Journal of Quality Measurement and Analysis *JQMA 4(1) 2008, 1-9*
Jurnal Pengukuran Kualiti dan Analisis

NUMERICAL ANALYSIS

JOHN BUTCHER

ABSTRACT

Mathematics has applications in virtually every scientific discipline. Many of these applications yield mathematical models which involve numerical approximations or evaluations to give a complete answer. Over the years, especially within the computer age, a body of knowledge has grown up which seeks to understand how calculations should best be performed to carry out these evaluations. Much of this knowledge is in the form of algorithms for solving certain standard and widely used problems. This is the subject of Numerical Analysis. In this broad survey of numerical analysis, I will attempt to survey some of the problem areas it deals with. I will then conclude by focussing specifically on some aspects of the area of my own specialist interests, that is differential equations and their numerical approximations. In the numerical solution of differential equations, where the numerical approximation is developed in small time-steps, there are typically three challenges. These are (a) to keep errors in each step small, (b) to make sure that the overall algorithm is stable and that errors generated in any step do not have an overwhelming affect on the accuracy of later steps and (c) to keep the computational costs as low as possible. Selecting a numerical method, or family of methods, and integrating the method or methods into a software package, deals with all these challenges and, of necessity, seeks compromises between them. It is found that the analysis of possible methods, and algorithm design questions, makes extensive use of results and techniques from many areas within the mathematical sciences, and even contributes to them.

Keywords: Numerical analysis; differential equations; numerical methods

1. Introduction

The applications of mathematics are everywhere, not just in the traditional sciences of physics and chemistry, but in biology, medicine, agriculture and many more areas. Traditionally, mathematicians tried to give an exact solution to scientific problems or, where this was impossible, to give exact solutions to modified or simplified problems. With the birth of the computer age, the emphasis started to shift towards trying to build exact models but resorting to numerical approximations.

Today many scientific problems can be modelled quite accurately using a combination of mathematical analysis and numerical computation. From the large body of knowledge and experience acquired from masses of problem solutions, a number of computational areas have become identified as forming a systematic structure. The purpose of this paper is to survey some parts of this systematic structure and to explore some details within it. This body of work forms the subject of numerical analysis.

In Section 2 we will survey the broad subject of numerical analysis and in Section 3, we will go into some aspects of numerical methods for differential equation. Finally, in Section 4, we will explore some links between other parts of mathematics and numerical analysis.

2. Numerical Analysis as a Mathematical Science

There is a natural progression from being a collection of unrelated techniques to a systematic scientific discipline. Numerical analysis began with the need to solve practical problems in

whatever way was needed. Many problems involve the solution of linear equation systems and it is natural that algorithms to solve problems in linear algebra should be at the heart of the evolving science of numerical analysis. The basic idea of eliminating variables one by one, can be made systematic and this leads to modern LU factorization algorithms using partial pivoting. For stability, QR factorization is preferred over triangular factorization and this has additional applications in eigenvalue computations. A natural mathematical question is whether algorithms which involve Cn^3 multiplications are optimal? Surprisingly they are not because the "Strassen algorithm" involves Cn^a multiplications, where $a = log_2 7 \approx 2.85$.

Numerical algorithms which evoke mathematical questions, and at the same time depend on sometimes sophisticated mathematics, are a common occurrence. The basic task of obtaining approximations to integrals has led to many very effective algorithms and at the same time has involved very deep mathematics. One classical question concerns "interpolational quadrature", in which integrals are approximated as the integrals of interpolation polynomials for a given function. For example, to calculate the integral $\int_0^1 \emptyset(x)dx$, a first step might be to find a first degree polynomial, which agrees with $\emptyset$ at $x = 0$ and $x = 1$. This polynomial is found to be $p(x) = (1 - x)\emptyset(0) + x\emptyset(1)$ and its integral is $\frac{1}{2}\emptyset(0) + \frac{1}{2}\emptyset(1)$. A good question to ask would be: if the result is to be based on exactly two points, say ξ_1 and ξ_2, what is the optimal choice for these points to yield the most accurate approximation possible for the integral? The answer is $\xi_1 = \frac{1}{2} - \frac{1}{6}\sqrt{3}, \xi_2 = \frac{1}{2} + \frac{1}{6}\sqrt{3}$. Why these numbers? And what is the best choice if n points $\xi_1, \xi_2, \ldots, \xi_n$ are used? The answers are related to the classical theory of orthogonal polynomials. What happens if the points are spaced uniformly on [0, 1]? Surprisingly the sequence of approximations as *n* increases might not even converge.

Linear equations and numerical integration are both used within methods for differential equations, so that there is a linkage between all these subjects which involves questions of practical and efficient computation as well as a wealth of mathematical disciplines.

3. Numerical Methods for Differential Equations

Given a differential equation, together with an initial value,

$$y'(x) = f\big(x, y(x)\big), \qquad y((x_0) = x_0,$$

the task that numerical integrators are asked to solve is to find approximations to $y(x_1), y(x_2), \ldots,$ for $x_1, x_2, \ldots$ an increasing sequence of points. We can regard this is a numerical integration problem

$$y(x) = y_0 + \int_{x_0}^{x} f\big(y(\xi)\big)d\xi,$$

and we can go from x_{n-1} to x_n by the formula

$$y(x_n) = y(x_{n-1}) + \int_{x_{n-1}}^{x_n} f\big(y(\xi)\big)d\xi,$$

One way of obtaining approximations to this integral, is to write

$$\int_{x_{n-1}}^{x_n} f\big(y(\xi)\big)d\xi \;\approx\; (x_n - x_{n-1})f(x_{n-1}),$$

and this gives the "Euler method"

$$y_n = y_{n-1} + (x_n - x_{n-1})f(y_{n-1}), \qquad n = 1,2,\ldots, \tag{1}$$

where y_n is the computed approximation to $y(x_n)$.

The method (1) is completely reliable but it is usually not very efficient. The reasons are that it is only first order and has bounded stability region.

"First order" means that asymptotically, for small stepsizes, the error in a computed answer is proportional to the stepsize. It would be better to have order 2 or three or more, because this would mean that errors would decrease more rapidly as additional computing resources are brought into play.

"Bounded stability region" means that, for so-called stiff problems, there would be an unacceptable bound on stepsizes to achieve stable calculations.

Important aims in the study of numerical methods for ordinary differential equations are first to obtain higher orders with a moderate increase in computational effort and secondly to achieve accurate results without being hampered by unreasonable stability restrictions.

I will briefly survey the most important approaches to improving on the Euler method in terms of higher order and greater efficiency. I will then say a little about obtaining stable computations.

3.1. *Linear multistep methods*

The simple idea of predicting the result at the end of a step, using a single approximation to the solution value, and a single value of the derivative, gives only first order accuracy. To improve on this, consider the possibility of basing a prediction on a number of previously computed solution values and a number of previously computed derivative values. For example, once we have got past the first step, we could use the sequence of numerical values satisfying the difference equation

$$y_n = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \beta_1 h f(y_{n-1}) + \beta_2 h f(y_{n-2}),$$

based on the approximation

$$y(x_n) \approx \alpha_1 y(x_{n-1}) + \alpha_2 y(x_{n-2}) + \beta_1 h y'(x_{n-1}) + \beta_2 h y'(x_{n-2}).$$

For this approximation to have order p, the Taylor expansion of

$$y(x_n) - \alpha_1 y(x_{n-1}) - \alpha_2 y(x_{n-2}) - \beta_1 h y'(x_{n-1}) - \beta_2 h y'(x_{n-2})$$

about x_n would need to be zero up to the h^p term. This Taylor expansion is

$$C_0 y(x_n) + C_1 h y'(x_n) + C_2 h^2 y''(x_n) + C_3 h^3 y'''(x_n) + \cdots,$$

where

$$C_0 = 1 - \alpha_1 - \alpha_2,$$
$$C_1 = \alpha_1 + 2\alpha_2 - \beta_1 - \beta_2,$$
$$C_2 = -\alpha_1 - 4\alpha_2 + 2\beta_1 + 4\beta_2,$$
$$C_3 = \alpha_1 - 8\alpha_2 - 3\beta_1 + 28\beta_2.$$

It seems to be possible for $C_0 = C_1 = C_2 = C_3 = 0$, and therefore to achieve order 3. This would mean $\alpha_1 = -4, \alpha_2 = 5, \beta_1 = 4, \beta_2 = 2$.

This introduces the new difficulty of stability. Even for the differential equation $y'(x) = 0$, whose solution is constant, the sequence of computed approximations would satisfy

$$y_n + 4y_{n-1} - 5y_{n-2} = 0,$$

and this difference equation has a solution which grows like powers of −4.

A method like this is said to be "unstable". Only stable methods can be used in practical computation.

At the expense of making methods slightly more expensive to use, (1) can be generalized to make it implicit:

$$y_n = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \beta_0 h f'(y_n) + \beta_1 h f(y_{n-1}) + \beta_2 h f(y_{n-2})$$

and this makes order as high as 4 possible without losing stability.

In the pioneering work of Dahlquist (1956), it was shown that, if k is odd, the maximum order possible for stable linear multistep methods is $k + 1$ and, if k is even, the maximum order is $k + 2$.

In practical computation, the choice is usually restricted to the (explicit) Adams-Bashforth (1883) methods, and the (implicit) Adams-Moulton (1926) methods

$$y_n = y_{n-1} + \sum_{i=1}^{k} \hat{\beta}_k y_{n-k,}$$

$$y_n = y_{n-1} + \sum_{i=1}^{k} \beta_k y_{n-k,}$$

used together as a "predictor-corrector" pair.

3.2. *Runge–Kutta methods*

Instead of basing an improvement to the Euler method on the use of more past history, it is possible to carry out more evaluations of the function f in each step. For example, instead of using a "lefthand rule" integration formula, we could use either the mid-point or trapezoidal rules. but with preliminary prediction used to obtain the off-step values:

$$y_n = y_{n-1} + hf\left(y_{n-1} + \frac{1}{2}hf(y_{n-1})\right),$$
$$y_n = y_{n-1} + \frac{1}{2}hf(y_{n-1}) + \frac{1}{2}hf\left(y_{n-1} + hf(y_{n-1})\right).$$

These methods from Runge (1895) paper are "second order", because the error in a single step behaves like $O(h^3)$. At a specific output point the error is $O(h^2)$. A few years later,

Heun (1900) gave a full explanation of order 3 methods and Kutta (1901) gave a detailed analysis of order 4 methods.

The most famous of all these "Runge–Kutta methods" is the fourth order method found by Kutta

$$
\begin{aligned}
Y_1 &= y_{n-1},\\
Y_2 &= y_{n-1} + \tfrac{1}{2}hf(Y_1),\\
Y_3 &= y_{n-1} + \tfrac{1}{2}hf(Y_2),\\
Y_4 &= y_{n-1} + \tfrac{1}{2}hf(Y_3),\\
y_n &= y_{n-1} + \tfrac{1}{6}hf(Y_1) + \tfrac{1}{3}hf(Y_2) + \tfrac{1}{3}hf(Y_3) + \tfrac{1}{6}hf(Y_4).
\end{aligned}
$$

Although methods of order p exist with *p* "stages", that is p evaluations of *f* per time-step, up to $p = 4$, above this it is impossible. By the time *p* has increased to 8, as many as 11 stages are necessary.

3.3. *General linear methods*

A large family of methods exists which combine the ideas of both linear multistep and Runge–Kutta methods. That is, they use multiple past information as input to each step and in each step a multiple number of stages is calculated. We give a single example. At the start of step number *n*, three quantities are available as input. These are $y_1^{[n-1]}$, which is an approximation to $y(x_{n-1})$, $y_2^{[n-1]}$, which is an approximation to $hy'(x_{n-1})$, and $y_3^{[n-1]}$, which is an approximation to $h^2y''(x_{n-1})$. In the step itself, three stages are calculated. These are $Y_1 \approx y\left(x_{n-1} + \frac{1}{3}h\right)$, $Y_2 \approx y\left(x_{n-1} + \frac{2}{3}h\right)$ and $Y_3 \approx y(x_{n-1} + h)$. The formula for these, together with the output values $y_1^{[n]} \approx y(x_n), y_2^{[n]} \approx hy'(x_n)$ and $y_3^{[n]} \approx h^2y''(x_n)$, are

$$
\begin{aligned}
Y_1 &= y_1^{[n-1]} + \tfrac{1}{3}y_2^{[n-1]} + \tfrac{1}{18}y_3^{[n-1]},\\
Y_2 &= \tfrac{1}{2}hf(Y_1) + y_1^{[n-1]} + \tfrac{1}{6}y_2^{[n-1]} + \tfrac{1}{18}y_3^{[n-1]},\\
Y_3 &= \tfrac{3}{4}hf(Y_2) + y_1^{[n-1]} + \tfrac{1}{4}y_2^{[n-1]},\\
y_1^{[n]} &= \tfrac{3}{4}hf(Y_2) + y_1^{[n-1]} + \tfrac{1}{4}y_2^{[n-1]},\\
y_2^{[n]} &= y_3^{[n-1]},\\
y_3^{[n]} &= 3hf(Y_1) - 3hf(Y_2) + 2hf(Y_2) - 2y_2^{[n-1]}.
\end{aligned}
$$

This is very close to being a Runge–Kutta method and is referred to as an "Almost Runge–Kutta" method. Other methods exist which are small modifications of linear multistep methods. For example there exist predictor-corrector methods with additional off-step predictors. However, general linear methods are now recognised in their own right. A comprehensive theory exists concerning their accuracy and stability. Furthermore there is a growing body of knowledge about how they can be efficiently implemented.

3.4. *Methods for stiff problems*

Instead of attempting to find a precise definition of stiffness, we can look at illustrative examples. Many stiff problem involve partial differential equations. When space derivatives are approximated by finite differences, to reduce these problems to large systems of ordinary

differential equations, we are in effect replacing unbounded operators on function spaces, by badly conditioned matrix operators on finite dimensional vector spaces. For example, the diffusion operator has a continuous spectrum, whereas finite difference approximations have a spectrum of widely spaced points on the negative real axis. In attempting to solve the discretized heat equation, we want to approximate the most slowly decaying components accurately. The accuracy of the approximations to the rapidly decaying components is relatively unimportant, but it is these components which create stability difficulties. We can isolate the rapidly decaying components from the rest of the problem, by looking at the solution of the simple linear problem $y'(x) = qy(x)$.

If q is a negative real number or, more generally, a complex number with negative real part, then we obtain a decaying solution. However, numerical approximations do not necessarily decay. With the simple Euler method, the numerical approximation changes by a factor $(1 + hq)$ in every timestep. But if hq is outside the circle $\{z: |1 + z| \leq 1\}$, computed solutions actually grow in magnitude. We can get round this by using the "implicit Euler method", in which the approximation to $y(x_n)$ is found from the equation $y_n = y_{n-1} + hf(y_n)$. It is an added complication to actually solve this algebraic equation to find y_n but the stability issue is now out of the way. To obtain stable numerical sequences, it is now only necessary that hq lies in $\{z: |z - 1| \geq 1\}$, which is always true if the real part of q is negative.

One source of methods for stiff methods is implicit linear multistep methods. For example the so-called "BDF2" method is

$$y_n = \tfrac{2}{3}hf(y_n) + \tfrac{4}{3}y_{n-1} + \tfrac{1}{3}y_{n-2}.$$

This has the desirable property of being "A-stable". This means that, for all z in the left half complex plane, the difference equation

$$\left(1 - \tfrac{2}{3}z\right)y_n - \tfrac{4}{3}y_{n-1} - \tfrac{1}{3}y_{n-2} = 0,$$

has only bounded solutions.

Unfortunately, it is not possible to find A-stable linear multistep methods with order higher than 2. However, A-stable Runge–Kutta methods exist for all orders. The simplest example, other than the implicit Euler method, is the implicit mid-point rule, which can be written:

$$y_n = y_{n-1} + hf\left(\tfrac{1}{2}(y_n + y_{n-1})\right).$$

This is based on the quadrature formula

$$\int_0^1 \emptyset(x)dx \approx \emptyset\left(\frac{1}{2}\right),$$

and one might ask if a similar use can be made of higher order Gaussian formulae, such as

$$\int_0^1 \emptyset(x)dx \approx \frac{1}{2}\left(\emptyset\left(\frac{1}{2} - \frac{1}{6}\sqrt{3}\right) + \emptyset\left(\frac{1}{2} + \frac{1}{6}\sqrt{3}\right)\right).$$

The answer is yes. For example, there exists a two stage implicit method with order 4. In general, by making use of the Legendre polynomial of degree s, it is possible to find an A-stable s stage implicit Runge–Kutta method with order $2s$.

But these are not ideal for other reasons, not least being their high implementation costs. By accepting less accuracy per stage, excellent implicit Runge–Kutta methods can be found for solving stiff problems. Even better methods can be found, within the larger general linear class.

4. Numerical Analysis and Mathematics

It is clear that problems in mathematical modelling frequently need to call upon computational methods. However, the nature of the relationship between numerical analysis and mathematics in general is much broader and much richer. In fact the relationship is two-sided. Numerical analysis is constantly drawing on and using sophisticated results from algebra and analysis but, at the same time, it is building new mathematics for its own purposes. This new mathematics can sometimes spill over into other parts of mathematics and into other sciences.

I mentioned the question of how quickly the cost of solving $n \times n$ linear equation systems grows as n increases. The observation that the cost (measured in terms of multiplications) can grow at the rate of $n^{log_2 7}$, rather at the rate n^3, hinges on the fact that partitioned matrix products

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix},$$

can be written in terms of only 7 matrix products:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} X_1 + X_4 - X_5 + X_7 & X_3 + X_5 \\ X_2 + X_4 & X_1 - X_2 + X_3 + X_6 \end{bmatrix},$$

where $X_1 = (A_{11} + A_{22})(B_{11} + B_{22}),\ X_2 = (A_{21} + A_{22})B_{11},\ X_3 = A_{11}(B_{12} - B_{22}),$
$X_4 = A_{22}(B_{21} - B_{11}),\ X_5 = (A_{11} + A_{12})B_{22},\ X_6 = (A_{21} - A_{11})(B_{11} + B_{12}),$
$X_7 = (A_{12} - A_{22})(B_{21} + B_{22}).$

The ideal properties of QR factorization as a means of reducing a linear algebra problem to a triangular system, hinges on the fact that multiplication by an orthogonal matrix is an isometry; that is it does not change the length of a vector on which it operates. This means that the "conditioning" of a matrix is not changed, and certainly not made worse, by multiplying by an orthogonal matrix.

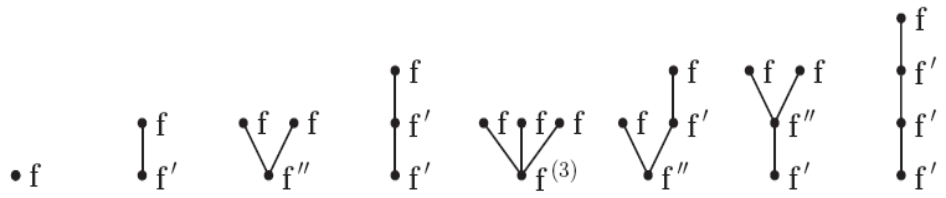
The fact that we can get accurate quadrature formulae at a reasonable cost depends on properties of classical orthogonal polynomials. These properties were discovered centuries ago and provided a ready-made resource waiting for numerical analysts to use. Orthogonal polynomials also play a number of roles in methods for solving differential equations, and some of these are quite surprising.

My final examples of the relationships between mathematics and numerical analysis relate to questions about the accuracy of numerical methods for ordinary differential equations. As a starting point, it is necessary to understand the Taylor expansion of the solution to the differential equation $y'(x) = f(y(x))$. It turns out that the formula for a higher-order

derivative $y^{(n)}(x)$ is related to the family of rooted trees with n vertices. Associated with the function f, which maps vectors to vectors, is its "Jacobian matrix" listing the various partial derivatives in a table. For convenience, write $\mathbf{f}$ for f evaluated at $y(x)$, and let $\mathbf{f}'$ denote the Jacobian matrix, also evaluated at $y(x)$. Similarly, write $\mathbf{f}''$, $\mathbf{f}'''$, ..., for higher derivatives of f. These are tensor quantities with an increasing number of indices. The following are the formulae of the first few derivatives of $y(x)$.

$$y'(x) = \mathbf{f},$$
$$y''(x) = \mathbf{f}'\mathbf{f},$$
$$y'''(x) = \mathbf{f}''(\mathbf{f},\mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f},$$
$$y^{(4)}(x) = \mathbf{f}^{(3)}(\mathbf{f},\mathbf{f},\mathbf{f}) + 3\mathbf{f}''(\mathbf{f},\mathbf{f}'\mathbf{f}) + \mathbf{f}'\mathbf{f}''(\mathbf{f},\mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f}'\mathbf{f}.$$

The connection with rooted trees becomes clear when the various terms are written as operator diagrams as follows:



The coefficients in the formulae for the derivatives are also related to classical combinatorial enumeration questions.

To find the order of a Runge–Kutta method, we need to find the Taylor expansion of the result computed after a single step, and match it against the Taylor expansion for the exact solution. Hence the number of order conditions increases as the number of rooted trees up to this order.

The coefficients which appear with each tree in the Taylor expansion of the exact solution characterize the method. It is possible to build an algebraic system for manipulating these collections of coefficients. The series itself is known as a "B-series" and the algebraic system related to it has recently been identified as being a "Hopf Algebra".

This is an example where numerical analysis not only produces new mathematics to understand and solve its own problems, but where it makes a wider contribution. The Hopf Algebra of rooted trees, which is at the heart of this part of numerical analysis, has recently been rediscovered for its applications in non-commutative geometry and in renormalization of the Feynman integrals of mathematical physics.

An account of the connection between these topics is given in a review paper by Brouder (2000).

References

Bashforth F. & Adams J. C. 1883. *An Attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid, with an Explanation of the Method of Integration Employed in Constructing the Tables which Give the Theoretical Forms of Such Drops*. Cambridge: Cambridge University Press.

Brouder C. 2000. Runge–Kutta methods and renormalization. *Eur. Phys. J. C.* **12:**521–534.

Dahlquist G. 1956. Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand*. **4**: 33–53.

Heun K. 1900. Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Z. Math. Phys*.**45** (1900), 23–38.

Kutta W. 1901. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Z. Math. Phys.* **46**: 435–453.

Moulton F. R. 1926. *New Methods in Exterior Ballistics*. Chicago: University of Chicago.

Runge C. 1895. Über die numerische Auflösung von Differentialgleichungen. *Math. Ann*. **46**: 167–178.

Department of Mathematics
The University of Auckland
Private Bag 92019
Auckland
NEW ZEALAND
Email: butcher@math.auckland.ac.nz