instead of searching for a function u such that $\mathcal{L}u = f$, where $\mathcal{L}$ is a differential operator and we impose suitable boundary conditions, we seek a function $u \in \mathcal{H}$ such that $\langle \mathcal{L}u - f, v \rangle_{\mathcal{H}} = 0$, $v \in \mathcal{H}_0$, where $\mathcal{H}$ is a Hilbert space of suitable functions (in practice, a *Sobolev space*) obeying the right boundary conditions, $\mathcal{H}_0$ is the same Hilbert space, except that boundary conditions are set to zero and $\langle \cdot\,, \cdot \rangle_{\mathcal{H}}$ is the underlying inner product. In a Galerkin method we seek a solution from a finite-dimensional subspace of $\mathcal{H}$.

All this is classical finite element theory, well understood and ably described in many monographs [16, 59]. Of course, this broad set of mathematical principles falls short of describing a viable computational approach. It needs to be supplemented with the practice of finite-element functions and their construction, the correct treatment of boundary conditions, and a plethora of mundane programming issues that often make the difference between a great mathematical idea of restricted practical importance and a great practical algorithm.

The role of functional analysis in the foundations of computational mathematics might thus be considered as the decisive, battle-winning weapon of a heroic, yet long past, age, the longbow of numerical analysis. This, however, is definitely false. Although we champion many new foundational themes and disciplines in this essay, functional analysis remains at the very heart of computational theory, and it is likely to stay there for a long time to come. The basic paradigm of functional analysis in the numerical setting, namely that approximation takes place in a finite-dimensional subspace of the infinite-dimensional space of all possible solutions, remains valid with regard to the many new methods for discretizing differential and integral operators: spectral and pseudospectral methods, boundary element and spectral element techniques, particle methods and, perhaps most intriguingly, multiresolution and multiscale techniques [21]. It is not an exaggeration to state that functional analysis provides the universal language and the rigour in the analysis of discretized partial differential equations.

An exciting recent development that brings together functional analysis, approximation theory and multiresolution techniques is *nonlinear approximation.* Most classical approximation problems, not least the problems from the previous subsection, are linear: we seek an approximating function from a linear space. As an example, consider the interpolation of univariate data by, say, cubic splines: we fix knots and interpolation points in the requisite interval and seek the unique member of the linear space of cubic splines with given knots that interpolates given data at the (fixed) interpolation points. This procedure and its error are well understood and, in particular, the practical solution of the problem can be reduced to the solution of a banded linear system of algebraic equations. However, suppose that we allow ourselves the freedom to vary the knots (and perhaps also the interpolation points) and are guided in our quest by the goal of minimising the approximation error in some norm. The problem ceases to be linear and it presents substantially more formidable challenge at both practical and analytic level. The choice of optimal knots and the degree of improvement in the quality of the approximation depend very sensitively upon the choice of the underlying function space. Yet, substantial progress has recently been made in nonlinear approximation, in particular in connection with multiresolution methods and wavelet functions [24, 61]. The underlying functional-analytic machinery is sophisticated, subtle and outside the scope of this brief essay. However, the importance of this development, like that of other important developments at the interface of mathematics and computation, is underscored by its impact on practical issues. "Free choice" of knots is nothing else but adaptivity, a central issue in real-life computing to which we will return in the next section.

Another fairly recent development, particularly relevant as numerical problems increase in size, is the phenonomenon of *concentration of measure*. This name refers to the increasing regularity seen in many structures as the dimension of the problem tends to infinity. One of the most striking ways to introduce the neophyte reader to this field is the following theorem of Lévy and Milman: a continuous function defined on the sphere $S^n = \{\boldsymbol{x} \in \mathbb{R}^{n+1} : \|\boldsymbol{x}\| = 1\}$ is, for large dimensions n, constant on almost all the sphere. More precisely, if our function f has median value M_f, and if A is that subset of points on the sphere at which f attains its median value, then

$$\mathbb{P}(\boldsymbol{x} \in A_\delta) \geq 1 - C \exp(-\delta^2 n/2),$$

where C is a (known) constant independent of n, $\mathbb{P}$ denotes normalized $(n-1)$-dimensional Lebesgue measure on the sphere, and

$$A_\delta := \{\boldsymbol{x} \in S^{n-1} : \rho(\boldsymbol{x}, A) \leq \delta\},$$

ρ denoting the *geodesic* metric on the sphere. This result seems to have been initially derived by Paul Lévy but "with a proof which has not been understood for a long time", according to Vitali Milman and Gideon Schechtman [49]. This beautiful result deserves to be better known, so we shall now sketch its derivation, further details being available in [49]. We shall need the *isoperimetric inequality* on the sphere, stated in the following form: for each $\eta \in (0,1)$ and $\delta > 0$, $\min\{\mathbb{P}(U_\delta) : U \subset S^n, \mathbb{P}(U) = \eta\}$ exists and is attained on a cap of suitable measure, that is, a set of the form $B(r) := \{\mathbf{x} \in S^n : \rho(\mathbf{x}, \mathbf{a}) \leq r\}$, where $\mathbf{a}$ can be any fixed point on the sphere. (To see the link with more familiar forms of the isoperimetric inequality, observe that $\mathbb{P}(U_\delta) \approx \mathbb{P}(U) + \ell(\partial U)\delta$, when the length $\ell(\partial U)$ of the boundary ∂U exists.) R.J. Gardner [28] has recently published a lucid derivation of this form of the isoperimetric inequality, based on the Brunn–Minkowski inequality. With the isoperimetric inequality to hand, we readily obtain the following crucial bound.

Theorem 1 *If $U \subset S^n$ is any measurable set satisfying $\mathbb{P}(U) \geq 1/2$, then*

$$\mathbb{P}(U_\delta) \geq 1 - C_1 \mathrm{e}^{-\delta^2 n/2}, \tag{2.1}$$

for all sufficiently large n, where C_1 is an absolute constant whose value is close to $1/2$.

Proof The isoperimetric inequality tells us that it is sufficient to establish (2.1) when U is a cap. Thus we must show that

$$\mathbb{P}(B(\frac{\pi}{2} + \delta)) \geq 1 - C_1 \mathrm{e}^{-\delta^2 n/2}.$$

Now

$$\mathbb{P}(B(\frac{\pi}{2} + \delta)) = \frac{\int_{-\pi/2}^{\delta} \cos^n \theta \, \mathrm{d}\theta}{\int_{-\pi/2}^{-\pi/2} \cos^n \theta \, \mathrm{d}\theta} = \frac{\int_{-\pi\sqrt{n}/2}^{\delta\sqrt{n}} \cos^n(\tau/\sqrt{n}) \, \mathrm{d}\tau}{\int_{-\pi\sqrt{n}/2}^{\pi\sqrt{n}/2} \cos^n(\tau/\sqrt{n}) \, \mathrm{d}\tau},$$

using the substitution $\theta = \tau/\sqrt{n}$. If we now introduce the function

$$f_n(\tau) = \begin{cases} \cos^n(\tau/\sqrt{n}), & |\tau| < \pi\sqrt{n}/2, \\ 0, & |\tau| \geq \pi\sqrt{n}/2, \end{cases}$$

we observe that

$$0 \leq f_n(\tau) \leq f(\tau) := \mathrm{e}^{-\tau^2/2} \qquad \text{for all } \tau \in \mathbb{R},$$

using the elementary inequality $\cos\sigma \leq \exp(-\sigma^2/2)$, for $|\sigma| < \pi/2$. Further, since $\lim_{n\to\infty} f_n(\tau) = f(\tau)$ at every point $\tau \in \mathbb{R}$, the dominated convergence theorem implies the relation

$$\lim_{n\to\infty} \int_{-\pi\sqrt{n}/2}^{\pi\sqrt{n}/2} \cos^n(\tau/\sqrt{n})\,\mathrm{d}\tau = \int_{\mathbb{R}} f(\tau)\,\mathrm{d}\tau = \sqrt{2\pi}.$$

Thus

$$1 - \mathbb{P}(B(\frac{\pi}{2} + \delta)) = \frac{\int_{\delta\sqrt{n}}^{\pi\sqrt{n}/2} \cos^n(\tau/\sqrt{n})\,\mathrm{d}\tau}{(1+o(1))\sqrt{2\pi}} \leq \frac{\int_{\delta\sqrt{n}}^{\infty} f(\tau)\,\mathrm{d}\tau}{(1+o(1))\sqrt{2\pi}},$$

and

$$\int_{\delta\sqrt{n}}^{\infty} f(\tau)\,\mathrm{d}\tau = \int_0^{\infty} e^{-(\sigma+\delta\sqrt{n})^2/2}\,\mathrm{d}\sigma \leq e^{-\delta^2 n/2} \int_0^{\infty} e^{-\sigma^2/2}\,\mathrm{d}\sigma,$$

whence

$$1 - \mathbb{P}(B(\frac{\pi}{2} + \delta)) \leq \frac{e^{-\delta^2 n/2}}{2(1+o(1))},$$

as $n \to \infty$. $\square$

Now let us define

$$A^+ := \{\mathbf{x} \in S^n : f(\mathbf{x}) \geq M_f\}$$

and

$$A^- := \{\mathbf{x} \in S^n : f(\mathbf{x}) \leq M_f\}.$$

By definition of the median, we have $\mathbb{P}(A^+) \geq 1/2$ and $\mathbb{P}(A^-) \geq 1/2$. Hence, applying our theorem, we deduce the inequalities

$$\mathbb{P}(A_\delta^\pm) \geq 1 - C_1 \mathrm{e}^{-\delta^2 n/2},$$

for all sufficiently large n. However, since $A = A^+ \cap A^-$ and $A_\delta = A_\delta^+ \cap A_\delta^-$, we deduce the relations

$$\mathbb{P}(A_\delta) = \mathbb{P}(A_\delta^+) + \mathbb{P}(A_\delta^-) - \mathbb{P}(A_\delta^+ \cup A_\delta^-) \geq \mathbb{P}(A_\delta^+) + \mathbb{P}(A_\delta^-) - 1 \geq 1 - 2C_1\mathrm{e}^{-\delta^2 n/2}.$$

Thus a continuous function defined on a large dimensional sphere is concentrated around its median value.

The concentration of measure phenomenon lies at the heart of much modern work on the geometry of Banach space, including Dvoretsky's famous theorem on almost Euclidean subspaces. For further work, we direct the reader to [49], and the delightful collection[43].

The importance we see here is that many numerical problems *benefit* from concentration of measure: the probability that we obtain an unpleasant example is small for large n. Random graph theory contains many examples of such properties, but we suspect that concentration of measure is also responsible for the paucity of unpleasant numerical examples in many fields when the dimension is large. For example, Edelman has computed the probability distribution of the condition number of symmetric matrices, and his results exhibit concentration of measure for large n. To take another example, the *field of values* of a symmetric $n \times n$ matrix A is precisely the values taken by the continuous function $f(\boldsymbol{x}) := \boldsymbol{x}^\top A\boldsymbol{x}$, when $\boldsymbol{x} \in S^{n-1}$. Thus we can immediately apply the Lévy–Milman theorem to deduce that the field of values will concentrate for large n.

2.4 Complexity theory

Classical complexity theory is at the very heart of the interface between computer sciences and mathematics. Its framework is based on the 'distance' between a problem and algorithms for its solution. Each algorithm bears a *cost,* which might be conveniently expressed in terms of computer operations, length of input and sometimes other attributes.[1] Minimizing the cost is an important (and often overwhelming) consideration in the choice of an algorithm and in the search for new algorithms. The *complexity* of a problem is the lowest bound on the cost of any algorithm for its solution. Clearly, understanding complexity places the search for algorithms in a more structured and orderly framework. No wonder, thus, that classical complexity theory has attracted a great deal of intellectual effort, and that its main conundrum, the possible distinction (or otherwise) between the class P (problems which can be computed in polynomial time) and the class NP (problems whose possible solution can be checked in polynomial time) is considered one of the outstanding mathematical conjectures of the age. Yet, classical complexity theory is at odds with the paradigm of numerical computation. The starting point of complexity theory is the Turing machine and its framework is that of discrete operations on integer quantities. Even if a floating-point number is represented as a finite sequence of binary numbers, this provides little insight in trying to answer questions with regard to the cost and complexity of algorithms for problems in analysis.

Having been denied the unifying comfort of the Turing machine, 'real number complexity' has developed in a number of distinct directions. Perhaps the oldest is the ubiquitous counting of *flops* (floating point operations), familiar from undergraduate courses in numerical linear algebra. It is relatively straightforward to calculate the cost of finite algorithms (e.g., Gaussian elimination or fast Fourier transform) and, with more effort, to establish the cost of iterative procedures for a raft of linear-algebraic problems. Yet, it is considerably more difficult to establish the complexity of the underlying problem from simple flop-counting arguments, and the few successes of this approach (e.g. Winograd's result on the complexity of the discrete Fourier transform

[1] We are assuming serial computer architecture. This is motivated not just by considerations of simplicity and economy of exposition, but also because complexity theory in parallel architectures is ill developed and, considering the sheer number of distinct and noncommensurable parallel architectures, arguably of little lasting importance.

[74]) are merely exceptions proving this rule. Moreover, counting flops becomes considerably less relevant once we endeavour to approximate *continuous* entities: differential and integral equations. The cost and complexity are equally relevant in that setting, but the question of how well a discrete system models a continuous one (to which we will return in a different setting in Subsection 2.6) is considerably more challenging than simply counting floating-point operations.

A more structured attempt to introduce complexity to real-number calculations is *information-based complexity.* The point of departure is typically a class of continuous problems, e.g. multivariate integrals or ordinary differential equations, defined very precisely in a functional-analytic formalism. Thus, the integrals might be over all functions that belong to a specific Sobolev space in a specific d-dimensional domain, ordinary differential equations might be restricted by size of the global Lipschitz constant and smoothness requirements in a specific interval etc. An algorithm is a means of using imperfect (finite, discrete, perhaps contaminated by error) *information* in a structured setting to approximate the underlying continuous problem. Its cost is measured in terms of the information it uses, and its performance assessed in some norm over the entire class of 'acceptable' functions. Information-based complexity has elicited much controversy [51, 64] but it is safe to state that, concentrating our minds and helping to define the underlying subject matter and its methodology in precise mathematical terms, it has contributed a great deal to our understanding of complexity issues.

The latest – and most ambitious – attempt to fashion complexity for real-number calculations is due to Steve Smale and his circle of colleagues and collaborators. It is elaborated in heroic scope in [5] and addressed by Cheung, Cucker and Ye [15]. Recall that the organising focus for 'discrete complexity' is provided by the Turing machine but that the latter is inadequate for real-number computations. The central idea is to replace the Turing machine by a different formal construct, capable of basic operations on *real-number* inputs and producing real-number outputs. Further details are beyond the scope of this essay, but it suffices to mention that the entire edifice of 'Turing-machine complexity' can be extended, at great intellectual effort, to a substantially more elaborate framework, inclusive of questions (and partial answers) to issues of the type "P $\neq$ NP?".

The formal-machine approach to real-number complexity is already feeding back into numerical analysis 'proper', not least for the calculation of zeros of nonlinear functions [5, 57], but also affecting optimization [15]. This, indeed, is the great virtue of complexity theory in its many manifestations: by focusing the mind, formalising sometimes vague activity in precise terms, quantifying the computational effort in an honest manner, it ultimately leads to a better understanding of existing algorithms and the frontiers of possible computation and, ultimately, to better algorithms.

2.5 Probability theory and statistics

The historical development of probability and statistics provides an interesting contrast with that of numerical analysis. Both fields were essentially despised for many years, but probability theory became respectable after Kolmogorov's measure-theoretic axiomatisation in 1933. However statistics has continued to be somewhat unfashionable, for reasons often similar to numerical analysis, and its relationship with probability theory has often faltered. In David Williams' memorable words in the preface to [72]: "Probability and Statistics used to be married; then they separated; then they got divorced; now they hardly see each other." There is no need for this breach to continue, and it is to be hoped that Williams' excellent text will help to introduce many mathematicians to statistics. Certainly, many of us were turned off by statistics courses concentrating on the mechanics of quantiles, medians and hypothesis testing, often omitting the beautiful mathematics underlying these topics. There is certainly an analogy to be made here with those numerical analysis courses which spend far too much time on floating point error analysis, whilst avoiding vastly more interesting (and important) topics. This is not to deny error analysis its place, merely to emphasize that it is just part of the picture, not its focal point.

It should be emphasized that there is really no clear boundary between numerical analysis and statistics. For example, given an approximation to a function known only through values corrupted by noise, how do we decide whether the fit is good? All numerical analysis texts cover least squares, but few cover the statistical theory of Chi-Squared tests. Similarly, many statistics texts cover least squares from the perspective of deriving minimum-variance unbiased estimators, but very few devote sufficient space (if any) to the singular value decomposition, or the fact that there are many reasons to *avoid* the least squares measure of error, given its particular sensitivity to outliers in the data, instead using the many excellent algorithms available for minimizing other norms of the error; see, for instance, [69]. Further, both numerical analysis and statistics are evolving rapidly, driven by the importance of stochastic simulation. One particular common ground is that of mathematical finance, which often moves between partial differential equations and stochastic simulations in order to price options.

One topic which perfectly illustrates the deep connections between numerical analysis and statistics is that of *random matrices*. We recently rediscovered some properties of symmetric Gaussian matrices, which deserve to be better known. Furthermore, the analysis is a good example of numerical linear algebra that is unknown to most probabilists, as well as some probability theory forgotten, or never properly absorbed, by numerical analysts.

A *Gaussian random matrix* is simply a matrix whose elements are independent, normalized Gaussian random variables. We concern ourselves with the special case of *symmetric* Gaussian random matrices, for which only the elements on, or above, the diagonal are independent. Such matrices have been studied for decades, by physicists and mathematicians, and we refer the reader to Alan Edelman's excellent (as yet unpublished) book [26] for their fascinating history and background. Here, we merely observe that their spectral properties display highly regular behaviour as the size of the matrices tends to infinity. Coming to this topic afresh via an initially unrelated problem in geometric integration, we began to consider spectral properties of such matrices, but were limited by the cost of the calculation. To generate the eigenvalues of a symmetric Gaussian random matrix, we simply generated $n(n+1)/2$ independent samples from a normalized Gaussian pseudo-random number generator and then computed the eigenvalues of the symmetric matrix, which requires $\mathcal{O}(n^3)$ floating point operations. This cubic cost in computation, and quadratic cost in storage, limit simulations to rather modest n and, displaying the cardinal mathematical virtue of impatience, we began to consider how this might be improved.

Now, given any vector $\boldsymbol{v} = (\, v_1, \dots, v_n \,)^\top \in \mathbb{R}$, it is possible to construct a reflection $Q \in \mathbb{R}^{n\times n}$ for which

$$Q\boldsymbol{v} = \begin{pmatrix} v_1 \\ \sqrt{v_2^2 + \cdots + v_n^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

and we observe that Q is a symmetric orthogonal matrix. In numerical analysis texts, it is traditional to refer to such orthogonal matrices as Householder reflections, in honour of A.S. Householder, a pioneer of numerical linear algebra. In fact, it is easy to check that

$$Q = I - 2\boldsymbol{u}\boldsymbol{u}^\top,$$

where $\boldsymbol{u} = \boldsymbol{w}/\|\boldsymbol{w}\|$ and $\boldsymbol{w}$ is defined by the equations

$$\boldsymbol{w} = \begin{pmatrix} 0 \\ v_2 + \|\boldsymbol{v}\| \\ v_3 \\ \vdots \\ v_n \end{pmatrix}.$$

Thus, given any Gaussian random matrix A and letting $\boldsymbol{v}$ be the first column of A, we obtain

$$A^{(1)} := QAQ = \begin{pmatrix} A_{1,1} & \sqrt{A_{1,2}^2 + \cdots + A_{1,n}^2} & 0 & \cdots & 0 \\ \sqrt{A_{1,2}^2 + \cdots + A_{1,n}^2} & & & & \\ 0 & & & & \\ \vdots & & & \hat{A} & \\ 0 & & & & \end{pmatrix}.$$

Of course, $A^{(1)}$ and A have identical eigenvalues because Q is orthogonal. Now let us recall that, if $\boldsymbol{X} \in \mathbb{R}^m$ is a normalized Gaussian random vector (i.e. its elements are independent normalized Gaussian random variables), then $Q\boldsymbol{X}$ is also a normalized Gaussian random vector. Further, the Euclidean norm $\|\boldsymbol{X}\|$ of a Gaussian random vector is said to have the χ_m distribution. Thus the elements $A_{1,2}^{(1)} = A_{2,1}^{(1)}$ have the χ_{n-1} distribution, whilst $\hat{A}$ is an $(n-1)\times(n-1)$ symmetric Gaussian random matrix. Recurring this contruction, we obtain a symmetric, tridiagonal random matrix T, with the same spectrum as the original matrix A, whose elements have the following properties: (i) the random variables $T_{1,1}, \dots, T_{n,n}$ and $T_{2,1}, T_{3,2}, \dots, T_{n,n-1}$ are independent, (ii) the diagonal elements are normalized Gaussian random variables, and (iii) $T_{k+1,k}$ has the χ_{n-k} distribution, for $k = 1, 2, \dots, n-1$. Now the square of a random variable with the χ_m distribution has, of course, the χ_m^2 distribution, and this is precisely a Gamma distribution (see [72]). Further, many papers have suggested efficient algorithms for the generation of pseudo-random numbers with the Gamma distribution (see, for example, [72]). Finally, the famous *QR algorithm* ([29]) enables the eigenvalues of a symmetric tridiagonal $n \times n$ matrix to be computed in $\mathcal{O}(n)$ operations. Thus we can now generate symmetric tridiagonal random matrices instead of random Gaussian matrices, the cost being reduced from cubic (in n) to linear.

We are unaware of an exposition of the above algorithm in the literature, hence our detailed treatment. Stewart [58] and Trotter [65] seem to have been the first to observe these facts, the former being concerned with the generation of random orthogonal matrices, while the latter used the distribution of the random matrix T to derive Wigner's celebrated circle law. Here let us remark that the *expected value* $\mathbb{E}|T|$ of the random matrix $|T|$ is identically zero, except that the subdiagonal and superdiagonal elements form the sequence $1, \sqrt{2}, \ldots, \sqrt{n-2}, \sqrt{n-1}$. Thus, the principal minors of the Jacobi matrix $\mathbb{E}|T|$ obey the same three-term recurrence relation as monic Hermite polynomials, and this observation is a crucial step in Trotter's derivation of the circle law.

It is frustrating to rediscover a mathematical property, but a familiar feeling for us all. However, there might be similar insights awaiting us for other classes of random matrices. Furthermore, one of the most active areas of modern graph theory is that of *random graph theory*, where many recent papers study spectral properties of the adjacency matrix generated by a random graph. We suspect that there are some beautiful results to be discovered at the intersection of random graph theory, random matrix theory and numerical linear algebra.

2.6 Nonlinear dynamical systems

At the first level of approximation, each mathematical problem is one of a kind, separated from the universe of 'other' problems. This comforting state of affairs does not survive further deliberation: the 'space of mathematical problems' is continuous, not discrete. In particular, most meaningful problems in the realm of analysis are equipped with a multitude of parameters. Some parameters are 'explicit', other are conveniently buried in the formulation of the problem as initial and boundary values. Discretization methods for such problems depend on all these parameters, in addition to further parameters (e.g., the step size or the error tolerance) introduced by the numerical procedure. The theory of dynamical systems endeavours to elucidate the behaviour of time-dependent systems, whether continuous *flows* or discrete *maps,* when parameters vary. The main focus is on long-term dynamics and on abrupt qualitative changes (bifurcation points) in the underlying parameters that lead to a change in asymptotic behaviour.

Classical dynamical systems focus on a particular flow or a particular map. In a computational context, though, the situation is subtly different: we must consider two distinct dynamical systems in unison: the *flow* representing the differential system and the *map* corresponding to the discretization method. It is the similarity (or otherwise) of their asymptotic behaviour and their bifurcation patterns which is of interest in a computational setting.

Perhaps the most intriguing asymptotic phenomenon, which has penetrated public consciousness far beyond mathematical circles, is *chaos* – the capacity of some dynamical systems to display very sensitive behaviour in response to tiny changes in parameters. A celebrated case in point is that of the *Lorenz equations,* three coupled nonlinear ordinary differential equations originating in mathematical meteorology which exhibit $\mathcal{O}(1)$ variation in response to arbitrarily small variation in their initial conditions. It is important to bear in mind that chaos is not associated with total disorder. Typically (as is the case with Lorenz equations) the range of all possible states in a chaotic regime is restricted to a well-defined, lower-dimensional object, a *chaotic attractor.* It is this balancing act, chaos embedded within order and governed by precise mathematical laws, which makes the theory of nonlinear dynamical systems so enticing.

In the context of numerical methods, chaotic behaviour assumes added significance. It is perfectly possible for the underlying flow to be non-chaotic, while the discretized map exhibits chaotic behaviour. Thus, the numerical method may display behaviour which is qualitatively wrong. Although this can be avoided in principle by a judicious choice of discretization parameters, this is not always easy and sometimes impossible, because of hardware limitations on storage and speed. Therefore, it is essential to understand this phenomenon, which is elaborated by Yamaguti and Maeda in [75].

"Numerical chaos" is just one possible way in which discretization can lead to the wrong conclusion with regard to the qualitative behaviour of the underlying differential system. With greater generality, the goal is to compare the entire menagerie of possible types of asymptotic behaviour (in other words, all global attractors) in flows and their discretization. A very great deal has been accomplished in that sphere. In particular, it is clear that not all discretization methods are alike: some are prone to exhibit incorrect asymptotic behaviour for even small time steps and space grids, while others are immune to this adverse behaviour [60]. The theory of nonlinear dynamical systems is, arguably, the most powerful tool at the disposal of a modern theoretical numerical analyst, attempting to explain long-time behaviour of discretization methods.

2.7 Topology and algebraic geometry

Mathematicians famously endeavour to reduce problems to other problems whose solutions are already known. The main concept of *continuation* (a.k.a. *homotopy*) methods is to travel in the opposite direction: commencing from a trivial problem, with known solution, deform it continuously to the problem whose solution we want to compute [1, 41, 44].

To first order, this is an exercise in nonlinear dynamical systems: we have two problems, $\boldsymbol{f}$ and $\boldsymbol{g}$, say, for example, systems of algebraic or differential equations or eigenvalue problems. The solution of $\boldsymbol{f}$ is known and, seeking the solution of $\boldsymbol{g}$, we consider a new function, $\boldsymbol{\varphi}_\tau = (1-\tau)\boldsymbol{f} + \tau\boldsymbol{g}$, where the *continuation parameter* τ resides in $[0,1]$. Since $\boldsymbol{\varphi}_0 = \boldsymbol{f}$, we already know its solution and, like Theseus in the Minoan labyrinth, we grab the thread and pursue it, moving in small steps from $\boldsymbol{\varphi}_{\tau_k}$ to $\boldsymbol{\varphi}_{\tau_{k+1}}$, where $0 = \tau_0 < \tau_1 < \cdots < \tau_m = 1$, until we can reach the exit $\boldsymbol{g}$. Unfortunately, neither Ariadne's love nor bifurcation theory are sufficient to attain our goal. The most vexing issue is the number of solutions, and we recall that Theseus left Ariadne. Suppose, for example, that both $\boldsymbol{f}$ and $\boldsymbol{g}$ are systems of multivariate polynomial equations. In general, we do not know the number of solutions of $\boldsymbol{g}$, hence cannot choose $\boldsymbol{f}$ to match it and follow each solution along a different homotopy path. Moreover, there is no guarantee that a homotopy path from a solution of $\boldsymbol{f}$ will indeed take us to a solution of $\boldsymbol{g}$ (rather than, for example, meandering to infinity) or that two solutions of $\boldsymbol{f}$ will not merge at the same solution of $\boldsymbol{g}$.

The basic mathematical tool to investigate how the zeros, say, of $\boldsymbol{\varphi}_\tau$ change as τ varies from 0 to 1 is provided by *degree theory,* which explains the geometry of solutions in topological terms [1, 41]. It allows the construction of algorithms that avoid unwelcome bifurcations and divergence.

In the important case of systems of multivariate polynomial equations we have at our disposal an even more powerful tool, namely *algebraic geometry.* At its very heart, algebraic geometry investigates *varieties,* sets of common zeros of polynomial systems, and the ideals of polynomials that possess such zeros. To illustrate the gap between the naive counting of zeros and even the simplest problems within the realm of homotopy methods, consider the problem of finding eigenvalues and normalised eigenvectors of a symmetric $n \times n$ matrix A. We can write it as a system of $n+1$ quadratic equations in $n+1$ variables,

$$\begin{aligned} g_1(\lambda, v_1, v_2, \ldots, v_n) &= \sum_{k=1}^{n} A_{1,k} v_k - \lambda v_1, \\ &\vdots \\ g_n(\lambda, v_1, v_2, \ldots, v_n) &= \sum_{k=1}^{n} A_{n,k} v_k - \lambda v_n, \\ g_{n+1}(\lambda, v_1, v_2, \ldots, v_n) &= \sum_{k=1}^{n} v_k^2 - 1. \end{aligned}$$

According to the celebrated *Bézout theorem,* the upper bound on the number of all possible solutions of a multivariate polynomial system is the product of the degrees of all individual polynomials. In our case this bound is 2^{n+1}, wildly exceeding the sharp bound of $2n$ (there are n eigenvalues and, if $\boldsymbol{v}$ is a normalised eigenvector, so is $-\boldsymbol{v}$). The theory underlying homotopy algorithms must be powerful enough to know the correct number of solutions without help from linear algebra, since such help is available only in limited number of cases. Moreover, it must ensure that homotopy paths neither meander, nor merge, nor diverge. Modern techniques in algebraic geometry are equal to this challenge and provide a powerful theoretical underpinning for continuation methods [44].

Homotopy is not the only concept at the intersection of topology and 'classilcal' computation. Another development of growing significance, pioneered by Enzo Tonti [63] and recently embraced by an increasing number of professionals, is the use of *combinatorial topology* to embed laws of physics directly into the geometry of discretization space. Lest it be said that this is yet another (strange, beautiful and useless) flight of fancy of pure mathematicians, we should state at the outset that the origin of this set of ideas is in practical needs of physics [36, 63], electromagnetism [37] and geometric modelling [14].

The work of Chard and Shapiro [14] is typical of this approach. Glossing over many technicalities and (essential) details, the construction of a 'topologically-aware' space discretization commences from an m-cell, a set homeomorphic to a closed unit m-ball in $\mathbb{R}^n$, $m \in \{0, 1, \ldots, m\}$. (The reader might think of an n-cell as an element in a finite-element tessellation and of the m-cells, $m < n$, as faces of higher-dimensional cells.) A *complex* is a collection $\mathcal{C}$ of cells, such that the boundary of each m-cell is a finite union of $(m-1)$-cells therein and a nonempty intersection of two cells in $\mathcal{C}$ is itself a cell in the complex. The *star* of a cell $X \in \mathcal{C}$, where $\mathcal{C}$ is a complex, is the set of all the adjacent cells of dimension greater than or equal to that of X. Chard and Shapiro propose constructing a collection of stars which automatically encodes for specific physical laws that admit topological interpretation. The most striking example is that of *balance:* a flow of physical quantity (energy, heat, mass) across a boundary of a domain is in perfect balance with storage, generation and destruction of that quantity inside. Another example is Stokes' theorem, which is automatically satisfied in the Chard–Shapiro 'starplex', in every cell, of any dimension. Once the physical model is formulated and discretized across this tessellation, it automatically obeys the laws in question. Rather than encoding physics in the software of the numerical method, it is already hardwired into the topology of the underlying space! Other applications of combinatorial topology to computation are similar. They can deal only with physical laws that can be expressed by topological concepts, but this is already powerful enough to allow for the incorporation of multiphysics into numerical methods for complicated partial differential equations [37].

2.8 Differential geometry

A central reason for a widespread aversion to practical computing is that it abandons the nobility of mathematics, its quest for qualitative entities, its *geometry,* for mundane number crunching. Yet, there is no need to abandon geometry for the sake of computation and the two can coexist in peace and, indeed, in synergy. This is the main idea underlying the emerging discipline of *geometric integration* [9, 10, 30, 46], whose language is differential geometry and whose aim is the discretization of differential equations whilst respecting their structural attributes. Many initial-value differential systems possess invariants, qualitative features that remain constant as the solution evolves in time:

- *Conservation laws,* i.e. functionals that stay constant along the solution trajectory. For example, the energy $H(\boldsymbol{p}(t), \boldsymbol{q}(t))$ of the *Hamiltonian* system

$$\boldsymbol{p} = -\frac{\partial H(\boldsymbol{p},\boldsymbol{q})}{\partial \boldsymbol{q}}, \quad \boldsymbol{q}' = \frac{H(\boldsymbol{p},\boldsymbol{q})}{\partial \boldsymbol{p}}, \qquad t \geq 0, \tag{2.2}$$

  remains constant along the solution trajectory. In other words, the exact solution of the system, which in principle is a trajectory in a $(2d)$-dimensional Euclidean phase space (where $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^d$), is restricted to the (usually nonlinear, differentiable) *manifold*

$$\mathcal{M} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^d \ : \ H(\boldsymbol{x}, \boldsymbol{y}) = H(\boldsymbol{p}(0), \boldsymbol{q}(0))\}.$$

- *Symmetries:* transformations which, when applied to dependent and independent variables, produce another solution to the same differential system. Perhaps the most basic is the SO(3) symmetry, which is shared by all laws of mechanics [2], since they must be independent of the frame of reference. Another is *self* *similarity,* addressed at some length in [10]. Euclidean symmetry and self similarity might seem very different animals, yet they are examples of a wider class of symmetries generated by *Lie groups*, and can be phrased geometrically as acting on the vector field, rather than the configuration manifold, of the underlying equation. In other words, in the same manner that conservation laws are associated with a manifold $\mathcal{M}$, say, symmetries are associated with the *tangent bundle* $\mathrm{T}\mathcal{M}$.

- *Symplectic structures* occur in the context of Hamiltonian equations (2.2) – in fact, they characterize such equations [2]. There are several equivalent formulations of symplecticity and for our purpose probably the most illustrative is

$$\frac{\partial(\boldsymbol{p}(t),\boldsymbol{q}(t))}{\partial(\boldsymbol{p}(0),\boldsymbol{q}(0))}^{\top}\begin{pmatrix} O & I \\ -I & O \end{pmatrix}\frac{\partial(\boldsymbol{p}(t),\boldsymbol{q}(t))}{\partial(\boldsymbol{p}(0),\boldsymbol{q}(0))}=\begin{pmatrix} O & I \\ -I & O \end{pmatrix}, \qquad t \geq 0.$$

  Brief examination confirms that the invariant in question evolves on the *cotangent bundle* $\mathrm{T}^*\mathcal{M}$ of some manifold $\mathcal{M}$: the set of all linear functionals acting on the tangent bundle $\mathrm{T}\mathcal{M}$.

We have mentioned three distinct types of invariants, and the list is by no means exhaustive. Two types of invariants have been illustrated by means of the Hamiltonian system (2.2) and, indeed, it is important to emphasize that the same equation might possess different invariants, possibly of different types. The common denominator to all invariants is that they can be phrased in the language of differential geometry. Indeed, this appears to be the language of choice in geometric integration, which can be applied to a long list of other invariants, e.g. conservation of volume, angular momentum, potential vorticity and fluid circulation.

Geometric integration is a recent addition to the compendium of tools and ideas at the interface of computation and mathematics. Its current state of the art is reviewed in [10], hence it will suffice to mention one example, for illustration, relevant to conservation laws of a special type. A *Lie group* is a manifold $\mathcal{G}$ endowed with a group structure, consistent with the underlying topology of the manifold. Useful examples comprise of the *orthogonal group* $\mathrm{O}(n)$ of $n \times n$ real orthogonal matrices and the *special linear group* $\mathrm{SL}(n)$ of $n \times n$ real matrices with unit determinant. The tangent space of $\mathcal{G}$ at its unit element has the structure of a *Lie algebra,* a linear space closed under commutation. For example, the Lie algebra corresponding to $\mathrm{O}(n)$ is $\mathfrak{so}(n)$, the linear space of $n \times n$ real skew-symmetric matrices, while $\mathfrak{sl}(n)$, the set of $n \times n$ real matrices with zero trace, is the Lie algebra of $\mathrm{SL}(n)$. The importance of Lie algebras in the present context is underscored by the observation that $X \in \mathfrak{g}$ implies $\mathrm{e}^X \in \mathcal{G}$, where $\mathcal{G}$ is a Lie group, $\mathfrak{g}$ 'its' Lie algebra and, in the case of matrix groups, e^X is the familiar matrix exponential of X.

A Lie group $\mathcal{G}$ is said to *act* on a manifold $\mathcal{M}$ if there exists a mapping $\lambda : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ which is consistent with the group multiplication: $\lambda(x_1, \lambda(x_2, y)) = \lambda(x_1 x_2, y)$ for all $x_1, x_2 \in \mathcal{G}$, $y \in \mathcal{M}$. A group action is *transitive* if, for every $y_1, y_2 \in \mathcal{M}$, there exists at least one $x \in \mathcal{G}$ such that $\lambda(x, y_1) = y_2$. A manifold endowed with a transitive group action is said to be *homogeneous.* Numerous manifolds arising in practical applications are homogeneous, e.g. $\mathrm{O}(n)$ acts transitively on the $(n-1)$-sphere $\mathrm{S}^{n-1} \subset \mathbb{R}^n$ by multiplication, while the *isospectral orbit*, consisting $\mathcal{I}(n, Y_0)$ of all $n \times n$ symmetric matrices which are similar to a specific matrix $Y_0 \in \mathrm{Sym}(n)$, can be represented via the orthogonal similarity action

$$\mathcal{I}(n, Y_0) = \{QY_0Q^\top \,:\, Q \in \mathrm{O}(n)\}.$$

Here $\mathrm{Sym}(n)$ is the space of $n \times n$ real symmetric matrices. As a matter of independent interest, we leave it to the reader to prove that it is also a homogeneous manifold (e.g. by using the *congruence action* $\lambda(X, Y) = XYX^\top$, where X is $n \times n$ and real, while $Y \in \mathrm{Sym}(n)$) and identify striking similarities between this and a familiar factorization procedure from numerical linear algebra.

Many differential equations of interest evolve on homogeneous manifolds. Thus, it is not difficult to verify that the solution of

$$\boldsymbol{y}' = A(t, \boldsymbol{y})\boldsymbol{y}, \quad t \geq 0, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0 \in \mathrm{S}^{n-1}, \tag{2.3}$$

where $A : \mathbb{R}_+ \times \mathrm{S}^{n-1} \to \mathfrak{so}(n)$, evolves on S^{n-1}, while the solution of the *isospectral flow*

$$Y' = [B(t, Y), Y], \quad t \geq 0, \qquad Y(0) = Y_0 \in \mathrm{Sym}(n),$$

where $B : \mathbb{R}_+ \times \mathcal{I}(n, Y_0) \to \mathfrak{so}(n)$ lives in $\mathcal{I}(n, Y_0)$. Yet, any attempt to discretize either equation with a standard numerical method whilst respecting its invariants (the Euclidean norm and the eigenvalues, respectively) is unlikely to be successful. Symplectic Runge–Kutta methods (e.g. Gauss–Legendre schemes) will stay on a sphere, but no classical algorithm may respect isospectral structure for $n \geq 3$ [40]. The intuitive reason is that S^{n-1}, $\mathcal{I}(n, Y_0)$ and other homogeneous manifolds are nonlinear structures, while classical methods for differential equations – Runge–Kutta, multistep, Taylor expansions, ... – do notoriously badly in adhering to nonlinear surfaces. This is a major justification for Lie-group solvers, the new breed of methods originating in geometric integration [40]. To give the simplest possible example, the standard Euler method, as applied to (2.3), produces the solution sequence $\boldsymbol{y}_{m+1} = [I + h_m A(t_m, \boldsymbol{y}_m)]\boldsymbol{y}_m$, $m \in \mathbb{Z}_+$, where $\boldsymbol{y}_m \approx \boldsymbol{y}(t_m)$ and $h_m = t_{m+1} - t_m$ is the step size. It is easy to check that, in general, $\boldsymbol{y}_{m+1} \notin \mathrm{S}^{n-1}$. Consider, however, the *Lie–Euler method:* Euler's scheme applied at the algebra level,

$$\boldsymbol{y}_{m+1} = \mathrm{e}^{h_m A(t_m, \boldsymbol{y}_m)}\boldsymbol{y}_m, \qquad m \in \mathbb{Z}_+.$$

Since $h_m A(t_m, \boldsymbol{y}_m) \in \mathfrak{so}(n)$, ii is true that $\mathrm{e}^{h_m A(t_m, \boldsymbol{y}_m)} \in \mathrm{O}(n)$, whence $\boldsymbol{y}_m \in \mathrm{S}^{n-1}$. We refer to [10, 40] for a thorough discussion of Lie-group solvers.

Differential geometry is relevant to computation also in situations when the conservation of invariants, the essence of geometric integration, is not at issue. An important example is computer modelling and the manipulation of spatial images. How do we identify where one object ends and the other starts? How does an image move? How do we blend images? All these might be activities natural to the human (or animal) brain but it is far more challenging to teach a computer basic geometry. The very fuzziness of the above questions is not intrinsic, but reflects the sheer difficulty of formalising "geometry in motion" in mathematical terms. As reported in [52], important strides in this direction have been made in the last decade, combining differential-geometric principles with partial differential equations, variational principles and computational methods.

2.9 Abstract algebra

Zeros of polynomials again.... Suppose thus that we wish to evaluate the zeros of a system of multivariate polynomial equations,

$$p_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \ldots, m, \qquad \boldsymbol{x} \in \mathbb{C}^n, \tag{2.4}$$

and, at the first instance, check whether this system has any zeros at all. This is one of the hallowed problems of mathematics, the 'master problem' of algebraic geometry, and immortalised by Hilbert's *Nullstellensatz.* Notwithstanding the success of purely numerical methods, upon which we have already commented in Subsection 2.7, they are accompanied by a multitude of perils. Some of them have to do with the notorious ill conditioning of polynomial zeros. Others originate in the fact that, although polynomials p_k often have 'nice' (e.g. integer) coefficients, numerical methods reduce everything to the lowest common denominator of floating-point arithmetic. Finally, numerical methods have a tendency to obscure: their final output, a long string of floating-point numbers, often hides interesting dependencies among the different polynomials and variables.

An alternative to numerics is symbolic computation, in our case a methodical conversion of $\{p_1, p_2, \ldots, p_m\}$ into a *single* polynomial in a *single* variable, while retaining integer (or rational) coefficients whenever such coefficients prevail in the original problem. An early success of symbolic computation was an algorithm, due to Bruno Buchberger [7], which accomplishes this task employing methods of *commutative algebra.* Since most readers of this essay are likely to be numerical analysts, with little experience of what happens inside symbolic calculation packages, let us consider a simple example of a trivial version of the Buchberger algorithm, as applied to the polynomial system (2.4) with $n = m = 3$, $\boldsymbol{x} = (x, y, z)$ and

$$p_1 = 3x^2y - z - 2, \qquad p_2 = xyz + 1, \qquad p_3 = xz - 2xy + y^2.$$

We commence by imposing a *total ordering* on all terms of the form $x^\alpha y^\beta z^\gamma$. Among the many alternatives, we opt for the (far from optimal!) lexicographic ordering implied by $x \succ y \succ z$. Let $\mathrm{ht}(p)$, the *height* of the polynomial p, denote its largest term subject to our ordering, thus $\mathrm{ht}(p_1) = x^2y$, $\mathrm{ht}(p_2) = xyz$, $\mathrm{ht}(p_3) = xy$. We now set p_4 as equal to the linear combination (easily available from their height functions) which eliminates the largest terms in $\{p_1, p_2\}$,

$$p_4 = zp_1 - 3xp_2 = -z^2 - 2z - 3x.$$

Note that we can now express x in terms of the 'least' variable z by setting $p_4 = 0$, namely $x = -(2z + z^2)/3$. Next, we choose the pair $\{p_2, p_4\}$, eliminate the highest elements and obtain

$$p_5 = -3 + 2yz^2 + yz^3.$$

Although we can now recover y in terms of z, we desist, since we cannot (yet) express y as a *polynomial* in z. Instead, we turn our attention to the pair $\{p_3, p_5\}$, eliminate the highest terms and obtain $q = -6x + xz^4 + y^2z^3 + 4xyz^2$: a rather messy expression! However, substracting multiples of p_3, p_4 and p_5, we can *reduce* q to a much simpler form, namely

$$p_6 = 9y + 12z + 6z^2 - 4z^4 - 4z^5 - z^6.$$

Thus, letting $p_6 = 0$, y is expressible as a polynomial in z. Finally, we shave off the highest terms in $\{p_2, p_6\}$, reduce using p_4, and derive

$$p_7 = -27 - 24z^3 - 24z^4 - 6z^5 + 8z^6 + 12z^7 + 6z^8 + z^9,$$

a univariate polynomial! The original system has been thus converted to the *Gröbner base* $\{p_4, p_6, p_7\}$.

On the face of it, we are dealing here with a variant of Gaussian elimination, eliminating polynomial terms by forming linear combinations with polynomial coefficients. This hides the main issue: the Buchberger algorithm *always* terminates and provides a constructive means to compute a Gröbner base [8, 19, 68]. Moreover, it can be generalised to *differential Gröbner bases,* dealing with differential (rather than polynomial) equations. In general, symbolic algebra is rapidly emerging as a major means of mathematical computation, capable of such diverse tasks as the verification of summation formulae [78] or computing symmetries of differential equations [39]. The theoretical framework rests upon commutative algebra, with substantial input from algebraic and differential geometry.

This review of "pure mathematics influencing computational mathematics" would not be complete without mentioning a fascinating example of computational mathematics reciprocating in kind and enriching pure mathematics, specifically abstract algebra (and theoretical physics). Needless to say, computation helps in understanding mathematics by its very nature, by fleshing out numerical and visual data. Yet, here we wish to draw attention to a different pattern of influence: seeking mathematical structure to explain the behaviour of computational processes may lead to an insight into seemingly-unrelated domains of pure-mathematical research.

A pioneering effort to understand order conditions for Runge–Kutta methods led John Butcher, in the late 1960s, to a theory that illuminates the complicated expansions of Runge–Kutta maps via *rooted trees.* The complex and unpleasant procedure of deriving order conditions for high-stage methods thus reduces to a transparent exercise in recursion over rooted trees and this, indeed, is the method of choice in the modern numerical analysis of ordinary differential equations. In particular, endeavouring to explain the interaction of order conditions once several Runge–Kutta schemes are composed, Butcher endowed his expansion with an algebraic structure [12]. For almost thirty years [12] was yet another a paper at the theoretical frontier of numerical analysis, only to be discovered and reinterpreted by algebraists in the late 1990s. It turns out that the *Butcher group* provides a superior representation of a *Hopf algebra,*, with very important implications for both algebra and its applications in theoretical physics [6, 18]. It is gratifying to confirm that computational mathematicians can sometimes repay pure mathematics in kind.

2.10 . . . and beyond

It is possible to extend the intellectual exercise of this section to most areas of pure mathematics, methodically progressing through the AMS (MOS) subject classification. Sometimes the connection is somewhat forced: most applications of *number theory* appear to be in random-number generation [77], say, and (to the best of our knowledge) the only significant way in which ergodic theory impacts upon computation is via the ergodic properties of some nonlinear dynamical systems. Sometimes the connection might appear mundane, at first glance, from the mathematical point of view: graph theory is often used in numerical analysis as a useful tool, e.g. in parallel computation, the numerical algebra of sparse matrices, expansions of Runge–Kutta methods, Lie-group and splitting algorithms in geometric integration, ..., but (arguably) these applications are not particularly profound from a graph-theoretical standpoint: they are all simple instances of planar graphs featuring in intricate analytic structures. Yet, one way or the other, our original contention stands: the mathematical foundations of computation are *all* of mathematics.

3 The 'computational' in computational mathematics

3.1 Adaptivity

Like any domain of intellectual endeavour, computational mathematics progresses from the simple to the complex. 'Simple' in this context means using an algorithm which is undifferentiated and does not respond dynamically to the computed solution. Thus, typically, the first program written by a student for the solution of differential equations might be a constant-step Runge–Kutta method: regardless of how the solution changes, whether it undergoes rapid oscillations, say, or stands still, whether the local error is huge or nil, the program ploughs on and spews out numbers which often bear little relevance to the underlying problem. Understandably, this is often also the *last* program that a disenchanted student writes....

Clearly, good computation requires the algorithm to respond to the data it is producing and change the allocation of computing resources (step size, size of the grid, number of iterations, ..., even the discretization method itself) accordingly. The purpose of computation is not to produce a solution with least error but to produce reliably, robustly and *affordably* a solution which is within a user-specified tolerance. In particular, in time-dependent problems, a well-structured algorithm will typically employ two intermeshed mechanisms: a monitor of the error incurred locally during the computation and a means to respond to this error bound (or its estimate) by changing the algorithm's parameters (e.g. step size or space tessellation) or locally improving the accuracy of the solution.

The solution of time-dependent differential and integral equations is not the only area where adaptivity is conducive to good computing practice. An intriguing illustration of adaptivity in a different setting is *lossy data compression*, in a recent algorithm of Cohen, Dahmen, Daubechies and DeVore, using wavelet functions [17]. Instead of adapting a spatial or temporal structure, it is the expansion of an image (expressible as a long string of "0" and "1"s) which is 'adapted', having been first arranged in a tree structure that lends itself to orderly extraction of the most significant bits.

All this emphasis on *adaptivity* is hardly 'foundational' and has been the focus of much effort in numerical analysis, scientific computing and software engineering. Having said this, adaptivity is linked to important foundational issues and, ideally, in place of its current niche of half-mathematics, half-art, should be brought into the fully mathematical domain.

It is premature to prophesize about the likely course of 'mathematical adaptivity', yet an increasing wealth of results implies at the potential of this approach. An important development is the emergence of *nonlinear approximation,* bringing together tools of functional and harmonic analysis and approximation theory to bear upon adaptivity in numerical methods for partial differential equations [24, 61]. This timely theory helps to elucidate the implementation of modern and highly flexible means of approximation, not least wavelet functions, to achieve the important goal of discretizing differential and integral equations. Another development that helps us to understand and fine-tune adaptivity is the introduction of monitor functions that reflect geometric and structural features of the underlying solution [10]. Thus Lie symmetries of the underlying differential equations can be used to adapt a numerical solution to deal successfully (and affordably) with discontinuities, blow ups and abrupt qualitative changes.

Adaptivity is not always free of conceptual problems. For example, 'classical' symplectic methods for Hamiltonian ordinary differential equations should be applied with constant (or virtually constant) step size, lest the benefits of symplecticity be voided [30]. At first glance, this represents a setback: we are forced to choose between the blessings of adaptivity and the benefits of symplecticity. However, as in the case of many other 'impossibility results', the main impact of this restriction is to motivate the development of algorithms, *Moser–Veselov integrators,* that, adopting an altogether-different approach, combine adaptivity and symplecticity [45].

The above three instances of adaptivity, namely nonlinear approximation, geometric monitor functions and Moser–Veselov integrators, are distinct and probably unrelated. They hint at the tremendous potential scope of theoretical and foundational investigation into adaptivity.

3.2 Conditioning

It is a sobering thought that, even when a computational solution to a mathematical problem has been found, often following great intellectual and computational effort, its merit might be devalued by poor stability of the underlying algorithm. Most problems do not exist in some discrete universe but form a continuum. Small changes in the statement of the problem, perturbations of a differential operator, of initial or boundary conditions or of other parameters of the problem, might lead to a significantly different *exact* solution. The situation is more complicated once we attempt to compute, because the numerical algorithm itself depends not only on all these parameters, but also on the intrinsic parameters of the algorithm, errors committed in the course of the solution, and the floating-point arithmetic of the underlying computer. This state of affairs is sometimes designated as "stability" or "well posedness" or "conditioning" – purists may argue *ad nauseam* over the precise definitions of these concepts, but it is clear that, one way or the other, they play an instrumental role in computational mathematics. Naive complexity theory, say, makes very little sense if the 'optimal' algorithm is prone to instability or, worse, if the very problem that we endeavour to solve is ill conditioned in the first place. A more sophisticated concept of complexity is required and, indeed, has emerged in the last few years [15].

Stability is perhaps "the most misused and unstable concept in mathematics" [4] and numerical analysis is probably the worst offender in this unwelcome proliferation. For example, in the numerical solution of ordinary differential equations we have A-stability, A(α)-stability, A_0-stability, AN-stability, algebraic stability, B-stability, BN-stability.... And many remaining letters to afford every worker in the field the dubious benefit of yet another eponymous stability concept. The time has come perhaps to introduce order and understanding into the universe of stability, well posedness and conditioning. Although an all-encompassing stability theory belongs to the future, it is possible to make observations and venture ideas even at this early stage.

Time-dependent mathematical systems give rise to at least two broad stability concepts, which we might term *well posedness* (small variations in initial and boundary conditions and in internal parameters induce small variations in the solution in compact intervals) and *structural stability* (small variations do not induce qualitative changes in global dynamics). Both are inherited by discretization methods, although often terminology obscures this. Thus, for example, well posedness of discretization methods for partial differential equations of evolution is termed *stability,* a source of enduring confusion among students and some experienced researchers alike. Note that 'discrete well posedness' is not identical to 'continuous well posedness' since extra parameters (time step, size of the grid) are a consequence of discretization and they play an important role in the definition of numerical well posedness. Yet, in essence, it remains well posedness and should be distinguished from numerical structural stability, the province of computational dynamics.

Another dichotomy, extant in both computational analysis and computational algebra, is between traditional *forward* stability analysis and the approach of *backward error analysis* (a misnomer: in reality it refers to backward stability or conditioning analysis). Forward analysis asks the question that we have already posed time and again in this section: how does the behaviour vary when parameters are perturbed? An alternative approach, pioneered by Jim Wilkinson in [71], is to investigate which perturbed problem is solved *exactly* by a computational algorithm. This is often the method of choice in numerical linear algebra, and it is gaining increasing prominence in the discretization of differential equations and in computational dynamics. Thus, for example, the theory of *shadowing* tells us that (as always, subject to a battery of side conditions) even imperfect modelling of chaotic systems produces exact orbits of the underlying system, with different parameter values [31]. This is indeed the reason why we can model chaotic attractors on a computer relatively successfully. Another powerful application of backward error analysis in the realm of differential equations is in the case of integrable Hamiltonian systems [30]. Since symplectic methods solve a nearby Hamiltonian problem (almost) exactly, their dynamics is identical to the original system (in particular, they evolve on invariant tori), Hamiltonian energy is conserved in an ergodic sense and, last but not least, numerical error accumulates much more slowly.

We do not delude ourselves that the preliminary and hesitant observations above are a basis for a unifying theory of stability and conditioning. Having said this, at least they seem to imply that it is possible to formulate meaningful statements within the realm of metastability. It is fair to assume that a better understanding of this issue might be one of the most useful consequences of the exploration of the foundations of computational mathematics.

3.3 Structure

Mathematics is not merely the science of number. It is the science of pattern and structure. Mathematical statements can be encoded in numbers (and the source file of this essay is merely a string of zeros and ones), but numbers as such are not an end in themselves. Once we wish to harness computation to *understand* mathematics, we are compelled to compute numerical results not as an end but as a means to reveal the underlying structure. This has a number of important implications for the future of computational algorithms.

Firstly, the impetus towards 'general' methods and 'general' software, which can cater to many different problems in a broad category, might be inimical to progress. Once we classify mathematical problems by their structural features, broad categories are much too coarse. We need to identify smaller sets of problems, each with its own common structural denominator and each deserving of separate computational treatment.

Secondly, the attitude of "don't think, the software will do it for you", comforting as it might be to some, will not do. There is no real dichotomy between mathematical and computational understanding. If you wish to compute, probably the best initial step is to learn the underlying mathematics, rather than rushing to a book of numerical recipes. Often when we air this point of view, some colleagues opine that it might create obstacles for scientists and engineers: this, we believe, is an unfair canard. In our experience, thoughtful scientists and engineers possess both the ability and the will to master necessary mathematics, often more so than the lazier members of the computational community.

Thirdly, "structure" itself can assume different guises. It can refer to an attribute of a time-dependent system, in finite time or asymptotically; it can concern algebraic, analytic, geometric or stochastic features. Thus, it requires, on a case-by-case basis, different mathematical themes. In Section 2, we mentioned nonlinear dynamical systems and differential geometry, but clearly this does not exhaust the range of all mathematical subjects relevant to the retention of structure in mathematical computation.

Finally, the retention of structure can be for any of three reasons: mathematical, physical or computational. Sometimes we might wish to retain structure because of its mathematical significance. In other cases structure might possess physical interpretation which is essential to the modelling of the underlying phenomenon from science or engineering. In other cases the retention of structure leads to more accurate and affordable computational algorithms [30]. And, needless to say, there might be cases when retention of structure is of little or no importance.

Computation is not an activity where we can afford to "paint by numbers", suspending our mathematical judgement and discernment. At its best, it requires us to be constantly on our mathematical guard and to deploy, as necessary, different weapons from the mathematical arsenal. The foundations of computational mathematics are the totality of mathematics and we believe that this is what makes our subject, once despised, so vibrant, exciting and full of promise. In a slightly different context, Goethe expressed this judgment beautifully in *Römische Elegien X*:

> Wenn du mir sagst, du habest als Kind, Geliebte, den Menschen
>   Nicht gefallen, und dich habe die Mutter verschmäht,
> Bis du grösser geworden und still dich entwickelt, ich glaub es:
>   Gerne denk ich mir dich als ein besonderes Kind.
> Fehlet Bildung und Farbe doch auch der Blüte des Weinstocks,
> Wenn die Beere, gereift, Menschen und Götter entzückt. [2]

Acknowledgements

We thank Raymond Brummelhuis (Birkbeck College, London), Felipe Cucker (City University, Hong Kong), Alan Edelman (MIT), Elizabeth Mansfield (University of Kent at Canterbury), Nilima Nigam (McGill University, Montreal) and Alex Scott (University College, London).

References

[1] E.L. Allgower and K. Georg. *Numerical Continuation Methods. An Introduction.* Springer-Verlag, Berlin, 1990.

[2] V.I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer Verlag, Berlin, 1989.

[3] B.J.C. Baxter. Positive definite functions on hilbert space. Technical report, Numerical Analysis Group, University of Cambridge, 2002.

[4] R.E. Bellman. *Stability Theory of Differential Equations.* Dover, New York, 1969.

[5] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer, New York, 1998.

[6] Ch. Brouder. Runge–Kutta methods and renormalization. *Eur. Phys. J.*, 12:521–534, 1999.

[7] B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes Math.*, 4:374–383, 1970.

[8] B. Buchberger and F. Winkler, editors. *Gröbner Bases and Applications*, number 251 in London Math. Soc. Lect. Note Series, Cambridge, 1998. Cambridge University Press.

[2]In David Luke's translation [66]:
When you were little, my darling, you tell me nobody liked you —
    Even your mother, you say, scorned you, until as the years
Passed, you quietly grew and matured; and I can believe it —
    It's rather pleasant to think you were a strange little child.
For though the flower of a vine may be still unformed and lack lustre,
In the ripe grape it yields nectar for gods and men.

[9] C.J. Budd and A. Iserles. Geometric integration: Numerical solution of differential equations on manifolds. *Phil. Trans Royal Soc. A*, 357:945–956, 1999.

[10] C.J. Budd and M.D. Piggott. Geometric integration and its applications. In P.G. Ciarlet and F. Cucker, editors, *Handbook of Numerical Analysis*, volume ??, pages ???–??? North-Holland, Amsterdam, 2003.

[11] M.D. Buhmann. Radial basis functions. *Acta Numerica*, 9:1–38, 2000.

[12] J.C. Butcher. An algebraic theory of integration methods. *Maths Comp.*, 26:79–106, 1972.

[13] J. P. R. Christensen C. Berg and P. Ressel. *Harmonic Analysis on Semigroups: theory of positive definite and related functions*. Number 100 in Graduate Texts in Mathematics. Springer, New York, 1984.

[14] J.A. Chard and V. Shapiro. A multivector data structure for differential forms and equations. *Maths & Comps in Simulation*, 54:33–64, 2000.

[15] D. Cheung, F. Cucker, and Y. Ye. Linear programming and condition numbers under the real number computation model. In P.G. Ciarlet and F. Cucker, editors, *Handbook of Numerical Analysis*, volume ??, pages ???–??? North-Holland, Amsterdam, 2003.

[16] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.

[17] A. Cohen, W. Dahmen, I. Daubechies, and R.A. DeVore. Tree approximation and optimal encoding. *Appl. Comput. Harmon. Anal.*, 11:192–226, 2001.

[18] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Comm. Math. Phys.*, 199:203–242, 1998.

[19] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, New York, 1997.

[20] H.B. Curry and I.J. Schoenberg. Pólya frequency functions IV. The fundamental spline functions and their limits. *J. d'Anal. Math*, 17:71–107, 1966.

[21] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.

[22] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[23] C. de Boor, K. Höllig, and S. Riemenschneider. *Box Splines*. Springer, New York, 1993.

[24] R.A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[25] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Anal. Numer.*, 10:5–12, 1976.

[26] A. Edelman. *Random Matrices.* unpublished course notes for MIT's class 18.325, 20xx.

[27] R. Franke. Scattered data interpolation: Tests of some methods. *Math. Comp.*, 38:181–200, 1982.

[28] R.J. Gardner. The Brunn–Minkowski inequality. *Bull. Amer. Math. Soc.*, 39:355–405, 2002.

[29] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[30] E. Hairer, Ch. Lubich, and G. Wanner. *Geometric Numerical Integration Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, Berlin, 2002.

[31] S.M. Hammel, J.A. Yorke, and C. Grebogi. Do numerical orbits of chaotic dynamical processes represent true orbits? *J. Complexity*, 3:136–145, 1987.

[32] R.L. Harder and R.N. Desmarais. Interpolation using surface splines. *J. Aircraft*, 9:189–191, 1972.

[33] G.H. Hardy. *A Mathematician's Apology.* Cambridge University Press, Cambridge, 1992.

[34] R.L. Hardy. Multiquadric equations of tomography and other irregular surfaces. *J. Geophys. Res.*, 76:1905–1915, 1971.

[35] S. Helgason. *Topics in Harmonic Analysis on Homogeneous Spaces.* Birkhäuser, Boston, 1981.

[36] D. Hestenes. *Clifford Algebra to Geometric Calculus. A Unified Language for Mathematics and Physics.* D. Riedel, 1984.

[37] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11:237–339, 2002.

[38] A. Hodges. *Alan Turing, The Enigma.* Burnet Books, London, 1983.

[39] E. Hubert. Factorization-free decomposition algorithms in differential algebra. *J. Symbolic Comput.*, 29:641–662, 2000.

[40] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.

[41] H.B. Keller. *Lectures on Numerical Methods in Bifurcation Problems.* Number 79 in Tata Institute of Fundamental Research Lectures on Mathematics and Physics. Springer-Verlag, Berlin, 1987.

[42] T.W. Körner. *Fourier Analysis.* Cambridge University Press, Cambridge, 1988.

[43] S. Levy, editor. *Flavors of Modern Geometry*, Cambridge, 1997. Cambridge University Press.

[44] T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In P.G. Ciarlet and F. Cucker, editors, *Handbook of Numerical Analysis*, volume ??, pages ???–??? North-Holland, Amsterdam, 2003.

[45] J.E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.

[46] R.I. McLachlan and G.R.W. Quispel. Six lectures on the geometric integration of ODEs. In R.A. DeVore, A. Iserles, and E. Sűli, editors, *Foundations of Computational Mathematics, Oxford 1999*, pages 155–210, Cambridge, 2001. Cambridge University Press.

[47] C.A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive functions. *Constr. Approx.*, 2:11–22, 1986.

[48] C.A. Micchelli. *Mathematical Aspects of Geometric Modeling*. SIAM, Philadelphia, 1995.

[49] V.D. Milman and G. Schechtman. *Asymptotic Theory of Finite-Dimensional Normed Spaces*. Lecture Notes in Mathematics 1200. Springer, Berlin, 1986.

[50] F. Natterer. Numerical methods in tomography. *Acta Numerica*, 8:107–142, 1999.

[51] B.N. Parlett. Some basic information on information-based complexity theory. *Bull. Amer. Math. Soc.*, 26:3–27, 1992.

[52] G. Sapiro. Introduction to partial differential equations and variational formulations in image processing. In P.G. Ciarlet and F. Cucker, editors, *Handbook of Numerical Analysis*, volume ??, pages ???–??? North-Holland, Amsterdam, 2003.

[53] I.J. Schoenberg. Remarks to Maurice Fréchet's article: 'Sur la definition d'une classe d'espace distanciés vectoriellement applicable sur l'espace d'Hilbert'. *Ann. of Math.*, 36:724–732, 1935.

[54] I.J. Schoenberg. On certain metric spaces arising from Euclidean space by a change of metric and their embedding in Hilbert space. *Ann. of Math.*, 38:787–793, 1937.

[55] I.J. Schoenberg. Metric spaces and completely monotone functions. *Ann. of Math.*, 39:811–841, 1938.

[56] S. Smale. Some remarks on the foundations of numerical analysis. *SIAM Rev.*, 32:211–220, 1990.

[57] S. Smale. Complexity theory and numerical analysis. *Acta Numerica*, 6:523–551, 1997.

[58] G.W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Num. Anal.*, 17:403–409, 1980.

[59] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[60] A.M. Stuart and A.R. Humphries. *Numerical Analysis of Dynamical Systems.* Cambridge University Press, Cambridge, 1996.

[61] V.N. Temlyakov. Nonlinear methods of approximation. *Found. Comp. Maths*, 2, 2002. To appear.

[62] A. Terras. *Fourier Analysis on Finite Groups and Applications.* Number 43 in London Mathematical Society Student Texts. Cambridge University Press, Cambridge, 1999.

[63] E. Tonti. Sulla struttura formale delle teorie fisiche (Italian). *Rend. Sem. Mat. Fis. Milano*, 46:163–257, 1976.

[64] J.F. Traub and H. Woźniakowski. Perspectives on information-based complexity. *Bull. Amer. Math. Soc*, 26:29–52, 1992.

[65] H.F. Trotter. Eigenvalue distributions of large Hermitian matrices; Wigner's semi-circle law and a theorem of Kac, Murdock and Szegő. *Adv. Math.*, 54:67–82, 1984.

[66] J.W. von Goethe. *Erotic Poems.* Oxford University Press, 1997. Translator D. Luke.

[67] J. von Neumann and I.J. Schoenberg. Fourier integrals and metric geometry. *Trans. Amer. Math. Soc.*, 50:226–251, 1941.

[68] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, New York, 1999.

[69] G.A. Watson. Choice of norms for data fitting and function approximation. *Acta Numerica*, 7:337–377, 1998.

[70] A.G. Werschulz. *The Computational Complexity of Differential and Integral Equations. An Information-Based Approach.* Oxford University Press, New York, 1991.

[71] J.H. Wilkinson. Error analysis of floating-point computation. *Numer. Math.*, 2:319–340, 1960.

[72] D. Williams. *Weighing the Odds.* Cambridge University Press, Cambridge, 2001.

[73] K.G. Wilson. Grand challenges to computational science. *Future Gen. Comput. Sys.*, 5:171–189, 1989.

[74] S. Winograd. On the multiplicative complexity of the discrete Fourier transform. *Adv. in Maths*, 32:83–117, 1979.

[75] M. Yamaguti and Y. Maeda. Chaos in finite difference scheme. In P.G. Ciarlet and F. Cucker, editors, *Handbook of Numerical Analysis*, volume ??, pages ???–??? North-Holland, Amsterdam, 2003.

[76] N. Young. *An Introduction to Hilbert Space.* Cambridge University Press, Cambridge, 1988.

[77] S.K. Zaremba, editor. *Applications of Number Theory to Numerical Analysis*, New York, 1972. Academic Press.

[78] D. Zeilberger. A fast algorithm for proving terminating hypergeometric series identities. *Discrete Maths*, 80:207–211, 1990.