PROGRESS IN NUMERICAL ANALYSIS

by

Beresford Parlett

Memorandum No. UCB/ERL M77/26

13 April 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

PROGRESS IN NUMERICAL ANALYSIS[†]

Beresford Parlett

Departments of Mathematics and Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley

Abstract

Developments in numerical analysis fall into two separate categories. The first comprises work on problems which are unsolved in the sense that either no feasible methods are available or else there is no reliable analysis for the methods which are in use. The second category comprises work on solved problems and its aim is to remove the human user from the solution process, in so far as this is possible, and also to improve efficiency in regard to other factors such as execution time, storage requirements or length of code.

Shortly after the introduction of modern digital computers and high level programming languages most of numerical analysis fell into the unsolved category. With every success in this category the second one has grown -- and vice versa. In order to judge properly the value of the multifarious research activities in numerical analysis it is important to grasp the evolution of this sprawling empire.

In this essay we point out some muddles caused by not discriminating between the two categories, we present a simple picture of the evolution of scientific computation, and we find how hard it is to dismiss any area as having no future promise.

Key Words: scientific computation, numerical analysis

[†] A talk presented at the Symposium honoring the 30th anniversay of the Office of Naval Research held during the 1976 SIAM meeting in Atlanta, GA. Research sponsored by Office of Naval Research Contract N00014-76-C-0013.

CONTENTS

Page

1. THE TERRITORIAL FALLACY 3

The Territorial Model of Progress 3

An Example of Extroverted Numerical Analysis 4

Defects in the Territorial Model 6

A. New Ways for Old Problems 6

B. Negative Results 7

C. Changing Vales 8

2. THE TOWER OF SCIENTIFIC COMPUTATION 8

Implications of the Tower's Growth 11

A. Hidden Computations 11

B. Reliability 11

C. Success Forfeits Attention 12

3. CANDIDATES FOR NEGLECT 13

A. Library Functions 13

B. Iterative Methods for Solving Ax = b 14

C. Small Matrix Computations 15

D. Nonstiff Systems of ODE's 18

4. THEORY -- NECESSITY OR LUXURY? 19

A. Examples 19

B. Paradigms 20

C. FEM 21

D. Ripe Areas 23

5. CONCLUSION 25

REFERENCES 26

1. THE TERRITORIAL FALLACY

The Territorial Model of Progress

Whether or not they readily acknowledge it, many people, including engineers, scientists, and scientific administrators, subscribe to the territorial model of progress. The model says that the experts in some field, nuclear reactor design say, can in principle describe all the problems which have ever been considered important in the field. The problems can be arranged according to their apparent difficulty and this arrangement constitutes the territory. At any given time there is a boundary between the solved and the unsolved problems. Progress is measured by the advance of this frontier.

In applying this structure to Numerical Analysis the territory consists not so much of problems but rather of mathematical tasks which a variety of users would like to delegate to the computer. A task is considered "solved" when there is known a reasonable way to accomplish it.

There is something attractive in this picture of progress. It seems to be quite satisfactory for those parts of numerical analysis in direct contact with users from other disciplines working on problems such as weather prediction, the responses of a bridge to an earthquake, the allocation of resources in an organization, or the design of a vehicle. Many, but not all of these applications require the approximate solution of special systems of partial differential equations. What is disconcerting about these areas is that the following boring pronouncement hangs like a fog over them. "Despite progress the crucial problems are still far from resolution. Further development is needed." This assessment can be put in a more positive light. Success with simple, often linear, approximations has whetted the appetites of users for more and more realistic models. Such is the open ended nature of science and technology.

An Example of Extroverted Numerical Analysis

In order to give some flavor of numerical analysis in these direct application areas we shall sketch briefly a single example.

In studying combustion the variables which describe the state of the gas are not globally smooth functions. They either start off with, or subsequently develop shocks and other discontinuities. The user wants to know the motion and interaction of these shocks in good detail. Now there are techniques for explicit shock fitting [Richtmyer and Morton, Ch. 13], but they are very difficult to apply and are not regarded as satisfactory for combustion problems.

A new type of explicit difference scheme has been proposed by Chorin [Chorin, 1976] and found to be very satisfactory in preliminary tests. The ideas behind the method are interesting because they were suggested by a formal proof of existence of weak solutions to strongly nonlinear hyperbolic systems [Glimm, 1965]. Thus can pure mathematics influence computation. However there is as yet no convergence theory for the new method because several hypotheses (e.g. nearly constant initial data) used in the existence proof are deliberately flouted in the numerical method. So there is scope for some good analysis.

At each time step t each dependent variable is regarded as a step function, i.e. a piecewise constant function over intervals of length Δx. Thus discontinuities can develop quite naturally.

Figure 1 illustrates how the solution vector is advanced through one time increment Δt. A single space variable x is used to simplify the explanation. Suppose that u is known at time t. Let x be a typical mesh point, the mid-point of a subinterval of length Δx on which the step function u is constant. Now

1. Solve for $v(\xi,t)$ the given differential equations with the simple initial data

$$v(\xi,0) = \begin{cases} u(x,t) & \text{for } \xi < 0 , \\ u(x+\Delta x,t) & \text{for } \xi \geq 0 . \end{cases}$$

This is called a Riemann problem and the details of its solution will not be discussed here.

2. Choose an equidistributed random number ρ in $[-\frac{1}{2},\frac{1}{2}]$ and set

$$u(x+\tfrac{1}{2}\Delta x, t+\tfrac{1}{2}\Delta t) = v(\rho\Delta x, \tfrac{1}{2}\Delta t) .$$

Proceed as in (1) and (2) for all mesh points x using the same ρ.

3. Solve for v the Riemann problem with initial data

$$v(\xi,0) = \begin{cases} u(x-\frac{1}{2}\Delta x, t+\frac{1}{2}\Delta t) & \text{for } \xi < 0 , \\ u(x+\frac{1}{2}\Delta x, t+\frac{1}{2}\Delta t) & \text{for } \xi \geq 0 . \end{cases}$$

4. Choose another equidistributed random variable σ in $[\frac{1}{2},\frac{1}{2}]$ and set

$$u(x,t+\Delta t) = v(\sigma\Delta x, \tfrac{1}{2}\Delta t) .$$

Proceed as in (3) and (4) for all mid mesh points using the same σ.

Of course, there is much more to the method than we have indicated. Appropriate modifications must be made at the boundary. For two space dimensions each time step is split into four alternating quarter steps (x,y,x,y), each of duration $\Delta t/2$, so that the x and y waves interact properly.

To the reader who is not well versed in this problem area we would say that this scheme is a radical departure from conventional difference schemes, especially in the use of random samples. Moreover even for flow problems with smooth solutions it is in general very difficult to concoct ways of replacing derivatives by differences in a way that works satisfactorily in practice. It is even harder to prove convergence of the few satisfactory techniques. So the domain of applicability is not clear yet. A challenging aspect of the Glimm-Chorin method is that it is not even consistent, a sacred property for linear problems.

Defects in the Territorial Model

There is an important part of Numerical Analysis which has grown steadily during the last 25 years and for which the territorial model provides a poor description of progress. We refer to the well developed areas in which there has been significant progress on rather basic tasks such as matrix calculations, quadrature, and minimization of functions. Dare we call it introverted numerical analysis?

The somewhat military picture of progress outlined initially has the following weaknesses.

A. New Ways for Old Problems

It fails to register the importance of finding a better way of performing an old task. Why recapture a town way behind your own front line? A dramatic example of this phenomenon was in the area of discrete Fourier analysis. The traditional way of computing the n Fourier coefficients belonging to a sequence of n function values uses a total of n^2 (complex) multiplications and that seemed to be a proper price to pay. However the discovery of a way to evaluate these coefficients using a mere $n \log_2 n$

multiplications [Cooley and Tukey, 1965] has encouraged the greater use of Fourier transforms and has revolutionized certain applications from signal processing to regular boundary value problems [Gentleman & Sande, 1966].

A less dramatic and generally unappreciated example was the introduction at last of reliable, efficient and accurate FORTRAN library functions, such as LOG, by the mid 1960's. On such foundations it is worthwhile to build up big, complex programs [Kuki, 1970].

Because most calculations are organized to exploit repetition there is great incentive to have rapid execution of all the tasks in the inner loops. As more ambitious projects are tackled so does the complexity of oft repeated tasks. For example a substantial linear programming problem may have to be solved at each step in some loop, or perhaps a two dimensional linear boundary value problem with different coefficients each time or a slightly different boundary each time.

B. Negative Results

There is no natural analogy in the territorial model for negative results in research. An example of this is the realization that the spectacularly good convergence rates for optimal alternating direction iterative methods could only be realized for very special problems and regions [Birkhoff & Varga, 1959].

Another example is Dahlquist's demonstration that A-stable methods which would be most attractive for stiff problems cannot have order of accuracy greater than two. The conclusion is that there is going to be no simple way of treating all stiff problems. It will be necessary to delineate such tasks in finer detail.

C. Changing Values

The model does not suggest the rapidity with which the importance of various computing tasks can change. In 1955 the automatic computation of polynomial zeros was considered an important task. By 1965 there were, at last, good routines available [B. Smith, 1967; Jenkins and Traub, 1970], but the demand had fallen off greatly. The polynomial zeros had turned into matrix eigenvalues. Going further in this direction it appears that some standard eigenvalue problems, $Ax = \lambda x$, actually come from generalized eigenvalue problems $By = \lambda Cy$ and are better posed in this form. However good methods for solving the latter problem did not become available until 1972 [Stewart & Moler, 1973], and the importance of this formulation is still not as widely appreciated as it should be.

Our conclusion is that an important part of our subject is inner directed (the numerical analyst's numerical analysis?) and the territorial model is quite misleading for understanding both its value and its development. An analogy with automobile production suggests itself. Electric cars, steam driven cars, emission-clean cars all represent external developments while electronic ignitions, fuel injection, and disk brakes represent interval developments. The latter perform tasks for which there existed satisfactory devices. Who can say what facets of the automobile need no further improvement?

2. THE TOWER OF SCIENTIFIC COMPUTATION

The principle reason why the observations made in Section 1 are not common knowledge is that when automatic digital computers first came into general use, about 1950, virtually all of Numerical Analysis was outer directed. During the last 25 years the methods used for certain tasks have disappeared from many a user's sight, having passed to the inner directed category. Thus the latter branch has quietly grown alongside the former and few people have noticed the change.

Figure 2, the tower of scientific computation, gives a picture of the structure on which so many applications depend. For various reasons the bottom layers, the hardware of the computer, the algorithms for actually performing + - × /, and the high level programming languages are not considered part of numerical analysis but rather of electrical engineering and computer science.

Our subject begins with the basic functions, such as square root, logarithm and sine, thoughtfully provided by the computer manufacturer. Creating such programs is a small, very specialized activity. Surprisingly it took nearly 15 years to produce reliable algorithms nicely tailored to each computer, the better to satisfy the conflicting demands of economy (in both time and space) and of accuracy. It is characteristic of this level that the domains of all these functions are clear and easy to describe. It appears that the manufacturers of hand held calculators have taken little note of this work. Their management's judgment that the problems are routine may compel this industry to repeat that learning process [Kahan and Parlett, 1976].

During the 1950's the tasks at the next level (No. 3) of the tower were on display in the sense that users came to computing centers with functions to be integrated, with big sets of 20 linear equations to be solved for the 20 unknowns, with systems of ordinary differential equations, and so on. Moreover they wanted to see the solutions. At this level the problems can sometimes be ill-posed and the numerical methods are sufficiently complicated that it is not obvious when they will fail. Nevertheless twenty-five years work on these standard problems has produced algorithms which enjoy a high degree of reliability and efficiency. In 1956 would any one have guessed that there would emerge a single method (the QR algorithm) which could calculate very rapidly the eigenvalues of all square matrices of orders up to 100? No one has been able to prove that this is the case, even ignoring roundoff errors, but no cases of failure have been reported.

There are now available, quite generally, program libraries whose programs are the result of the collaboration of the experts in the field. The Argonne National Laboratory has undertaken the testing, documentation, and dissemination of four major collections:

EISPACK (for most eigenvalue/eigenvector computations)

FUNPACK (for the evaluation of special functions)

LINPACK (for most linear equations and linear least squares problems)

MINPACK (for minimizing functions of several variables, with and without constraints)

Similar work is being undertaken in Britain by the Numerical Algorithms Group (NAG). In the private business sector there are libraries catering to a bewildering variety of well specified computations in statistics, smoothing, approximation by splines, standard matrix tasks, and so on. The best known are IMSL (International Mathematics and Statistics Libraries, Inc.) and SSP (IBM Scientific Subroutine Package).

The next level in the tower (No. 4) has developed only during the last decade. It comprises big programs which aim to be portable (directly transferable from one system to another) and to solve real engineering problems. The pioneers here were the civil engineers who had developed the Finite Element Method (usually designated by the acronym FEM) for analyzing the response of complicated structures to various loads and forces. Such programs have between 10,000 and 100,000 Fortran statements. Also at this level belong tasks of computing with large, sparse matrices. Notice that the user with a problem is directly involved at this level. He manages the package but does not write the program.

The top level is reserved for the enormous codes that have grown up at certain institutions for local problems in nuclear reactor design, curcuit design, satellite tracking, magneto-hydrodynamic calculations and such like. These massive programs are not designed to be shared with other installations. One hears that sometimes no one person understands the codes and one wonders how often such codes produce plausible but completely erroneous output.

Of course particular applications are not obliged to rest on the top level. The key fact is that more and more work is based on Level 3 rather than Level 2 and this movement will continue.

Implications of the Tower's Growth

A. Hidden Computations

These days most well specified computations are HIDDEN. This means that the human user sees neither the data nor the output. In a big calculation the data for a subtask (a Fourier Transformation, perhaps) will be generated by some program and the results promptly used by another. This is characteristic of introverted Numerical Analysis. It is the passing of Level 3 out of the user's sight which has made progress on this level intractable by the territorial model. The growth of layers on the tower has, as it were, added an extra dimension to Numerical Analysis.

B. Reliability

Algorithms for hidden computations need to be much more reliable than those whose results will be seen by a human eye. Execution time seems to be less crucial but both reliability and efficiency are wanted. To what extent, in each case, can we have both? That is an interesting question. The difficulties are suggested by pondering the following two desiderata.

P1. A program may fail but should not lie.

P2. A program should expend only a negligible amount of time, space, or code checking for rare eventualities.

A detailed understanding of the problem space and of the nature of an algorithm are needed to produce a happy match. Insufficient homage is paid to failure. The only honorable response to an ill-posed problem is failure to produce an answer. Yet acceptance of this burden raises the problem of discriminating between those computations which deserve no solution and those which are not amenable to the algorithm or algorithms embodied by the program.

The simplest way to sense the importance of reliability in the foundations of a computation is to imagine the effect on the programs at Level 3 in EISPACK, LINPACK, or on a good Ordinary Differential Equations solver, of an arithmetic unit in which one multiplication in 10,000 produces a totally false but reasonable result. Scientific computation would not be halted but it would become messier, more cumbersome, and harder to analyze.

C. Success Forfeits Attention

There is more to scientific computation than numerical methods. Data management and user interfaces are examples. Moreover success in the development of numerical methods serves to shift the bottleneck in complex calculations to these other factors. Thus the importance of the numerical solution sometimes appears to diminish according to several objective measures of effort. This is a simple fact of life and reminds us that the highest form of art is the art which conceals itself! Here is an example. The idea of discretization is very old and a sceptical engineer might wonder how much he has been helped by twenty-five years of fussing with this idea.

In 1976 Rice [Rice, 1976] made a study, for four boundary value problems, of the decrease in solution time which can be attributed solely to improvements in numerical methods. His estimates are encouraging, to say the least. In the tables below we show some of his figures for two of the problems.

Task No. 1: Solve Poisson's Equation $\Delta u = f$ on the unit cube with Dirichlet boundary conditions to an accuracy of 0.1%.

Table 1 goes here

Task No. 2: Solve $Lu = f$ in D with an accuracy of 0.1% where L is a 2nd order, non-separable, elliptic operator with variable coefficients and D is a plane simple domain with one or more curved boundaries.

Table 2 goes here

An important part of this overwhelming improvement in efficiency is the result of attention to problems which were already "solved."

3. CANDIDATES FOR NEGLECT

Our object here is to show that it is not easy to write off any area of numerical analysis as complete.

A. Library Functions

This is almost a counterexample to our assertion. Today the elementary functions are evaluated to working precision (almost) and, of more importance, properties such as monotonicity or symmetry are preserved as far as is possible. What more is there to be done? The advent of hand held calculators and microcomputers should revive interest in new algorithms which have minimal storage requirements.

B. Iterative Methods for Solving Ax = b

This was the most fashionable research topic from 1950-1965. Has the point of diminishing returns been reached? Here is a quotation from a sophisticated user. Italics are mine.

> Iterative techniques for processing large sparse linear systems were popular in the late 1950's and early 1960's (*and their decaying remains still pollute some computational circles*). When iterative methods finally departed from the finite element scene in the mid 1960's -- having been replaced by direct sparse-matrix methods -- the result was a quantum leap in the *reliability* of linear analysis packages, which contributed significantly to the rapid acceptance of FE analysis at the engineering group level. (This effect, it should be noted, had nothing to do with the relative Computational Efficiency, in fact iterative methods can run faster on many problems if the user happens to know the optimal acceleration parameters.) Presently, linear FE analyzers are routinely exercised as black box devices; ...

Our own view of the situation is different. By their training the experts in iterative methods expect to collaborate with users. Indeed the combination of user, numerical analyst and iterative method can be incredibly effective. Of course, by the same token, inept use can make any iterative method not only slow but prone to failure. Gaussian elimination, in contrast, is a classical black box algorithm demanding no cooperation from the user. As the tower of scientific computation grew so did the value of a reliable black box program. In the 1950's it did not seem possible that Gaussian elimination could ever be adapted efficiently for large systems of orders exceeding 200. So the alacrity with which serious users switched to direct methods in the 1960's must have been a painful surprise to the adepts of relaxation methods. As computer systems developed one saw time and again that as soon as direct methods became feasible for a particular problem they were preferred to their more computer efficient iterative rivals.

Surely the moral of the story is not that iterative methods are dead but that too little attention has been paid to the user's current needs? The asymptotic convergence theory is only part of the picture. Some exciting new directions are making their appearance.

The recent work of Brandt [Brandt, 1977], and of Frederickson, can be seen as a way of getting good starting approximations for relaxation methods. We are used to the idea of refining a discretization mesh in order to improve accuracy. Brandt's idea is to go back temporarily to a cruder mesh when this is called for. For example, if the error at the end of a relaxation sweep is judged to be decaying slowly it may be better to go back to a coarser mesh for the purpose of purging the recalcitrant low frequency components of the error rather than wait for relaxation to do it. Please note that the programs needed to effect these powerful ideas are decidedly more complicated than standard favorites like Successive Overrelaxation.

C. Small Matrix Computations

Good methods have been invented for nearly all the variants of the three basic tasks: (1) solve $Ax = b$ for x, (2) find (minimal length) x to minimize $\|b-Ax\|$, and (3) solve $Ax = \lambda x$ for λ and x. These methods assume that all the elements of A can be held simultaneously in the fast store of the computer. Such matrices are called small (or stored) although a 100 × 100 matrix would have seemed huge in 1950. In fact the methods are so well understood that they have been turned into black box programs demanding no tricky choices of tolerances or accuracy. This is a valuable byproduct of successful theoretical analysis of these methods.

Excellent. When the LINPACK collection of programs joins the EISPACK collection we should turn our energies to other areas.

Such a reaction is an example of territorial thinking. Often, but not always, these matrix computations are hidden parts of larger programs. It is very desirable that these hidden computations be performed as reliably as is the square root function. Is this possible? Is it a reasonable aim? Will our programs fail to return a solution only when there is no reasonable solution? For example, in linear least squares problems a decision must be made explicitly or implicitly, as to the rank of the coefficient matrix A. Often the algorithm can make the appropriate decision with no fuss but there are cases where the use to which the solution will be put should influence that decision. *The less one knows about a computing task the easier it is to demand reliability*. Nevertheless reliable black boxes are needed here if really complex programs are to be built above this level in the tower. As the human user recedes standards must go up. The small matrix programs referred to above are good but are they fail-safe? Do they respond charitably to ill-posed tasks? For the most part no.

Here is the last sentence from Forsythe (1966):

> The proper treatment of fine points is the reason why professionals should concentrate very hard on completely *foolproofing* the algorithms they devise, before putting them on the shelf for widespread use.

To make a program foolproof is almost the same as providing a proof of correctness. Sanderson [paper presented at the Atlanta SIAM meeting, 1976] has attempted such a proof for TQL1, the Eispack program for reducing a symmetric tridiagonal matrix to diagonal form. He found two Fortran statements which obstructed such a proof and permitted failure on a few artfully constructed examples. Yet the failure was not inevitable. The statements can be rewritten so as to preserve the mathematical relationships used in the exact arithmetic convergence proofs. Sanderson was able to prove correctness for the amended algorithm under reasonable hypotheses on the roundoff properties of the arithmetic unit. What we are seeing here is an example of a new thrust in theoretical numerical analysis. Such analysis is strongly influenced by work in computer science but it should not be classified as Mathematical Software which is concerned with the use and dissemination of programs, not with a particular detailed analysis. Others might wish to classify these activities as part of Mathematical Software on the grounds that numerical analysis should confine itself to mathematical methods and keep away from programs. The reader may take his choice.

There is more work for matrix specialists than the foolproofing of their algorithms. For those problems in which a black box approach is not feasible there will be a need for somewhat more elaborate programs which include sensitivity analysis and describe the whole solution set to a bunch of numerically indistinguishable problems. At present condition numbers for various problems (a simple measure of the sensitivity of the solution to infinitessimal changes in the data) are found only in theoretical work. Despite the extra work involved we can expect future versions of current methods to produce condition numbers along with the basic solutions.

Last but not least comes the appearance of new problems involving stored matrices. Here is a small selection.

(a) Update the factors of a matrix when the matrix is changed slightly (by addition of a rank one matrix or by deletion of a row, for example). The method should be more efficient than starting from scratch.

(b) Given matrices A, B, C find a matrix X, if it exists, such that

$$AX + XB = C \ .$$

This is a special set of linear equations and warrants special treatment. See [Bartels & Stewart, 1972] for a stable solution via the Schur form.

(c) Find exp(tA) or $A^{1/2}$. See [Moler & van Loan, 1977] for a discussion of the former.

(d) A given matrix should be positive definite but is not because of key punching errors. Detect the elements which are most likely to be in error.

We conclude that small matrix computations are moving down novel avenues not contemplated twenty years ago. This is largely because standard matrix calculations have sunk down out of the user's sight and are taken for granted.

D. Nonstiff Systems of ODE's

A recent survey [Shampine, Watts & Davenport, 1976] shows that programs based on the traditional methods (Runge-Kutta and multi-step) have gone about as fas as they can go. The article brings out the fascination of trying to satisfy several conflicting demands centering on quick execution and reliability. Has the point of diminishing returns been reached? Let us see.

The price paid for swiftness is that the user of the best of available programs is required to select a tolerance for the error per step (or the error per unit step). Most of these methods use variable step size and variable order schemes so the effects of various tolerance choices are hard to predict. The need for this choice keeps these ODE computations in the external mode, in direct contact with users.

Methods which furnish a global error estimate are beginning to appear [Stetter, 1972; Shampine & Watts, 1976]. They are significantly more demanding of machine time than are the traditional programs. We expect to see research on the extent to which the user can be removed from the solution of ODE's. If the price is not exorbitant we can expect these computations to join matrix computations at the very useful hidden level of the tower of scientific computation. Again we find the need for more work in this area.

4. THEORY -- NECESSITY OR LUXURY?

A. Examples

Perhaps the previous pages seem overly preoccupied with programs, ignoring the mathematical highlights of numerical analysis. Our reflections on this question suggest that in considering the subject as a whole we must give pride of place to computing tasks and to algorithms for accomplishing them. The mandate for numerical analysts is to invent methods and to explain their behavior. This statement is too sweeping, there must be a judgment that the tasks are worthwhile. To ignore this constraint is to court decadence. Theoretical work thus has a natural role in explaining and predicting the performance of worthwhile methods. In addition the best work in our field shows the powerful influence of good theory on the development of good methods. Here are some examples.

1. The Lax-Richtmyer theory [Richtmyer & Morton, 1967] provided a framework for understanding the behavior of difference schemes for linear initial value problems. The conflict between accuracy and stability explained the difficulties in finding useable high order schemes in problems in two or more space dimensions.

2. The Dahlquist theory [Henrici, 1962] explained why some accurate multi-step methods for ODE's which were fine for use with desk calculators failed as automatic (black box) programs. It gave a precise limit on the accuracy of stable multi-step methods.

3. [Frankel, 1950] and Young's 1950 thesis [Young, 1954] turned into a science Southwell's art of overrelaxation for solving linear boundary value problems. The comparison of the asymptotic behavior of rival linear stationary methods was soon well understood. Quickly the older techniques were refined and adapted to an impressive variety of PDE's.

4. Wilkinson explained away the worries which had beset those who, in the early 1950's, wanted to use Gaussian elimination as an automatic procedure for solving large sets of linear equations with twenty or thirty unknowns. The subject of roundoff error analysis seemed to be important, boring, and difficult. Wilkinson developed an approach which made relatively simple the analysis of matrix computations executed in noisy arithmetic. Once this point of view was grasped new, reliable methods came quickly, an unavoidable consequence of right thinking. Who, in 1950, would have thought of effecting the Gram-Schmidt process via a product of orthogonal matrices?

5. Schoenberg's early work [Schoenberg, 1946] on splines was pure approximation theory but it showed how much could be done with functions which are not very smooth globally. The practical impact was slow in coming but it has spread astonishingly far. Almost no text books prior to 1970 mention approximation by piecewise polynomials.

B. Paradigms

Each of these theories didmore than present nice results; they established what Thomas Kuhn [Kuhn, 1962] would call a paradigm, an accepted way of looking at the topics. Yet the very success of a paradigm can lead to stagnation. A good example of this phenomenon is the comparatively long time that elapsed before it was realized that Dahlquist's model was not the most general way of saving information from the past for use in Predictor-Corrector methods. Thus the Dahlquist barrier is a limitation on one particular class of PC methods. See [Gear, 1971] for more details.

In fact it is all to easy to forget that the stability and convergence notions which dominate our thinking are asymptotic qualities developed for linear problems. They are not sacred, they are guides. The Particle-in-Cell techniques for fluid flow calculations needed the nonlinearity of the differential equations in order to work. Another case is the

Glimm-Chorin method described in Section 1. It is not even consistent in the usual polynomial sense and would be hopeless on linear problems.

However the fact that success can lead to blinkers is no reason for avoiding good paradigms.

Perhaps the most important, albeit indirect function of good theoretical numerical analysis is the intellectual formation of a group of experts. In the process of mastering the theory the paradigm is absorbed with sufficient thoroughness to be a basis for thought. If this be the case it goes some way in explaining why ideas and exposition are more important than "mere" results.

C. FEM

Let us now turn to the Finite Element Method which appears to be a counterexample to the notion that a good paradigm is necessary for the development of good methods.

To those readers not familiar with the FEM this paragraph is addressed. The region on which a boundary value problem is to be solved is divided up into simple pieces called elements (rectangles and triangles in 2 dimensions) and the solution is approximated by polynomials of a certain degree on each element. The polynomials are matched on the boundaries of each element to have as low order of smoothness as one can get away with. The piecewise polynomial function with minimum "energy" is chosen as the best approximation. The definition of energy is particular to each problem. In order to find this function it is necessary to solve a set of algebraic equations. Frequently this set of equations can be interpreted as a set of difference equations but for realistic problems such an interpretation is strained. Ordinary finite difference approximations would never come up with such equations and so FEM really is a rival approach. Its flexibility is well suited to complicated regions, such as bridges or buildings, and the local character of the method makes for efficiency.

To the mathematician it is clear that the FEM is an instance of the Rayleigh-Ritz-Galerkin methods which were well understood before the era of modern numerical analysis. What is striking is that the FEM was developed and in widespread use before its intimate connection with Rayleigh-Ritz was appreciated. All this looks like evidence against the need for a good paradigm before useful methods can be developed. However the FEM was not found by numerical analysts. It is quite plausible that the civil engineers' understanding of structural analysis was more than adequate for pointing the way to more useful approximation techniques than they would have learned from numerical analysis courses.

The importance of FEM in numerical analysis is that it, coupled with direct methods for systems of linear equations, has moved linear structural analysis in two, and even three dimensions down from the external to the inner directed category. You can model some very complicated structures with a program "off the shelf". There is no need for an expert in numerical methods!

Ever since they found out about the method the numerical analysts have done an excellent job in explaining it and seeing to what other problems it can be applied. Error estimates in terms of the strain energy come fairly naturally from approximation theory and the Rayleigh-Ritz-Galerkin framework. What was not so obvious was how to bound the error in the least squares or uniform norms and clever work has been done on this topic.

Other problems are the rigorous analysis of the effect of using curved boundaries for some of the elements, the degree to which the thinness of some finite elements degrades the solution, and the discovery of conditions under which the proper degree of smoothness in the trial solutions can be safely avoided. The honors list of those who have made impressive contributions to FEM theory is growing so fast that we dare not try to give one here. Their reward is in the groves of Academe.

D. Ripe Areas

There are several areas in which there has been significant activity in algorithm creation and the time is ripe for a more complete theory. We will give a few examples.

Adaptive Algorithms: In most quadrature schemes a definite integral is approximated by a weighted sum of function values. Traditionally the points at which evaluation occurred were fixed in advance as part of the scheme. A radical idea, stemming from computer science rather than mathematics, was to let the program choose the next evaluation points in the light of the current state of the approximation. Such a scheme offers the hope that the points will be located where the function seems to be roughest and so the number of evaluations needed for a given accuracy could be independent of the global smoothness of the integrand. Various adaptive quadrature methods, as these are called, have been developed and their success has been mixed. They certainly have their place but they are not a general panacea. There are plenty of things to be explained. Rice [Rice, 1975] has presented an asymptotic theory showing that indeed, in the limit, functions with integrable singularities can be integrated as efficiently as smooth ones. In practice, however, Lyness and Kaganove [Lyness & Kaganove, 1976] have shown that adaptive schemes can be very inefficient, indeed disastrous, in certain situations. Moreover these situations are not easily recognized in advance. It turns out that the problem of termination is far more tricky with these more sophisticated methods than with formula evaluation. This is a typical problem of inner directed numerical analysis; subtle and unexpected side effects come from greater flexibility in the algorithm. We wish to emphasize that the clarification of the behavior of adaptive methods is a task for theoretical numerical analysis not mathematical software.

Stiff Systems of ODE's: The adjective stiff is applied somewhat loosely for any problem on which the traditional methods fail. Strictly speaking a differential equation itself should not be called stiff, rather it is the trio involving the equation, the interval of integration and the accuracy requirement which can be labelled stiff. Moreover problems can be stiff in radically different ways and there may, or may not be methods which are useful on all stiff problems. In some cases the type of stiffness may be fixed and known in advance. In other cases not.

This area is very hot at present, see [Miranker, 1975; Gear, 1971]. But it seems fair to say that no generally accepted paradigm has yet emerged.

Convergence of the QR Algorithm: For over ten years the QR algorithm has been the champion method for finding eigenvalues of small matrices, both symmetric and unsymmetric. It was quite surprising to the experts that one single method triumphed in a wide variety of situations. A complete and very satisfactory convergence theory has been made for the symmetric case but, contrary to popular belief, there is no explanation for the unfailing convergence of the shift strategy used for general (nonnormal) matrices. It would be nice to understand the methods we put "onto the shelf". Faith is admirable but it should be conserved for more profound issues.

5. CONCLUSION

As a result of thirty years work on numerical methods and on computers many of the basic tasks discussed in introductory textbooks on numerical analysis can now be carried out by canned programs. As a result more ambitious calculations are being undertaken now than ever before. Consequently there is room for ever more complicated fiascos in which the numerical output is completely misleading. As the tower of scientific computation grows there will be a need for people who understand numerical methods at the various levels in the tower. There will be many phenomena to be explained but it is unlikely that there will be, or need be, general convergence theorems or error bounds for nonlinear problems per se. Instead we expect a proliferation of special results designed to exploit all the features of particular applications. There will be a boom in the study of numerical methods but it is quite likely that courses entitled Numerical Analysis will disappear as the subject evolves into distinct applications just as Engineering has done.

REFERENCES

R. Bartels and G.W. Stewart, "Algorithm 432, solution of the matrix equation $AX + XB = C$," Communications of the Association for Computing Machinery 15 (1972) 820-826.

G. Birkhoff and R.S. Varga, "Implicit alternating direction methods," Transactions of the American Mathematical Society 92 (1959) 13-24.

J. Bramble and S. Hilbert, "Bounds for a class of linear functionals with application to Hermite interpolation," Numerische Mathematische 16 (1971) 362-369.

A. Brandt, "Multi-level adaptive solutions to boundary value problems," Mathematics of Computation 31 (1977) 1-

A. Chorin, "Random choice solution of hyperbolic systems," Journal of Computational Physics (1976)

J. Cooley and J.W. Tukey, "An algorithm for the machine computation of complex Fourier coefficients," Mathematics of Computation 19 (1965) 297-301.

B.T. Smith et alii, "Matrix eigensystem routines," (EISPACK) Lectures Notes in Computer Science 6, Springer-Verlag (1976).

G.E. Forsythe, "Today's computational methods of linear algebra," SIAM Review 9 (1967) 489-515.

S.P. Frankel, "Convergence rate of iterative treatment of partial differential equations," Mathematical Tables and Aids to Computation 4 (1950) 65-75.

C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice Hall, Englewood Cliffs, N.J. (1971).

M. Gentleman and Sande, "Fast Fourier transforms for fun and profit," AFIPS Fall Joint Computer Conference (1966) 563-578.

J. Glimm, "Solutions in the large for nonlinear hyperbolic systems of conservation laws," Communication of Pure and Applied Mathematics 18 (1965) 697.

P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, Wiley and Sons, New York (1962).

M.A. Jenkins and J.F. Traub, "A three-stage algorithm for real polynomials using quadratic iteration," SIAM Journal of Numerical Analysis 7 (1970) 545-566.

W. Kahan and B.N. Parlett, "Can you count on your calculator?" ERL Memorandum M77/21, University of California, Berkeley (1977).

J. Kuhn, The Structure of Scientific Revolutions, Dover Press, New York (1962).

H. Kuki, "Mathematical function subprograms for basic 1970 system libraries--objectives, constraints and trade-off," in Mathematical Software (J. Rice, ed.), Academic Press, New York (1971).

J.N. Lyness and J.J. Kaganove, "Comments on the nature of automatic quadrature routines," ACM Transactions on Mathematical Software 2 (1976) 65-81.

W.L. Miranker, "The computational theory of stiff differential equations," Pubblicazioni Serie III No. 102, Istituto per le Applicazioni del Calcolo "Mauro Picone", Roma, 1975.

C. Moler and G.W. Stewart, "An algorithm for generalized matrix eigenvalue problems," SIAM Journal of Numerical Analysis 10 (1973) 241-256.

C. Moler and C. van Loan, "Nineteen ways to compute the exponential of a matrix," SIAM Journal of Applied Mathematics (1977).

J.R. Rice, "A metalgorithm for adaptive quadrature," Journal of the Association for Computing Machinery 22 (1975) 61-82.

J.R. Rice, "Algorithmic progress in solving PDE's," Computer Science Department, TR 173, Purdue University, 1976.

R.D. Richtmyer and K.W. Morton, Finite Difference Methods for Initial Value Problems, 2nd ed., Interscience, Division of John Wiley, New York (1967).

I.J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions, Parts A and B," Quarterly Journal of Applied Mathematics 4 (1946) 45-99, 112-141.

L.F. Shampine and H.A. Watts, "Global error estimation for ordinary differential equations," ACM Transactions on Mathematical Software 2 (1976) 172-186.

L.F. Shampine, H.A. Watts and S.M. Davenport, "Solving nonstiff ordinary differential equations--the state of the art," SIAM Review 18 (1976) 376-411.

B. Smith, "ZERPOL, a zero finding algorithm for polynomials using Laguerre's method," Department of Computer Science, University of Toronto, 1967.

H.J. Stetter, "Local estimation of the global discretization error," SIAM Journal of Numerical Analysis 8 (1971) 512-523.

J.H. Wilkinson, "Error analysis of direct methods of matrix inversion," Journal of the Association for Computing Machinery 8 (1961) 281-330.

J.H. Wilkinson and C. Reinsch, Handbook for Automatic Computation. Vol. II. Linear Algebra, Springer-Verlag, Berlin (1971).

D. Young, "Iterative methods for solving partial difference equations of elliptic type," Transactions of the American Mathematical Society 76 (1954) 92-111.

Table 1

Poisson Problem on Unit Cube in 3 Dimensions

Date	Method	Multiplies	Storage
1945	(A) 7-pt. star Gauss-Seidel	437×10^6	190,000
1965	(B) Fast Fourier Transform	628×10^3	32,000
1965	(C) 27-pt. star Tensor product	8.2×10^3	1,500

Time Ratios: A/B = 670, A/C = 53,000

Note: Method C [Direct solution of PDE's by tensor product methods, by Lynch, Rice, and Thomas, Numer. Math. 6 (1964), 185-199] has been analyzed but not implemented.

Table 2

General Elliptic Operator on a Simple, Plane Domain D

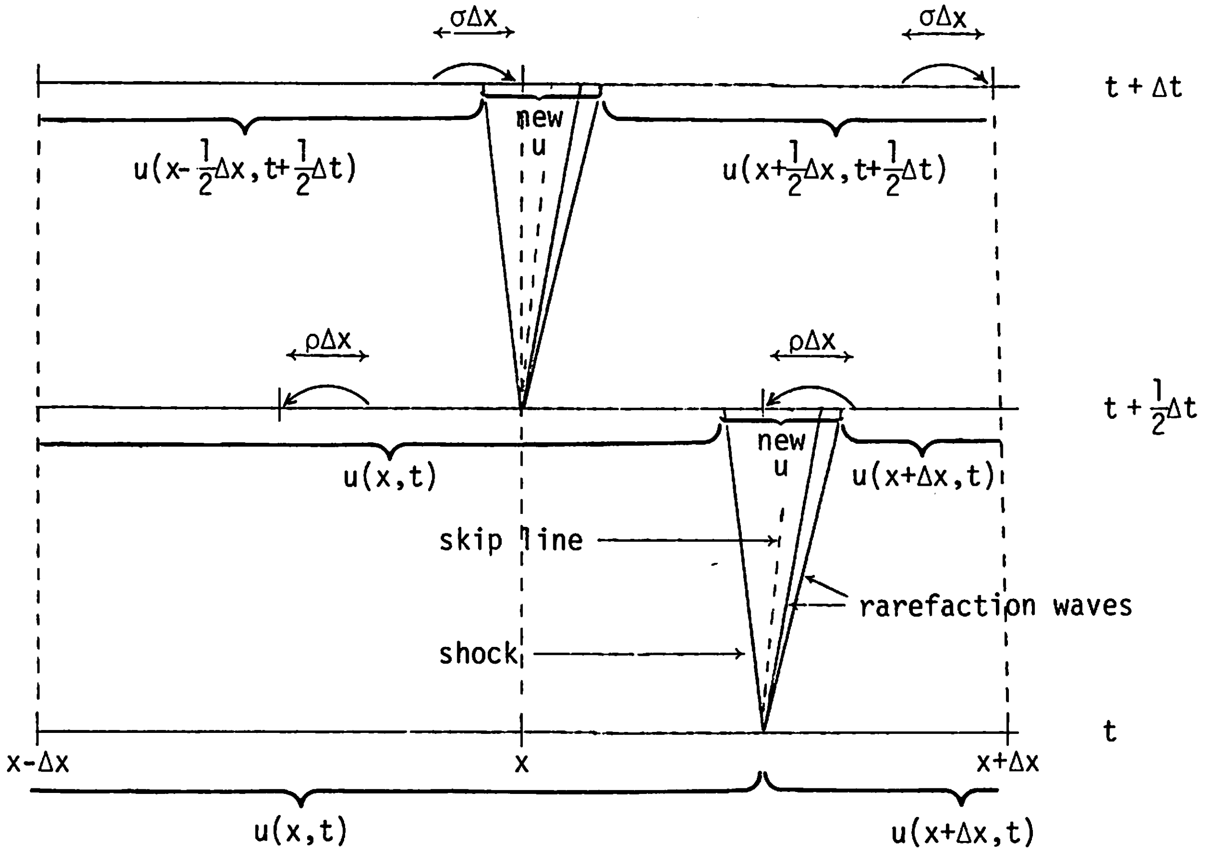
Date	Method	Multiplications
1945	(A) Finite Differences Gaussian Elimination	9×10^{11}
1965	(B) Better boundary approximations Iteration	20×10^6
1975	(C) Collocation with Hermite Cubics Tensor product methods	2.5×10^6

Time Ratios: A/B = 4.5×10^4, A/C = 3.5×10^5

Additional speed ups by factors of 10 or more are possible with the latest techniques.

Figure 1

Diagram of Glimm/Chorin Method

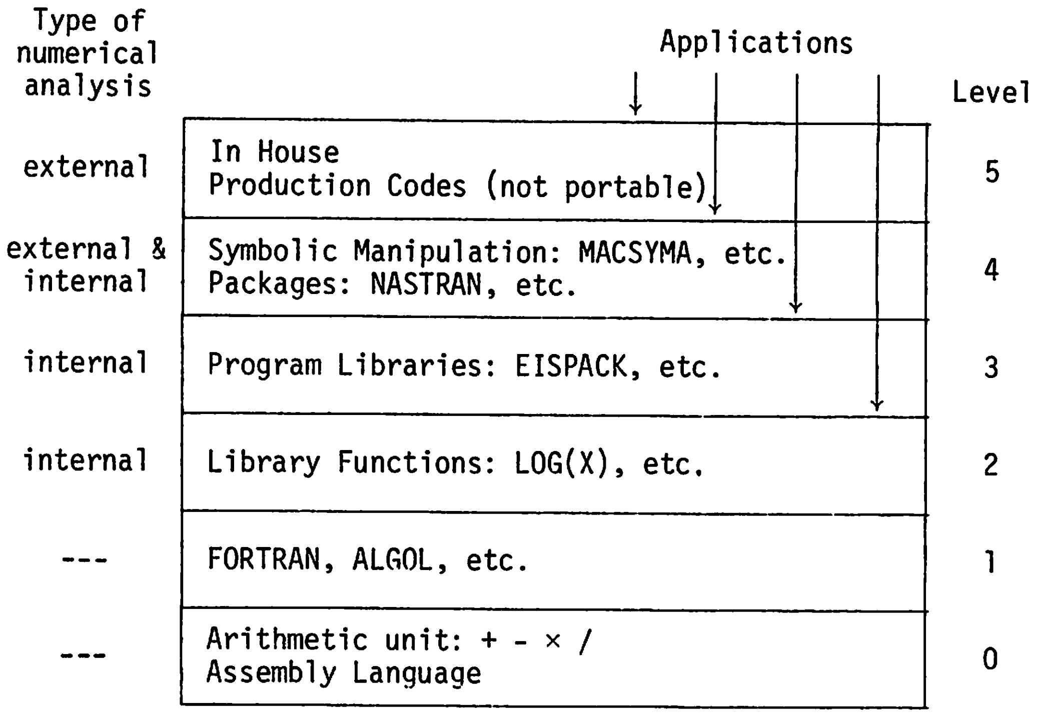

σΔx
σΔx
t + Δt
new
u
$u(x-\frac{1}{2}\Delta x, t+\frac{1}{2}\Delta t)$
$u(x+\frac{1}{2}\Delta x, t+\frac{1}{2}\Delta t)$
ρΔx
ρΔx
$t + \frac{1}{2}\Delta t$
new
u
u(x,t)
u(x+Δx,t)
skip line
rarefaction waves
shock
t
x-Δx
x
x+Δx
u(x,t)
u(x+Δx,t)

Derivation of u(x,t+Δt)

ρ and σ are random equidistributed numbers in $[-\frac{1}{2},\frac{1}{2}]$

Figure 2

The Tower of Scientific Computation



Type of numerical analysis		Level
external	In House Production Codes (not portable)	5
external & internal	Symbolic Manipulation: MACSYMA, etc. Packages: NASTRAN, etc.	4
internal	Program Libraries: EISPACK, etc.	3
internal	Library Functions: LOG(X), etc.	2
---	FORTRAN, ALGOL, etc.	1
---	Arithmetic unit: + - × / Assembly Language	0