ESSAY 13

Numerical Analysis and Its (Invisible?) Role in Mathematical Application

Anthony F. Peressini

The obtaining of numerical results in mathematical applications, the so-called "number crunching" is quite often taken for granted, both in "ordinary life" and even in relatively more sophisticated settings in science and engineering. We blithely expect our calculators and computers to produce numerical answers, flawlessly and unambiguously. Whatever mathematics is lurking behind those calculations is hidden, obscured, invisible. This invisible mathematics is known as numerical analysis.

Numerical analysis is not really invisible. Its relative lack of visibility is due largely to inattention. In contemporary science and mathematics, much of the work of numerical analysis escapes attention because it does its job so well. But its history is both visible and perhaps even prominent

in the development of modern mathematics. What is more—if one looks in the right places, it is still playing a developing role in burgeoning areas of science.

In this paper I examine numerical analysis—what precisely it is and why it is important. I begin by presenting a selective conceptual reconstruction of one suggestive line in its historical development. I then expand my focus to a general account of what numerical analysis consists today. I suggest a set of concerns and techniques that are characteristic of numerical analysis. Finally I distinguish three different roles numerical analysis plays in application and illustrate how these distinct roles shed light on its nature and varying visibility.

Numerical analysis studies the means of obtaining numerical results for mathematical expressions: Numerical methods are used ingeniously and indirectly to arrive at the actual numbers implied by a formal solution. While contemporary numerical analysis is inextricably wed to the computer, numerical analysis has its origins in the early 16^{th} century's development of tables of logarithms (Goldstine [1977]), which saw mathematicians working to develop the *means* of generating these tables in increasingly clever and indirect ways in order to avoid cumbersome hand calculations. Logarithms were being used as tools to perform what would have been prohibitively difficult calculations—especially in navigation and calculating interest. While it is Napier and Burgi who typically get the lion share of credit for developing logarithms, it was Henry Briggs (1556–1630) who in extending their work, discovered numerical techniques that persist today.

Practically any use of mathematics beyond counting involves numerical analysis. This mathematics is often hidden because it is embedded in the calculators, processors, and other computing machinery used to generate the numerical solutions.[1] Although the application of numerical analysis is rarely seen, it is essential to prediction, confirmation, and even explanation in science, as well as in any other setting in which the mathematics involved goes beyond arithmetic—including a calcula-

[1] The notion of a "numerical solution" is importantly ambiguous. As I will develop below it can indicate either the numerical results obtained from the formal (explicit and exact) solution to a problem or the numerical results obtained to a problem without a formal solution.

tor and its square root function, the calculation of which bear not the slightest resemblance to the hand algorithm taught in grade school.

Numerical analysis involves using mathematical techniques to generate numerical solutions to mathematical expressions. If, for example, you used math to determine that you need to build a wall that is $\sqrt{2}$ meters long, you probably used trigonometry and/or calculus. If you go on to actually find the value of $\sqrt{2}$ to nine decimal places, then you used numerical analysis. The modern theory of numerical analysis includes developing numerical methods, but it also seeks to ground the techniques in theory to establish why they work, how accurately, how generally, and how robustly. It works to develop the means of generating numerical solutions not just to simple expressions like $\sqrt{2}$, but also to problems on the cutting edge of science and engineering.

1 A Conceptual/Historical Example

In order to develop a better sense of what numerical analysis is (or better yet what numerical analysis does), let me describe a simple example. Consider how square roots are actually calculated.

Naturally, if we need to know a (non-rational) square root, we go to a calculator, but how do calculators (and/or computers) actually solve the problem? A first guess might be that they do so by imitating the old fashion long-division-like algorithm, but they don't. An historical first stop for square root calculations is Babylonia. The Babylonians (circa 1700 BCE) discovered a method for successively approximating the square root of a number Q that involves only division and averaging. The method goes as follows:

(1) Take an initial guess at $\sqrt{Q}$, call it A.

(2) Compute a better guess, B, using the formula, $B = \frac{A+Q/A}{2}$.

(3) Next iterate your refined guess B through the same formula in place of A to get your 3rd guess, $C = \frac{B+Q/B}{2}$.

(4) Repeat until desired accuracy is obtained.

Iteration	Value
1	1.500000000
2	1.416666667
3	1.414215686
4	1.414213562
5	1.414213562
6	1.414213562

Table 1: Babylonian Method

Table 1 shows six iterations of the method for $\sqrt{2}$ and initial guess = 1. You'll notice that the result converges to nine decimal places by the 5^{th} iteration and that the results match the actual value to nine decimal places by the 4^{th} iteration.

The method of successively approximating a value by repeating the same calculation on the result from the last time is called an iterative method. Typically successive approximations are denoted by subscripts (instead of successive letters of the alphabet) so that the third guess is A_3 instead of C and in general the n^{th} guess A_n, thus the formula can be written as $A_{n+1} = \frac{A_n + 2/A_n}{2}$. With respect to iterative methods important questions arise like the following:

(1) For what values of the initial guess will the approximations converge?

(2) How many iterations are required for a desired accuracy?

(3) For what (kind of) roots will the approximations converge?

These are the questions that gave rise to the practice of numerical analysis.

Before moving on to elaborate on the methods of numerical analysis, let me set the stage a bit more by tracing the development of a few more numerical means of calculating square roots.

Notice that finding the value of $\sqrt{2}$ is equivalent to finding the value of x where the function $f(x) = x^2 - 2$ is zero, i.e., solving $x^2 - 2 = 0$. Thus a

Iteration	Low Approx	High Approx
1	1.0	2.0
2	1.0	1.500000000
3	1.250000000	1.500000000
4	1.375000000	1.500000000
5	1.375000000	1.437500000
6	1.406250000	1.437500000
7	1.406250000	1.421875000
8	1.414062500	1.421875000
9	1.414062500	1.417968750
10	1.414062500	1.416015625
11	1.414062500	1.415039062
12	1.414062500	1.414550781

Table 2: Bisection Algorthm

function evaluation problem becomes the problem of finding the zeros of a related function. This illustrates an important technique in numerical analysis, namely, transforming one numerical problem into another in order to bring more powerful techniques to bear. And indeed, casting the problem in terms of finding zeros opens up a wealth of theory and technique.

One of the simplest approaches to finding a zero is known as the bisection algorithm. It is theoretically based in the Intermediate Value Theorem (due to Bolzano, early 19^{th} century). This method begins with two initial guesses (one high and one low) and then splits the difference by squaring the value in the middle and if it is higher than the number whose square root we are seeking, we then keep the midpoint as the new high value for the next iteration. Alternatively, if the midpoint squared is less than the number whose root we seek, then the midpoint becomes the new low value. This is then repeated until desired accuracy is obtained. Table 2 shows the first twelve iterations. Notice how much more slowly it converges—only three decimal places after twelve

iterations, compared to nine decimal places after four iterations with the Babylonian Method. (I will return to why this is the case below.) Despite being slower to converge, the bisection algorithm does have a real advantage: It is perfectly stable and is guaranteed to converge.

As a third method, consider a variation of the bisection algorithm—the *Regula Falsi* Method. This method, rather than *bisecting* the interval each time, splits the interval where a "false line" from the function evaluated at the end points of the interval hits the x-axis. As you can see from Figure 1, this way of choosing the next interval, by using more information about the shape of the function, generates intervals that converge more quickly to the actual answer. (The formula for the split point is $c = \frac{ab+2}{a+b}$.) See Table 3 for the first twelve iterations.

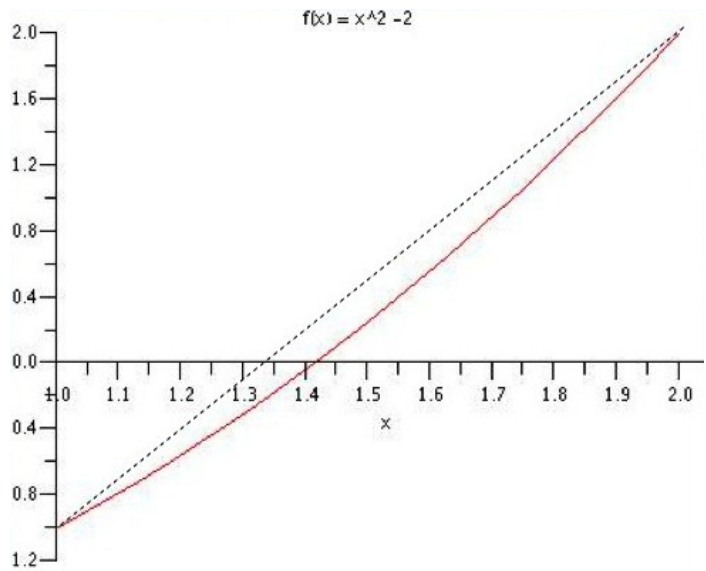

Figure 1: *Regula Falsi* Method

This method is an improvement over the bisection algorithm. While this method doesn't converge qualitatively faster (both methods have what numerical analysts call *linear convergence*), it does converge quan-

Iteration	Low Approx	High Approx
1	1.00000000	2.0
2	1.33333333	2.0
3	1.40000000	2.0
4	1.411764706	2.0
5	1.413793103	2.0
6	1.414141414	2.0
7	1.414201183	2.0
8	1.414211438	2.0
9	1.414213198	2.0
10	1.414213500	2.0
11	1.414213552	2.0
12	1.414213561	2.0

Table 3: *Regula Falsi* Method

titatively faster—by a factor of about four.

An important thing to notice about the *Regula Falsi* method is the way that it refines the general method of the bisection algorithm, taking into account further information and thereby increasing the rate of convergence. This is yet another technique that is typical of numerical analysis.

As a final method of calculating $\sqrt{2}$, let's consider one better than the *Regula Falsi*. While *Regula Falsi* used a false line to narrow the interval, with differential calculus we can actually determine the line tangent to the function at an initial guess and use its x-intercept as the next approximation. As your geometric intuitions might tell you, the x-intercept of such a tangent line closes in on the zero even faster than the false line of *Regula Falsi*. The method is known as Newton–Raphson Method (see Figure 2).

The formula for the Newton–Raphson Method is $A_{n+1} = A_n - \frac{f(A_n)}{f'(A_n)}$. It turns out that the Newton–Raphson method converges *qualitatively* faster than the Bisection or *Regula Falsi* Methods. It has what numer-

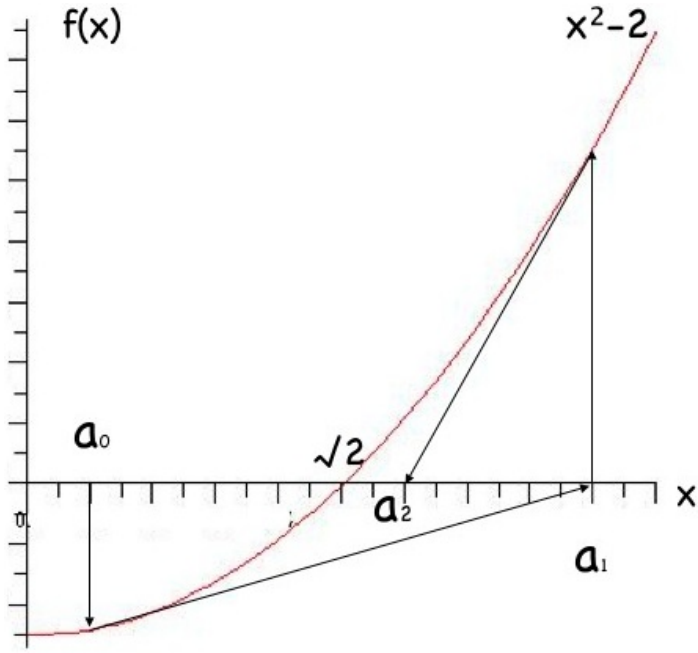

Figure 2: Newton–Raphson Method

ical analysts call *quadratic* convergence for problems like square roots.[2] Notice also that if we substitute the transformed square root function, $x^2 - 2$ in for f, along with its first derivative $2x$, and do a little simplifying, lo and behold, we get the Babylonian Method! It turns out that the Babylonian Method is a special case of Newton–Raphson Method . . . and this explains why it converged so much faster than the bisection or *Regula Falsi* methods. It turns out that the Babylonians discovered a method that it took Newton to generalize by grounding it in the calculus and then only later, modern numerical analysts to explain its rapid

[2]Quadratic convergence is essentially an order of magnitude faster than linear convergence. Specifically, if the limit of the ratio of the difference between the converging function and its limit to x^2 is a positive number, then the convergence is quadratic. It is linear if the above obtains with an x instead of x^2.

rate of convergence.

Despite its speedy convergence, the Newton–Raphson Method has a drawback. While numerical analysts have proven that for a sufficiently small interval around a simple zero like $\sqrt{2}$, the method is guaranteed to converge, if an initial guess is not within that critical interval, then strange non-convergent things can happen. Additionally, if the first derivative shares the zero then the method will have additional problems. So the Newton–Raphson Method is somewhat lacking in numerical stability. In actual practice, canned programs for finding a zero use a combination of methods—usually the Newton–Raphson Method and bisection method—for just this reason.

Now that we are back to calculators and zeros—is this then how calculators find square roots? No. We have to go all the way back to one of the people I mentioned above: Henry Briggs and logarithms.

Briggs, in developing his log tables in the 17^{th} century, developed all sorts of ingenious ways to simplify the laborious calculating of huge tables of logarithms. The reason logarithms are useful (and the reason Briggs, Napier, Burgi, and others sought to develop tables of them) is that they simplify calculations by turning multiplication/division into addition/subtraction and exponentiation into multiplication.

The way a calculator comes up with $\sqrt{2}$ is by using the following identities.

$$\sqrt{2} = 2^{1/2} = e^{(1/2)\cdot\log(2)} \tag{1}$$

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots \quad [-1 < x < 1] \tag{2}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \tag{3}$$

The calculator calculates $\log(2)$ to machine precision by adding the first few terms of the log series (2). In accordance with (1), this value is then multiplied by $1/2$ and the result is exponentiated by adding up the first few terms of the anti-log series (3), again to machine precision.

It is interesting to note that the series approach to logarithms does actually have its origin in Briggs' work. In his ingenious method for computing logarithm tables, Briggs needed to repeatedly calculate square roots and in developing a workable way to do this he discovered the bi-

nomial theorem for $(1+x)^{1/2}$, which is the first use of a series calculation of a noninteger exponent. So ironically, Briggs used a series approach in computing his logs, but he used it not to calculate the logs directly, but rather to calculate square roots, which he then used to calculate logarithms.

2 The Field of Numerical Analysis

Having discussed what animates a small corner of numerical analysis, how it developed, and some of its activities, methods, and concepts, I now consider how the field of numerical analysis unfolds into areas beyond calculating square roots.[3]

A standard introduction to numerical analysis will begin with methods of solving equations in one variable—that is, finding the x's, such that $f(x) = 0$. Now as we've seen, the problem of calculating a square root can be transformed into just such a problem, and as you may imagine, so can many others. The methods of solving these equations of one variable get more and more specialized and powerful, and typically the more narrowly one focuses on such a problem, the more efficient the methods get. For example, if one assumes that the function is a polynomial, then the methods available are extremely efficient and stable.

Another cottage industry in numerical analysis is the problem of approximating a function for which one knows only a few of its values. Typically one does this by determining a "well-behaved" function (like a polynomial) that hits all the known data points and then uses this function to estimate unknown data points. This area is known as polynomial approximation and interpolation. Again it generalizes to more general approximation techniques, including using least square methods, orthogonal polynomials, rational functions, and trigonometric polynomials by way of Fourier transformations.

The problems of integration and differentiation are also mainstays of numerical analysis. For any of the real world problems that are solved by calculus (characterized by the operations of differentiation and in-

[3]It should be noted that as I move on to contemporary numerical analysis an implicit shift takes place from "pencil and paper" calculations to the electronic computer. While considering this transition is an important one in its own right, I set it aside for the most part in this paper, though I return briefly to it in the final section.

tegration), in order to actually use the solution, numerical results will be needed. Again such methods begin with well-behaved functions of one variable and branch out from there to very complicated functions of multiple variables. In the case of both, the crucial tools are classes of "well-behaved" functions—like the polynomials. It can be proven that for any continuous function of interest, a polynomial can be found that is arbitrarily close to the function of interest at *every* point in a closed interval. So to get a numerical result of a given accuracy to a definite integral or a derivative of a function at a point, one generates an appropriately close polynomial and then treats it as the function of interest to differentiate or integrate. Numerical analysis is further used to prove results concerning error bounds on the difference between the actual and approximate values. Another important area of inquiry concerns how to best choose the original data points by which to generate the approximating function. *Adaptive quadrature* methods chose the location and density of such node points by analyzing the functional variation and using more nodes in regions of higher variation.

One of the most utilized areas in numerical analysis is the solution of systems of linear equations. Equations in multiple unknowns (like on the left below) are cast into matrices (like on the right):

$$\begin{aligned} x + 3y - 2z &= 12 \\ 2x + y - z &= 3 \\ -x - 3y + z &= 1 \end{aligned} \quad \Longrightarrow \quad \left[\begin{array}{ccc|c} 1 & 3 & -2 & 12 \\ 2 & 1 & -1 & 3 \\ -1 & -3 & 1 & 1 \end{array}\right]$$

They are solved by diagonalizing the matrix using row multipliers and row additions to replace existing rows until we have the following diagonal matrix:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

This indicates that the solutions to the original equations is $x = -2$, $y = 1$, and $z = 3$. The number of arithmetic operations required by this method (Gaussian Elimination) for a given matrix size is given in Table 4 (Burden, Faires, and Reynolds [1981]: 272).

Matrix size $(n \cdot n)$	**Multiplications/ Divisions**	**Additions/ Subtractions**
3	17	11
10	430	375
50	44,150	42,875
100	343,300	338,250

Table 4: Gaussian Elimination Operations

Obviously this ends up requiring a lot of arithmetic—and the problem is that each operation introduces rounding error, which can under certain conditions practically swamp out the solution, rendering it worthless. Methods for solving many other kinds of problems require the solution of very large (hundreds of thousands) systems of equations, hence means of reducing the storage space and operations required are as important as accuracy. Numerical analysts continue to develop more efficient methods requiring less storage and fewer operations. Additionally they have developed means of characterizing the matrix with respect to how it will respond to Gaussian Elimination Methods. Depending on its "condition," supplementary matrix treatments may be used, such as maximal or scaled pivoting to reduce rounding error, which is highly dependent on the relative size of the matrix elements and their position in the matrix.

Important results in numerical matrix theory have also been developed for matrices with special structure (e.g., upper triangular, tridiagonal, Hessenberg, diagonally dominant, positive definite, and sparse). In these cases, theorems can be proven to show that variations of Gaussian elimination are stable and will not require row interchanges because of small or zero elements in key positions. One of the more important developments in numerical linear equations is *iterative techniques*, much like the ones for the square root function considered above. These methods are particularly important for sparse and other specially structured matrices that are generated by methods of solving ordinary and partial differential equations. (See Stewart [1973] for details.)

Differential equations are one of the most ubiquitous mathematical

tools in all of applied mathematics. There are ordinary and partial differential equations and among these there are initial value problems and boundary value problems. And then there are systems of each of these kinds as well. These equations arise both in describing the fundamental laws of nature and also in very application-oriented engineering contexts. I won't go any further into these here, but this area is one of the more lively areas in numerical analysis today. Many of the methods of numerically solving differential equations involve solving massive systems of linear equations and/or numerical integration and differentiation. Hence again, we see numerical analysis transforming one kind of problem into another more numerically tractable one.

Having presented the major problems of numerical analysis let me now summarize a bit. Numerical analysis involves problems like approximating a function using "simpler" functions, calculating numerical derivatives, solving initial value and boundary value problems for differential and partial differential equations, finding values for definite integrals, solving systems of linear equations, finding fixed points and characteristic values for functions, and fitting curves to data sets. Numerical techniques are often *iterative.* An iterative technique involves performing a fixed series of arithmetic calculations, called iteration, on an initial approximation. As the iterations progress, the approximation gets "nearer" the actual solution. Theorems are then sought to establish results such as

- the conditions under which the series of iterations will converge for particular initial values;
- restrictions on the initial value;
- the rate at which the series will converge;
- the accuracy of the approximation;
- the sensitivity to round-off error; and
- the average and maximum number of iterations required for a given accuracy.

It is worth pointing out that while my distilled account of the field of numerical analysis might suggest that its major problems have been

worked out, such is not the case. In each of the areas I mentioned (and others I did not) work continues on particularly recalcitrant classes of problems. This work largely consists of discovering and characterizing such numerically ill-conditioned problems and developing more subtle, efficient, and effective means of obtaining results.

3 Numerical Analysis as Applied Mathematics

In the times of Briggs, Newton, and Bolzano (to name a few) the relationship between the formal analytic problems and the numerical were still being worked out. In the wonderful stewpot that was mathematics from the 16^{th} century on, we see the concurrent development of formal notation, logical foundations, analysis, and the topology of the real numbers, which are precisely the theoretical underpinnings of contemporary numerical analysis. In this period numerical concerns were naturally quite visible.

I'd like to suggest that some of this persists today. Let me offer the following three rough and ready ways in which numerical analysis functions in scientific application to draw this out.

Instrumental Exact solutions are available, e.g., finding the zeros of a complicated polynomial.

Essential No exact solution is available (even in principle), e.g., certain differential equations.

Explanatory/exploratory The problem is sufficiently complicated that the relationship between the formal mathematical and the numerical is not understood, e.g., general relativity theory.

The first role is instrumental in that a closed exact solution is known and the numerical methods are used only to "do the calculations." Imagine an engineer needing to find the zeros of a complicated cubic polynomial—messy work, but work that could be done without numerical analysis. Things get a bit more fuzzy when one has a simple first-order homogenous differential equation that has a *trigonometric* solution: Obtaining the exact solution requires no numerical analysis, but evaluating the trigonometric solution at values of interest does.

But more often than not, even in well-understood settings, exact solutions are unavailable. In this case the role of numerical analysis is essential.[4] Consider something as straightforward as a second-order differential equation with initial values (a so called "initial value problem"). Such a problem might arise in the modeling of the efficacy of the brakes of a car, taking into account wind resistance and friction. When exact solutions are not available (as they mostly are not), then the only way to

(1) confirm that the differential equation models the braking situation and

(2) make use of the differential equation to determine a breaking distance

is by way of numerical analysis and its highly theoretical grounding in pure mathematics. Nonetheless, the numerical analysis required to perform (1) and (2) is understood to the point that it now covered in "first courses" in numerical analysis. So while essential, these applications of numerical analysis are so routine as to be "invisible."

The third explanatory/exploratory way is important here because numerical analysis applications of this kind are clearly *not invisible* in any sense. These are examples in which the applied mathematical model and its analytic solutions are sufficiently complicated that the relationship between the mathematical model and numerical results are not understood very well at all. Examples like this seem to be present in general relativity, quantum mechanics, and even some classical systems that are chaotic or Brownian.

For example, in the study of the dynamics of deterministic systems using differential equations, the numerical methods are well understood. It turns out, however, that most real phenomena have a stochastic or random component; the mathematical tools used to describe such stochastic dynamics are stochastic differential equations. Many of the

[4] I intentionally didn't use the term "indispensable." There are philosophers who think that indispensability is the mark of something metaphysically significant for mathematics. I don't, but for this discussion it doesn't matter. See Peressini [1997] and [2003] for indispensability issues.

numerical methods developed to handle the dynamics of deterministic systems with analytic solutions have been *applied* to systems with stochastic solutions, but relatively little is yet known of the behavior (how stable, how convergent, etc.) of such methods in stochastic settings. In general, whenever numerical methods developed on classical (Newtonian) physics are used outside their theoretically understood domain of application—with theory struggling to catch up—we find numerical analysis playing this explanatory/exploratory role. (For specific examples see Montaldi [2000]; Sprekels [2003]; and Burrage, Lenane, and Lythe [2007].)

The explanatory/exploratory use of numerical analysis is even more pronounced in general relativity. Within the study of general relativity a focus called *numerical relativity* has developed. (For specifics and discussion see Arnold, Ashtekar, and Laguna [2002]; Gentle [2002]; and Carlson [2006].) The study of gravity waves involves looking at massive objects (black holes or neutron stars) in very close proximity, since it is only under such conditions that waves large enough to be detected will be produced. While the formal solutions to such problems are straightforward, obtaining numerical results for such a system is incredibly difficult due to a variety of factors that are not fully understood. The numerical analysis being done in this case is unusual in that it is not being approached by bracketing the problem and then applying the pure analytic theory of numerical analysis. Rather, (i) the science is being reconsidered in the hope of discovering some physically motivated insight or simplification or reduction that will render the numerical problem tractable, and (ii) the application of the (unsuccessful) numerical methods to the actual problem *are being simulated* on related but simpler and better understood problems in order to gain insight into why the method goes fails on the actual problem.

The explanatory/exploratory role of numerical analysis as in the above examples is characterized by the fact that its presence is asserted (if you will) to such an extent that it has its own rich relationship with the two traditional players: the formal applied mathematics and the physical world. In the more familiar roles (instrumental and essential), we see the reciprocal movement between the formal theory and the physical world in which theory is confirmed by its prediction/explanation of physical phenomena and the further enriching,

expanding, and revising of the formal theory inspired/necessitated by additions and complications in the physical phenomena, . . . , and so on back and forth. But in settings in which numerical analysis is playing an explanatory/exploratory role, the relationship between the formal theory and the physical world is richly mediated by the numerical analysis. In such cases the formal theory's numerical behavior is not fully understood. It requires the explanatory and exploratory machinery of numerical analysis to shed light on it in order for the model to make tractable predictions. Similarly, numerical insights can motivate mathematically equivalent ways of decomposing or understanding the formal model that can in turn shed light on the physical system itself.

It is worth mentioning here that the role of the computer in the development of numerical analysis should not be understated, but neither should it be taken as single-handedly making numerical analysis possible. As we have seen, remarkable developments in numerical analysis happened before the advent of computing machinery and indeed were necessary precisely because certain calculations were cumbersome to the point of being impractical in the absence of computers. So while numerical analysis itself is not essentially tied to computers, nonetheless, computers are responsible for the relatively recent emergence of the explanatory/exploratory role of numerical analysis. In particular, the very possibility of numerically exploring these contemporary models of physical systems takes for granted the availability of computers.

What is going on in the explanatory/exploratory category resembles what was going on in the more rough and tumble days of the 16^{th} and 17^{th} centuries, where the path from mathematical expression to numerical results was largely uncharted. It is informative to consider why it might be that numerical analysis is particularly visible in these two rather distinct settings.

In the period before the rigorous formalization of mathematics, perhaps as far back as Babylonia through the early 17^{th} century (the cut off here is somewhat arbitrary), it is rather anachronistic to interpret any of the work being done as *numerical analysis.* To be sure much of this early work was foundational for numerical analysis but only because it was foundational for the rigorous formal axiomatic mathematics we have today. Prior to the formal development of analysis, while one can point out mathematical work on numerical techniques, properly speak-

ing, no numerical *analysis* was being done since the means of comparing and classifying convergence rates and error analysis were not available. During this period (and ensuing periods leading to modern numerical analysis) the *visibility* of work on numerical methods was pervasive due to its being explored concurrently with the development of analysis in general.

In contrast, contemporary numerical analysis' visibility is limited largely (as we have seen) to its explanatory/exploratory role in scientific application. In the instrumental and even essential roles characterized above, its invisibility is due to the numerical tractability of the theory employing numerical analysis. In such settings, the numerical landscape is charted and it can fade invisibly into the background. In other application settings, ones that are numerically problematic, numerical analysis plays an active and visible role.

This difference in the visibility and role of contemporary numerical work in mathematics grew out of two distinct (though perhaps not independent) developments: the modern formalization of analysis and the advent of the computer. Once in place, these two developments allowed the three roles of numerical analysis described above to coalesce into more or less the form we find them in now.

Acknowledgments

I'd like to thank an anonymous referee for many helpful comments and Conor Peressini for technical assistance.

Bibliography

Arnold, D.; A. Ashtekar; and P. Laguna [2002]. "Proceedings of The Institute for Mathematics and its Applications Workshop on Numerical Relativity." URL: http://www.ima.umn.edu/nr/ (visited July 2007).

Burden, R.; J. Faires; and A. Reynolds [1981]. *Numerical Analysis*. 2nd ed. Boston: Prindle, Weber & Schmidt.

Burrage, K.; I. Lenane; and G. Lythe [2007]. "Numerical methods for second-order stochastic differential equations." *SIAM Journal on Scientific Computing* **29**. 245–64.

Carlson, E. [2006]. "Numerical General Relativity." *Eric Carlson Home Page*. URL: http://www.wfu.edu/~ecarlson/numerical.html (visited July 2007).

Gentle, P. [2002]. "Regge Calculus: A Unique Tool for Numerical Relativity." *General Relativity and Gravitation* **34**. 1701–18.

Goldstine, H. [1977]. *A History of Numerical Analysis from the 16th Century through the 19th Century*. New York: Springer Verlag.

Montaldi, J., ed. [2000]. "Numerical Analysis." *Mechanics and Symmetry in Europe Home Page*. URL: http://www.maths.manchester.ac.uk/~jm/MASIE/projects/sec2.html (visited July 2007).

Peressini, A. [1997]. "Troubles with Indispensability: Applying Pure Mathematics in Physical Theory." *Philosophia Mathematica* **5** (3). 210–27.

— [2003]. "Mark Colyvan's The Indispensability of Mathematics." *Philosophia Mathematica* **11** (3). 208–33.

Sprekels, J., ed. [2003]. "Numberical analysis of complex stochastic models." *Weierstrass Institute for Applied Analysis and Stochastics Home Page*. URL: http://www.wias-berlin.de/publications/annual_reports/2002/node77.html (visited July 2007).

Stewart, G. [1973]. *Introduction to Matrix Computations*. Orlando, FL: Academic Press.