SYMPOSIUM ON NUMERICAL MATHEMATICS

DURBAN, 10–11 APRIL 1975

COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF NATAL DURBAN

SYMPOSIUM ON NUMERICAL MATHEMATICS

DURBAN, 10 - 11 APRIL 1975

COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF NATAL
DURBAN

Printed by the University of Natal.

PREFACE

During the period 15 March - 15 April 1975 Professor Lothar Collatz from the University of Hamburg visited the Computer Science Department of the University of Natal, Durban.

In order to stimulate interest in the research in and teaching of numerical mathematics in South Africa, it was decided to organize a symposium to coincide with his stay. This first South African Symposium on Numerical Mathematics was held in Durban on 10 and 11 April, 1975. Apart from being attended by most of the researchers active in this field, it was also attended by Professor Fritz John from the Courant Institute for Mathematical Sciences, New York.

The visit of Professor Collatz was sponsored by the Visiting Lecturers Trust Fund of the University of Natal, Durban and the Alexander von Humboldt Foundation, Bonn-Bad Godesberg, Germany.

In order to speed up the publication of the proceedings of the symposium all papers and abstracts are published in the form they were received from authors.

G.R. Joubert

CONTENTS

Page

Preface iii

Monotonicity and numerical mathematics by L. Collatz 1

Generalized conjugate directions in function optimization by D.J. van Wyk 3

Non-local convergence of Newton-Raphson iteration by R.M. Walker ... 13

Stiff ordinary differential equations and their numerical solution by H.W. Kropholler and E.T. Woodburn 29

Automatic step-size control in parabolic partial differential equations by D.P. Laurie 47

Numerical procedures for solving a fourth-order parabolic partial differential equation by G.R. Joubert 63

A Lanczos-type algorithm for the eigenvalues of an arbitrary matrix by C.J. Wright 71

Non-well posed problems in analysis by F. John 73

On the numerical solution of systems of non-linear differential difference equations by J.D. Neethling 75

A method for determining the eigenvalues and eigenvectors of a perturbed matrix by C.A. Botsaris 77

MONOTONICITY IN NUMERICAL ANALYSIS

L. Collatz
University of Hamburg
Germany

SUMMARY

1. There are two important tasks for Numerical Analysis.
   1. Development of new powerful and effective numerical methods for computers.
   2. Theoretical investigation of these methods, especially construction of exact error bounds for the approximate solutions one has got on the computer.

   This lecture deals only with No. 2, for which functional analysis is a very helpful aid. The most useful ideas are norms, distances and orderings; the orderings are perhaps the most fundamental.

2. On the orderings are based the following ideas:
   1. The comparability: Suppose f,g two elements of a partial ordered space R with $f<g$, then they define an interval $J = \langle f,g \rangle = \{h, f<h<g\}$
   2. Lattices: If for every pair $f,g \in R$ exists inf (f,g) and sup (f,g) then R is a lattice.
   3. Operators T of monotonic type: they have the property, that $Tf<Tg$ implies $f<g$. Wide classes of linear and nonlinear initial and boundary value problems for ordinary and partial differential equations are covered by the theory of operators of monotonic type.
   4. Monotone operators. T is called syntone (resp. antitone) if $f<g$ implies $Tf<Tg$ (resp. $Tf>Tg$). A sum of a syntone and an antitone operator is called a monotonically decomposible operator (M.D.O.). For M.D.O. exists a theory for existence and inclusion of solutions. Every Hammerstein nonlinear integral operator is under weak conditions a M.D.O.

3. Applications to many different cases are described:
   1. To inverse problems of boundary value problems in potential theory.
   2. To expansive integral operators.
   3. To monotonicity properties in the method of finite elements.
   4. To mixed boundary value problems (Dirichlet-Neumann) conditions (example: vertical laminer rivulet flow with gas-liquid interface) and many other problems.

GENERALIZED CONJUGATE DIRECTIONS IN FUNCTION OPTIMIZATION

D. J. van Wyk
Department of Pure and Applied Mathematics
Potchefstroom University for C.H.E.
Potchefstroom

Many algorithms for the unconstrained optimization of a function $f(x)$, $x \in R_n$, consist basically of the following iteration. Starting with an initial approximation x^o of the minimum point $\bar{x}$,

$$x^{i+1} = x^i + \alpha_i s^i, \; i = 0;1;2;\ldots, \qquad (1)$$

where the parameter α_i is chosen to minimize $f(x^i + \alpha s^i)$ as a function of the single variable α. The vector s^i can therefore be interpreted as a direction in which we move from x^i with x^{i+1} the optimal point in this direction. Such a method is called a conjugate direction method if the directions satisfy the following condition.

<u>Definition</u>: A set of non-zero vectors $u^o, u^1, \ldots, u^{n-1}$ are called *conjugate* to each other with respect to a given positive definite symmetric matrix A if

$$u^{i^T} A u^j = 0, \; i \neq j.$$

A general method to produce a set of A-conjugate vectors is the Gram-Schmidt orthogonolization procedure. Starting with a set of n linearly independent vectors $\{v^i\}$, the conjugate set $\{u^i\}$ can be developed by application of the recursion formula

$$u^{k+1} = v^{k+1} - \left(\frac{u^{1^T} A v^{k+1}}{u^{1^T} A u^1} \; u^1 + \ldots + \frac{u^{k^T} A v^{k+1}}{u^{k^T} A u^k} \; u^k\right)$$

The theory of conjugate directions in function minimization relates almost entirely to quadratic functions. The main reason for this is that the behaviour of a minimization algorithm on a quadratic function is indicative of its behaviour in the neighbourhood of the minimum of a general function $f(x)$, since near the minimum $f(x)$ can be approximated by a quadratic. The importance of conjugate directions in the minimization of quadratic functions is stressed by the following theorem.

<u>Theorem</u>: If the iteration (1) is applied to a quadratic function with positive definite Hessian G and the directions $s^o, s^1, \ldots, s^{n-1}$ are G-conjugate, then the minimum is found in at most n iterations, and moreover, every x^i is the minimum point in the subspace generated by the initial approximation x^o and the directions $s^o, s^1, \ldots, s^{n-1}$.

The minimization of a quadratic function is equivalent to the solution of a set of linear equations. The idea of using the property of conjugacy was originally applied to the latter problem. The prototype for this class of algorithms was described by Fox, Huskey and Wilkinson (1948); for the equations $Gx+b=o$, with G an n×n matrix and $b \in R_n$,

$$g^{k-1} = Gx^{k-1}+b$$

$$\mu_{k-1} = -\frac{u^{k-1^T} g^{k-1}}{u^{k-1^T} G u^{k-1}} \qquad k=1;2;\ldots;n \qquad (2)$$

$$x^k = x^{k-1}+\mu_{k-1}u^{k-1}$$

The directions $u^o, \ldots, u^{n-1}$ were constructed recursively as linear combinations of the unit vectors in such a way that they were conjugate. The well-known conjugate gradient method was discovered independently by Hestenes and Stiefel (1952) and is a special case of (2).

It can be shown that the coefficient α_i in (1) can be expressed by

$$\alpha_i = -\frac{g^{i^T} s^i}{s^{i^T} G s^i}$$

if the iteration is applied to a quadratic function of the form

$$f(x) = a+b^T x+\tfrac{1}{2}x^T G x, \qquad (3)$$

where the gradient $g(x^i) = g^i$. Thus, the similarity between (1) and (2) is obvious if the directions s^i in (1) are conjugate.

Stewart (1973) introduced a generalization of the notion of conjugacy leading to a variety of finitely terminating iterations for solving systems of linear equations. We have found that an adaptation of Stewart's ideas to minimization problems establishes a similar generalization of the conjugate direction algorithm for function minimization.

We note that the definition of conjugacy can also be phrased as follows. If the vectors $u^o, u^1, \dots, u^{n-1}$ are the columns of an n×n matrix U, then they are A-conjugate if $U^T AU$ is diagonal. The generalization is achieved by introducing a second set of vectors $v^o, \dots, v^{n-1}$.

<u>Definition</u>: Let A,U and V be non-singular n×n matrices. Then (U,V) is an *A-conjugate pair* if $V^T AU$ is lower triangular.

The generalized algorithm for solving the equations $Gx+b=o$ is a slight variant of (2):

$$\begin{aligned} g^{k-1} &= Gx^{k-1}+b \\ \mu_{k-1} &= -\frac{{v^{k-1}}^T g^{k-1}}{{v^{k-1}}^T Gu^{k-1}} \qquad k=1;2;\dots;n, \qquad (4) \\ x^k &= x^{k-1}+\mu_{k-1}u^{k-1} \end{aligned}$$

where $U = [u^o, \dots, u^{n-1}]$ and $V = [v^o, \dots, v^{n-1}]$ form a G-conjugate pair.

Stewart developed an algorithm for constructing an A-conjugate pair as follows. Given non-singular matrices V,A and P, the vector u^k is determined as a linear combination of $p^o, p^1, \dots, p^k$ (k=o;...;n-1) such that U and V are A-conjugate. The resulting algorithm is:

$$\begin{aligned} u^o &= s_o p^o \\ u^1 &= s_1\left(p^1 - \frac{{v^o}^T Ap^1}{{v^o}^T Au^o}u^o\right) \\ &\vdots \\ u^k &= s_k\left(p^k - \frac{{v^o}^T Ap^k}{{v^o}^T Au^o}u^o - \frac{{v^1}^T Ap^k}{{v^1}^T Au^1}u^1 - \dots - \frac{{v^{k-1}}^T Ap^k}{{v^{k-1}}^T Au^{k-1}}u^{k-1}\right) \end{aligned}$$

The constants s_k are chosen to give u^k some predetermined scaling.

The analogous generalized conjugate direction method for the minimization of a function f(x) we formulate as follows. Suppose U and V form a conjugate pair:

$$\begin{aligned} x^o &= \text{arbitrary} \\ g^o &= g(x^o) \end{aligned}$$

For i=o;1;...,

$$\begin{aligned} x^{i+1^*} &= x^i+\alpha_i v^i, \text{ where } \alpha_i \text{ minimizes } f(x^i+\alpha v^i) \\ g^i &= g(x^i),\ g^{i+1^*} = g(x^{i+1^*}) \qquad (5) \\ \beta_i &= -\alpha_i \frac{{v^i}^T g^i}{{u^i}^T (g^{i+1^*}-g^i)} \\ x^{i+1} &= x^i+\beta_i u^i. \end{aligned}$$

Regarding this algorithm, it is possible to show that the β_i are equivalent to the μ_i in (4), as well as to prove the following theorem.

<u>Theorem</u>: If the iteration (5) is applied to the quadratic (3), where (U,V) form a G-conjugate pair, the minimum is found in at most n iterations and moreover, x^n lies in the subspace generated by x^o and $v^o, \dots, v^{n-1}$.

By varying the choice of the vectors v^i and p^i in the conjugation algorithm, one may therefore obtain from (5) various finitely terminating iterations for minimization. When applied to a general function the Hessian G can be eliminated from the conjugation algorithm by substituting $\frac{1}{\alpha_i}(g^{i+1^*}-g^i)^T$ for ${v^i}^T G$. Putting the scaling constants equal to 1, the choice of V=U reduces the conjugation algorithm to the Gram-Schmidt procedure and (5) becomes an ordinary conjugate direction algorithm. Finally, we can show that in this case variation of the vectors p^i leads to some well-known algorithms.

The Fletcher and Reeves (1964) algorithm: The conjugate directions in the basic iteration (1) are defined by

$$\begin{aligned} s^o &= -g^o, \\ s^k &= -g^k + \frac{{g^k}^T g^k}{{g^{k-1}}^T g^{k-1}} s^{k-1}, \quad k=1;2;\dots;n-1. \end{aligned}$$

If the columns of P are chosen successively, $P = [-g^o, -g^1, \dots, -g^{n-1}]$, the directions u^k in the generalized conjugation algorithm reduce to these s^k.

The Fletcher and Powell (1963) algorithm: Here the conjugate directions are defined recursively by

$$s^i = -H^i g^i,$$

where H^i is initially (i=0) any positive definite symmetric matrix, and thereafter

$$H^i = H^{i-1} + A^{i-1} + B^{i-1}$$

with $A^{i-1} = \dfrac{\beta_i s^{i-1} s^{i-1^T}}{s^{i-1^T}(g^i - g^{i-1})}$, $B^{i-1} = \dfrac{H^{i-1}(g^i - g^{i-1})(g^i - g^{i-1})^T H^{i-1}}{(g^i - g^{i-1})^T H^{i-1}(g^i - g^{i-1})}$, β_i being the steplength. Myers (1968) showed that if the initial direction is chosen as steepest descent (as is usually the case) the directions in this method are respectively scalar multiples of those in the Fletcher-Reeves method. Hence, theoretically the same choice of P would lead to this method.

The Smith (1962) and Powell (1964) algorithms without derivatives: Both these methods, Powell's being an improvement on Smith's, consist of basic computation cycles. We will only describe Powell's. To distinguish the vectors in the different cycles the cycle number will be used as vector subscripts.

Cycle 1: $r_1^{o} = e^1,\ r_1^{1} = e^2, \ldots,\ r_1^{n-1} = e^n. \quad x_1^{o} = \text{arbitrary}.$

For i=1;...; n, $x_1^{i} = x_1^{i-1} + \lambda_{1,i-1}\, r_1^{i-1}$, where $\lambda_{1,i-1}$ minimizes $f(x_1^{i-1} + \lambda r_1^{i-1})$

Cycle 2: $r_2^{o} = r_1^{1} = e^2,\ r_2^{1} = r_1^{2} = e^3, \ldots,\ r_2^{n-2} = r_1^{n-1} = e^n,$

$$r_2^{n-1} = x_1^{n} - x_1^{o} = \sum_{i=o}^{n-1} \lambda_{1,i}\, r_1^{i} = \sum_{i=o}^{n-1} \lambda_{1,i}\, e^{i+1}.$$

$x_2^{o} = x_1^{n} + \bar{\lambda}_1\, r_2^{n-1}$, where $\bar{\lambda}_1$ minimizes $f(x_1^{n} + \lambda r_2^{n-1})$

For i=1;...;n, $x_2^{i} = x_2^{i-1} + \lambda_{2,i-1}\, r_2^{i-1}$, where $\lambda_{2,i-1}$

minimizes $f(x_2^{i-1} + \lambda r_2^{i-1})$

$\vdots$

Cycle n: $r_n^{o} = r_{n-1}^{1} = e^n,\ r_n^{1} = r_{n-1}^{2} = \sum_{i=0}^{n-1} \lambda_{1,i}\, e^{i+1}, \ldots,$

$$r_n^{n-2} = r_{n-1}^{n-1} = \sum_{i=o}^{n-1} \lambda_{n-2,i}\, r_{n-2}^{i},\ r_n^{n-1} = x_{n-1}^{n} - x_{n-1}^{o} = \sum_{i=o}^{n-1} \lambda_{n-1,i}\, r_{n-1}^{i}$$

$x_n^{o} = x_{n-1}^{n} + \bar{\lambda}_{n-1}\, r_n^{n-1}$, where $\bar{\lambda}_{n-1}$ minimizes $f(x_{n-1}^{n} + \lambda\, r_n^{n-1})$

For i=1;...;n, $x_n^{i} = x_n^{i-1} + \lambda_{n,i-1}\, r_n^{i-1}$, where $\lambda_{n,i-1}$ minimizes $f(x_n^{i-1} + \lambda r_n^{i-1})$

After the n-th cycle the n directions are

$$s^{o} = r_n^{1} = \sum_{i=o}^{n-1} \lambda_{1,i}\, e^{i+1},\ s^{1} = r_n^{2} = \sum_{i=o}^{n-1} \lambda_{2,i}\, r_2^{i}, \ldots,$$
$$s^{n-1} = \sum_{i=o}^{n-1} \lambda_{n,i}\, r_n^{i},$$

which can be shown to be conjugate. These are the same as the u^k in the conjugation algorithm for the choice of

$$P = \left[\sum_{i=o}^{n-1} \lambda_{1,i} e^{i+1},\ \sum_{i=o}^{n-2} \lambda_{2,i} e^{i+2}, \ldots,\ \sum_{i=o}^{n-k} \lambda_{k,i} e^{i+k}, \ldots,\ \lambda_{n,o} e^n\right].$$

The Portan method of Shah, Buehler and Kempthrone (1964): Here the iteration is

$$x^1 = x^o - \mu_o g^o,$$

and for i=1;...;n,

$$z^i = x^i - \mu_i g^i,$$
$$x^{i+1} = z^i + \lambda_i (z^i - x^{i-1}),$$

where the μ_i and the λ_i are chosen by optimum line searches. We can write the iteration as

$$x^{i+1} = x^i + \frac{1}{m_i}\left[-g^i + n_{i-1}(x^i - x^{i-1})\right],$$

where $m_i = \frac{1}{\mu_i(1+\lambda_i)}$ and $n_{i-1} = \frac{\lambda_i}{\mu_i(1+\lambda_i)}$. Now, Rutishauer (1959) showed that the Fletcher-Reeves method can be written in exactly this way if

$$m_i = \frac{g^{i^T}Gg^i}{g^{i^T}g^i} - n_{i-1}$$

and if

$$n_{i-1} = m_{i-1}\frac{g^{i^T}g^i}{g^{i-1^T}g^{i-1}}\ .$$

The points x^i obtained by the Partan method would therefore be the same as those obtained by the Fletcher-Reeves method if the two definitions for m_i and n_{i-1} were the same. This is indeed the case; the directions are therefore theoretically the same in the two methods.

REFERENCES

FLETCHER, R. & Powell, M.J.D. (1963):A rapidly convergent descent method for minimization. Computer J. 6, 163-168.

FLETCHER, R. & Reeves, C.M. (1964): Function minimization by conjugate gradients. Computer J. 7, 149-154.

FOX, L., Huskey, H.D. & Wilkinson, J.H. (1948): Notes on the solution of algebraic linear simultaneous equations. Quart. J. Mech. Appl. Math. 1, 149-173.

HESTENES, M.R. & Stiefel, E. (1952): The method of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Standards 49, 409-436.

MYERS, G.E. (1968): Properties of the conjugate-gradient and Davidon methods. J. Optim. Th. Appl. 2, 209-219.

POWELL, M.J.D. (1964): An efficient method of finding the minimum of a function of several variables without calculating derivatives. Computer J. 7, 155-162.

RUTISHAUER, H. (1959): Theory of gradient methods in refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems. M. Engeli et al. (editors). Birkhäuser, Basel.

SHAH, B.V., Buehler, R.J. & Kempthrone, O. (1964): Some algorithms for minimizing a function of several variables. SIAM J. 12, 74-92.

SMITH, C.S. (1962): The automatic computation of maximum likelihood estimates. NCB Sc. Dept., report SC 846/MR/40.

STEWART, G.W. (1973): Conjugate direction methods for solving systems of linear equations. Numer. Math. 21, 285-297.

NON-LOCAL CONVERGENCE OF NEWTON-RAPHSON ITERATION

R. M. Walker
Department of Applied Mathematics
University of the Witwatersrand
Johannesburg

Whittaker and Robinson in Chapter Six of their book "Calculus of Observations" show that "if $f(x) = 0$ has a root between x_1 and x_2 and if $f'(x)$ and $f''(x)$ do not vanish in the interval $[x_1, x_2]$, then Newton-Raphson converges to the root if the iteration is started from the bound where $f(x)$ and $f''(x)$ have the same sign. The convergence is also monotonic". This is an example of non-local convergence.

A second example of non-local convergence is contained in the following theorem:

<u>Theorem 1</u>. Suppose two real values of x, x_A and X_B, exist $(x_B > x_A)$ such that $f(x_B) > 0$ and $f(x_A) < 0$ and that $f'(x) > 0$ for $x \in [x_A, x_B]$, and that

$$x_A = x_B - \frac{f(x_B)}{f'(x_B)} \quad \text{and} \quad x_B = x_A - \frac{f(x_A)}{f'(x_A)} . \tag{1.1}$$

[This means that when we apply the N-R iteration starting say with x_A we obtain $x_A, x_B, x_A, x_B, \ldots$ so that the iterative process may be said to "alternate". In any case it does not converge when either x_A or x_B are taken as initial values.]

Suppose also that $f''(x)$ vanishes once only in the interval $[x_A, x_B]$, and that the zeros of $f'(x)$ are all real. Then the N-R iteration converges if we start with an initial value of x lying inside the interval $[x_A, x_B]$.

<u>Proof</u>: <u>Case 1</u>. Suppose $f(X) = 0$ and $f''(X) = 0$ where X is the required root in the interval. [see FIGURE 2]. For $x > X$ $f''(x) < 0$ and for $x < X$ $f''(x) > 0$. Let x_1 be less than X_B and greater than X, and let

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad \text{and} \quad x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} .$$

We first prove $x_3 < x_1$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = \phi(x_2) = \phi\{\phi(x_1)\} \quad [\phi(x) = x - \frac{f(x)}{f'(x)}]$$

$$\therefore \quad \frac{dx_3}{dx_1} = \frac{d\phi(x_2)}{dx_2} \cdot \frac{d\phi(x_1)}{dx_1} = \frac{f(x_2)f'(x_2)f(x_1)f''(x_1)}{f'(x_2)^2 \; f'(x_1)^2} \tag{1.2}$$

$$\therefore \quad \frac{dx_3}{dx_1} \geq 0.$$

For $x_1 = X$, $\frac{dx_3}{dx_1} = 0$ and for $x_1 = X + \alpha$ we can clearly choose α so that $\frac{dx_3}{dx_1} < +1$.

Also $$\frac{d^2x_3}{dx_1^2} = \frac{d^2\phi(x_2)}{dx_2^2}\left[\frac{d\phi(x_1)}{dx_1}\right]^2 + \frac{d\phi(x_2)}{dx_2}\,\frac{d^2\phi(x_1)}{dx_1^2}$$

$$\frac{d^2\phi(x)}{dx^2} = \frac{f''(x)}{f'(x)} + \frac{f(x)f'''(x)}{f'(x)^2} - \frac{2f(x)f''(x)^2}{f'(x)^3}$$

$$= \frac{f''(x)}{f'(x)} - \frac{f(x)f''(x)^2}{f'(x)^3} - \frac{f(x)}{f'(x)^3}[f''(x) - f'(x)f'''(x)] \tag{1.3}$$

If the roots of $f'(x) = 0$ are all real it can be shown that $f''(x)^2 - f'''(x)f'(x) > 0$. This can then be used to shown that

$$\frac{d^2\phi(x_2)}{dx_2^2} > 0 \quad \text{and} \quad \frac{d^2\phi(x_1)}{dx_1^2} < 0 \tag{1.4}$$

so that

$$\frac{d^2x_3}{dx_1^2} \geq 0. \tag{1.5}$$

We can now sketch a rough graph of x_3 against x_1.

[See FIGURE 3.]

For $x_1 = x_B - \varepsilon$, $\frac{dx_3}{dx_1}$ becomes very large and tends to $+\infty$ as $\varepsilon \to 0$. If therefore we plot two graphs $x_3 = \phi\{\phi(x_1)\}$ and $x_3 = x_3' = x_1$, i.e. we have reached the point where the iterative process alternates $(x_3 = x_1 = x_B)$

$$\therefore \quad \text{for } x_1 < x_B$$
$$x_3 < x_1$$

Similarly it can be shown that $x_4 > x_2$.

We now have a monotonic decreasing sequence $x_1, x_3, x_5, \ldots$ bounded below, and a monotonic increasing sequence $x_2, x_4, x_6, \ldots$ bounded above. They cannot tend to two different limits since that would imply another alternating cycle which from FIG. 2 is impossible. Therefore both sequences tend to the same limit, namely the root X.

<u>Case 2</u>. The more general case of $f''(X) \neq 0$. Let $f''(x_i) = 0$. For x_1 lying between X and x_i N-R converges in the monotonic sequence of Whittaker and Robinson.

As before $$\frac{dx_3}{dx_1} = \frac{f(x_2)f''(x_2)f(x_1)f''(x_1)}{f'(x_2)^2 f'(x_1)^2}$$

$\frac{dx_3}{dx_1} = 0$ (a) when $x_1 = X$, (b) when $x_2 = X$, (c) when $x_1 = x_i$. Thus the graph of x_3 against x_1 has three turning points. The maximum of these three is at $x_1 = x_i$ and clearly $x_3(x_i) < x_1 = x_i$.

[See FIG 4 and FIG 5.]

When x_1 is chosen so that $x_2 = X$, clearly $\frac{dx_3}{dx_1} = 0$ and $x_3 = X$. When x_1 is greater than this value but less than x_B we can show as in Case 1 that $\frac{d^2x_3}{dx_1^2} > 0$.

As before $x_3 < x_1$ when $x_1 < x_B$ and $x_1 > X$. We thus have a monotonic decreasing sequence $x_1, x_3, x_5, \ldots$ until the value of a lies in the interval $[X, x_i]$ when N-R converges by the Whittaker and Robinson case.

<u>Note</u>: If $f''(x)$ vanishes once only in the interval $[x_A, x_B]$ and if there are no other alternating pairs in the interval, then N-R converges if we start with a value of x lying within the interval $[x_A, x_B']$. In other words, it is not necessary then to assume that the zeros of $f'(x)$ are all real.

In 1946 Dr. E. Bodewig proved that, if the roots of an algebraic equation are real and distinct, Laguerre's iteration converges no matter what real value is taken for the starting point. An identical result is clearly not true for N-R iteration since for example the points for which $f'(x) = 0$ are points of non-convergence. However a similar proposition which the author thinks is true but which he has been unable to prove in its entirety is the following:

<u>Hypothesis 2</u>: Suppose the roots of a polynomial are all real. Then the points of non-convergence of the Newton-Raphson iteration form a set of measure zero.

<u>Partial proof</u>: First, enumerate the basic points of non-convergence. The points for which $f'(x) = 0$ come into this category. [There will be $\leq(n-1)$ of them if the algebraic equation is of degree n.] Then the points for which N-R alternates as in Theorem 1 are also points of non-convergence. These will also be finite in number as can be seen by eliminating x_A between the two equations (1.1). We can, however, also obtain points of non-convergence by an extension of the alternating phenomenon. e.g. suppose

$$x_B = x_A - \frac{f(x_A)}{f'(x_A)} = \phi(x_A); \quad x_C = \phi(x_B); \quad x_A = \phi(x_C); \tag{2.1}$$

Thus the points x_A, x_B and x_C form a repetitive cycle. Clearly we could have four or more points forming a repetitive cycle. All such points are points of non-convergence. By elimination, as with the 2-cycles, it is clear that there will be a finite number of points corresponding to each such repetitive cycle. The set of basic points of non-convergence is a countable set.

Consider now any one of these basic points and label it x_N. This point can be reached from a finite number of points x'_N which are the real roots of the equation

$$x'_N - \frac{f(x'_N)}{f'(x'_N)} = \phi(x'_N) = x_N .$$

Again each of these points can be reached from a finite set of points x''_N which are the real roots of $\phi(x''_N) = x'_N$. Continuing in this way it can be seen that associated with x_N there exists a countable infinity of points which lead by N-R iteration to the point x_N.

First number the basic points of non-convergence: $x_{11}, x_{21}, x_{31}, \ldots, x_{N1}, \ldots$. Then number the points described in the previous paragraph from which these basic points can be reached: $x_{N1}, x_{N2}, x_{N3}, \ldots, x_{NM}, \ldots$. This double sequence of points is clearly a countable set and is therefore of measure zero.

Consider however, a 3-cycle (x_A, x_B, x_C) [see (2.1)]. It is possible that an infinite N-R sequence exists $x_1, x_2, x_3, \ldots$ such that

$$\begin{aligned} &x_1, x_4, x_7, \ldots, \to x_A \\ &x_2, x_5, x_8, \ldots, \to x_B \\ &x_3, x_6, x_9, \ldots, \to x_C . \end{aligned} \qquad (2.2)$$

If this occurs then it will be possible to find finite intervals about x_A, x_B and x_C such that any initial value within any one of these intervals converges by N-R to the 3-cycle. In which case Hypothesis 2 is false.

We must therefore prove that these points in the various cycles are points of <u>repulsion</u> with respect to direct Newton-Raphson iteration. It can easily be shown that for the case of the 3-cycle this implies

$$\left| \frac{f(x_A)f''(x_A)f(x_B)f''(x_B)f(x_C)f''(x_C)}{f'(x_A)^2 \; f'(x_B)^2 \qquad f'(x_C)^2} \right| > 1 \quad .$$

Analogous inequalities should hold for the other cycles. We have only been able to prove the Hypothesis for the quadratic and cubic equations. The following are the lines of an attempted proof.

Consider first the 2-cycles. Clearly the ones described in Theorem 1 are points of repulsion with respect to Newton-Raphson. Consider however a 2-cycle such as that in FIGURE 5. For points x_1 and x_A. As $x_1 \to x_D$, $x_3 \to X_1$. As $x \to x_C-$, $x_3 \to +\infty$. Suppose that there are more than one intersection with the line $x'_3 = x_1$.

$$\Phi'(x'_A) > 1 \quad \Phi'(x''_A) < 1.$$

But from (1.4) $\Phi('(x''_A) > \Phi'(x'_A)$, we have a contradiction and there is only one 2-cycle in this region.of the graph. Also $\Phi'(x_A) > 1$. Therefore the 2-cycle is repulsive to N-R iteration.

Consider now a 3-cycle. [see FIGURE 7(b)]. Form an inverse Newton-Raphson sequence $x_1, x_2, x_3, x_4, \ldots$ starting with x_1 such that $f'(x_1) = 0$. Then $x_4 > x_1$; $x_1 < x_7 < x_4$; $x_7 < x_{10} < x_4$ etc. i.e. in the sequence $x_1, x_4, x_7, x_{10}, \ldots$. each point lies in the interval formed by the previous two points. In other words, the sequence $x_1, x_7, x_{13}, \ldots$ is monotonic increasing and bounded above, while the sequence $x_4, x_{10}, x_{16}, \ldots$ is monotonic decreasing and bounded below. If we could prove that these two sequences tended to the same limit, that limit would be a point in the 3-cycle and we would have shown that the 3-cycle is attractive with respect to inverse N-R iteration (and therefore repulsive with respect to direct N-R iteration.) However, it is possible that the sequence converges to a 6-cycle, and the author has been unable to prove that this is not the case. If in fact the sequence does converge to a 6-cycle, then within that cycle there will either be another 6-cycle or the 3-cycle repulsive with respectto inverse N-R iteration. Thus in order to prove the Hypothesis 2 one must show that the above sequence converges to the 3-cycle. One can, however, prove that the 3-cycle is *unique*. For, suppose another 3-cycle exists in the same regions of the real axis. Let the given cycle be (x_A,x_B,x_C) and the second one be (x'_A,x'_B,x'_C) with $x'_A < x_A$. Then $x'_B > x_B$, $x'_C < x_C$ and $x'_A > x_A$ which is a contradiction. Therefore the original 3-cycle (x_A,x_B,x_C) is unique.

The same sort of analysis can be applied to any n-cycle where n is *odd*.

Consider now a 4-cycle [FIGURE 7(c)] As before form an inverse N-R sequence $x_1,x_2,x_3,\ldots$ starting with x_1 such that $f'(x_1) = 0$. Clearly $x_1,x_5,x_9,\ldots$ is a monotonic increasing sequence bounded above and therefore has a limit x_A (say) which is a point in the 4-cycle. Thus we have shown that a 4-cycle exists which is attractive with respect to inverse N-R iteration. However we cannot as with the 3-cycle show that this 4-cycle is unique with respect to the regions within which the points of the 4-cycle lie. In fact if it is not unique, then one can show that at least another two 4-cycles exist, one of which is repulsive, the other attractive with respect to inverse N-R. Thus to prove Hypothesis 2 one must show that the 4-cycle is unique.

The same sort of analysis can be applied to any n-cycle where n is *even*.

We can show however that Hypothesis 2 is true for quadratics and cubics whose roots are all real. The proof for quadratics is elementary and will not be given.

Theorem 3 Suppose the roots of a cubic equation are all real. Then the points of non-convergence of Newton-Raphson iteration form a set of measure zero.

Proof: First consider the case where the three roots X_1,X_2 and X_3 are real and distinct. $[X_1 < X_2 < X_3]$. Between the two zeros X'_1 and X'_2 of $f'(x)[X'_1 < X'_2]$ there exists a 2-cycle (X_A,X_B) which by Theorem 1 is unique. Again by Theorem 1, starting with a value of x lying inside the interval $[X_A,X_B]$, N-R converges to the root X_2. Also, starting with a value of x within the interval $[X_B,X'_2]$, say x_1, we have a sequence $x_1,x_2,x_3,\ldots$ such that $x_3 > x_1$; $x_4 < x_2$ etc until the odd or even term lies outside one of the intervals $[X'_2,+\infty)$ or $(-\infty,X'_1]$ in which case N-R iterates to X_3 or X_1 respectively. The same argument applied if we start with a value of x lying within the interval $[X'_1,X_A]$. Finally starting with a point within $[X'_2,+\infty)$ iterates to X_3, and with a point within $(-\infty,X'_1]$ iterates to X_1. We have of course excluded points which iterate to X'_1 or X'_2, but as shown in the discussion of Hypothesis 2 these points form a set of measure zero. Also X_A and X_B are isolated points i.e. no other points in the X-axis iterate to them.

This completes the proof. When either two of the roots of the cubic are equal or when all three are equal, it is easy to show that the theorem still holds.

To illustrate the relevance of the condition that the roots be real, consider a cubic equation with only one real root, namely

$$f(x) = x^3 - 2x + 2 = 0.$$

It can easily be shown that $x_A = 0$; $x_B = +1.0$ constitutes a 2-cycle for this cubic.

Also $f''(x_A) = f''(0) = 0$.

Therefore

$$\frac{f(x_A)f''(x_A)f(x_B)f''(x_B)}{f'(x_A)^2 \quad f'(x_B)^2} = 0 < 1 .$$

Thus this 2-cycle consists of points which are points of attraction with respect to direct N-R iteration. Small intervals about x_A and x_B exist for which the N-R sequence converges to this 2-cycle. To illustrate this consider the following N-R iteration sequence:

$x_1 = 0.01$; $x_2 = 1.00015$; $x_3 = 0.00000081$;
$x_4 = 1.0000012$ etc.

Clearly this N-R sequence is converging to the pair $x_A = 0$; $x_B = 1.0$. There is thus a set of non-zero measure for which N-R does not converge to the real root of $x^3 - 2x + 2 = 0$.

However a further fact of interest is that sometimes the points of non-convergence of Newton-Raphson form a set of zero measure even when all the roots of the algebraic equation are not real. For example, with the binomial equation

$x^{2n} - a = 0 (n > 1; a > 0)$ this is true, despite the fact that the equation has $(2n-2)$ complex roots.

The conditions under which the points of non-convergence of N-R form a set of zero measure clearly require further investigation.

The non-local convergence of, for example the other Schroder iterations and also of the Laguerre iteration will also merit some attention.

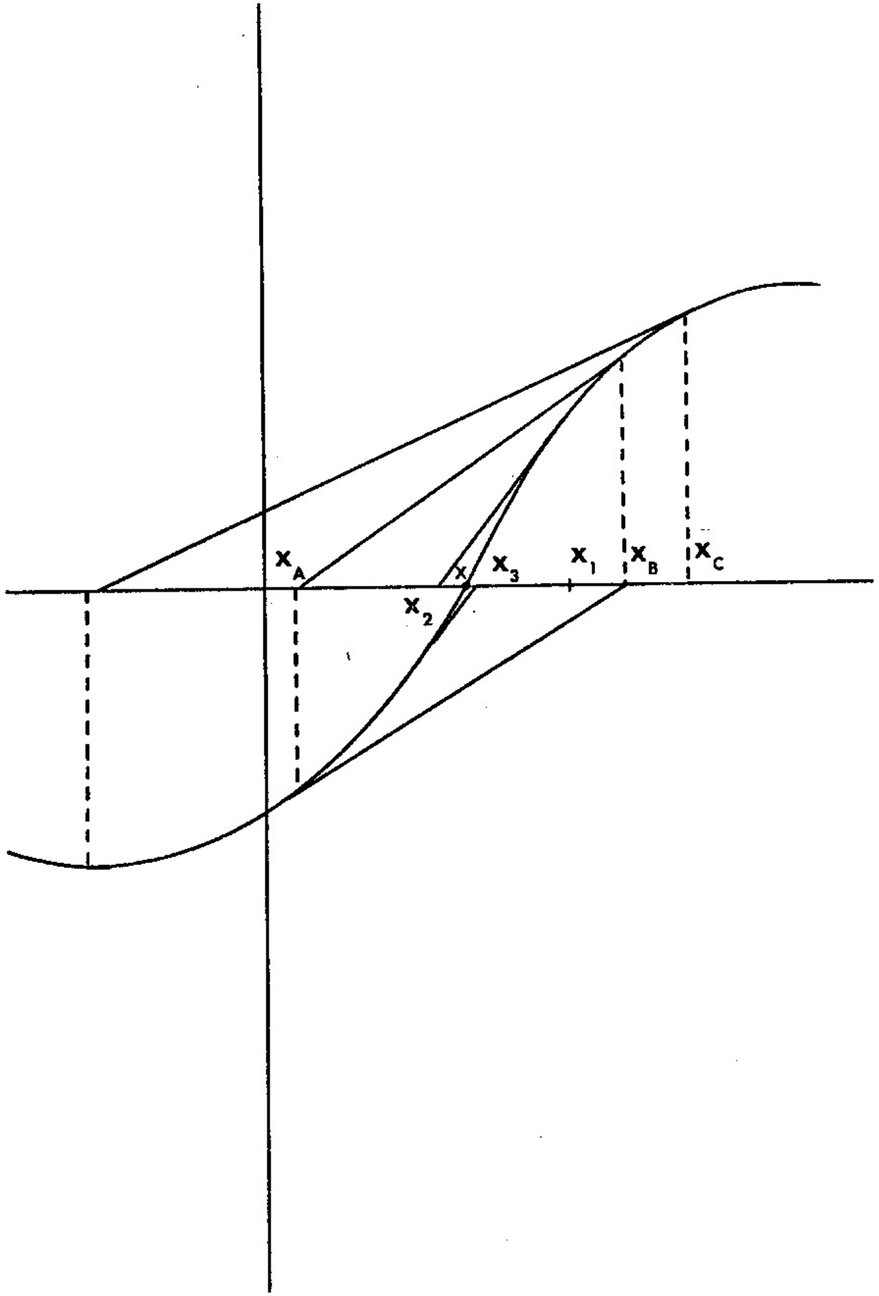

FIGURE 2

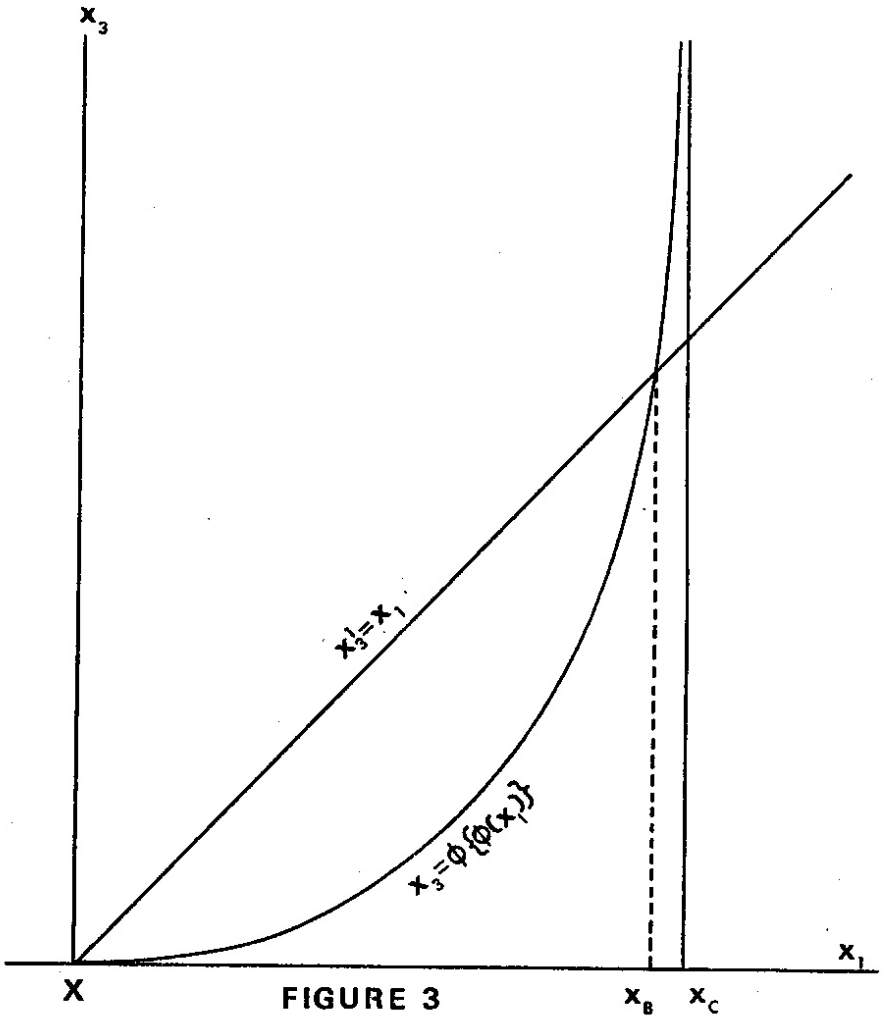

FIGURE 3

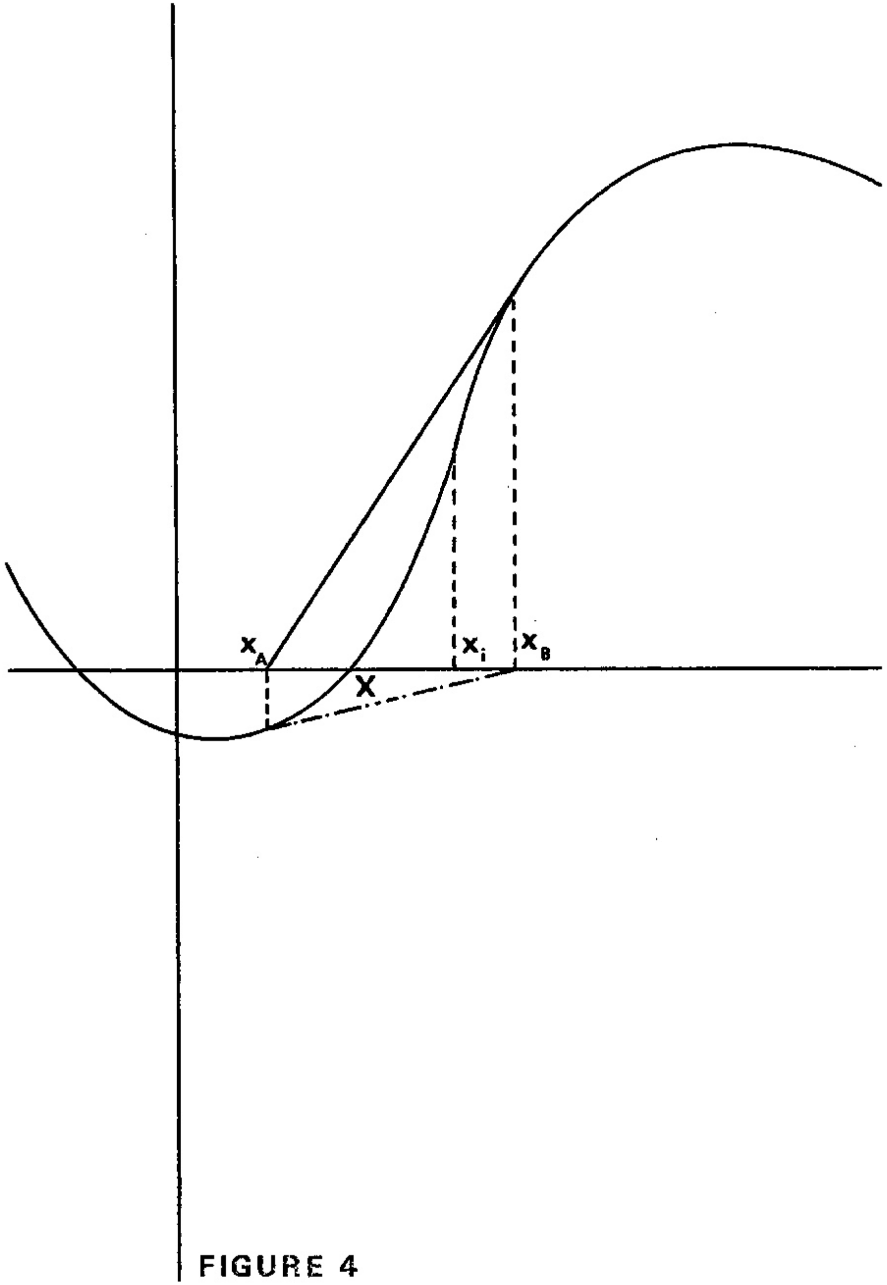

FIGURE 4

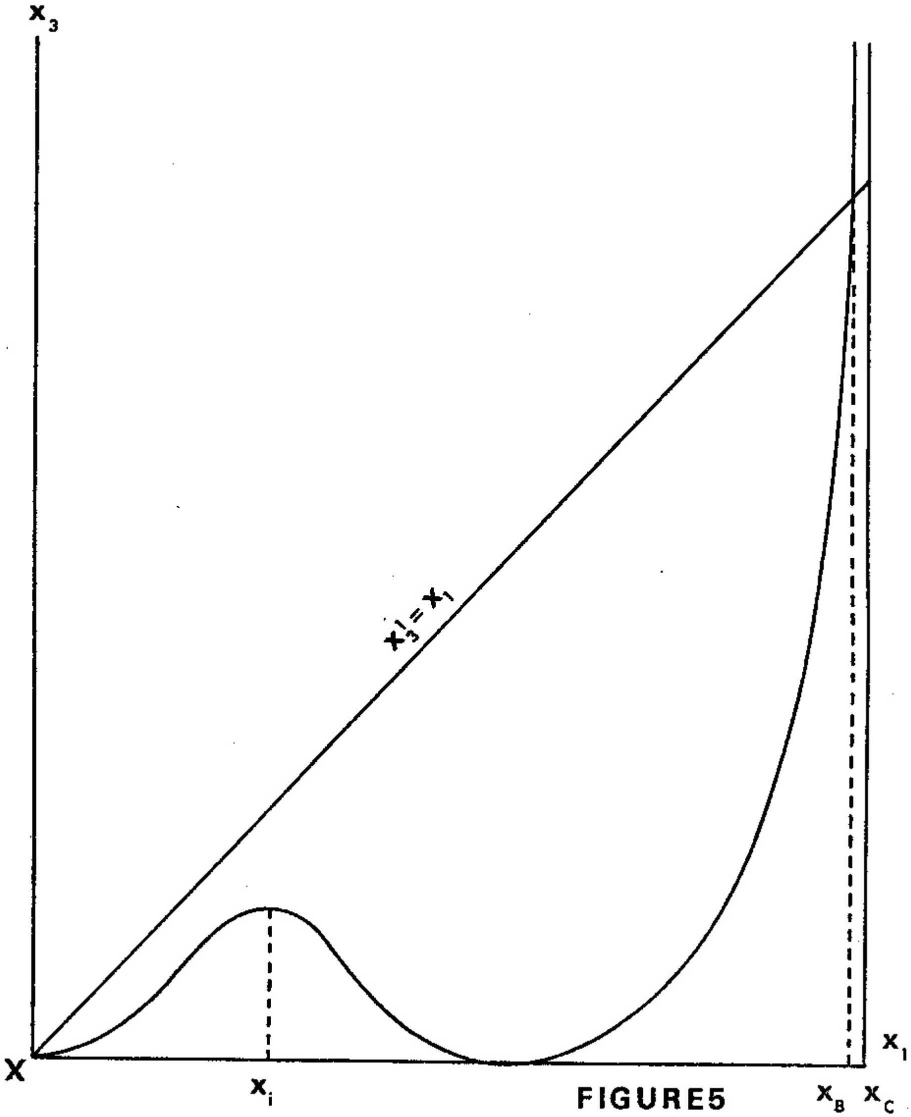

FIGURE5

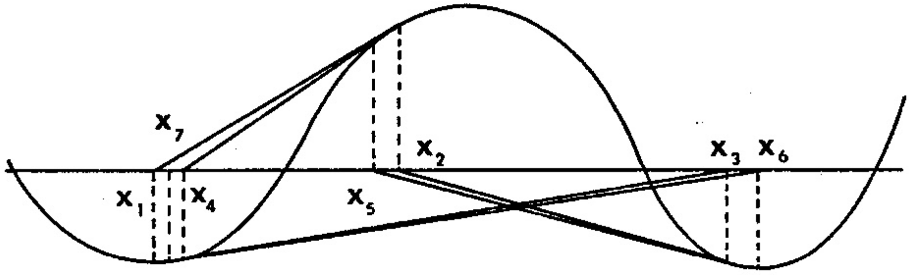

FIGURE 7(b)

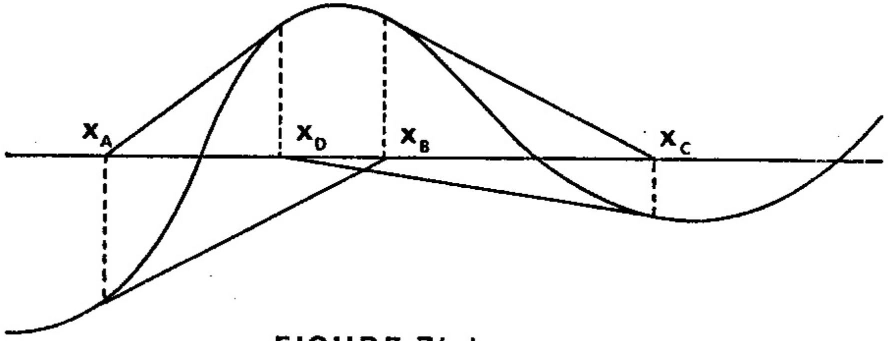

FIGURE 7(c)

STIFF ORDINARY DIFFERENTIAL EQUATIONS AND THEIR NUMERICAL SOLUTIONS

H. W. Kropholler
Department of Polymer and Fibre Science
University of Manchester Institute of Science and Technology

E. T. Woodburn
Department of Chemical Engineering
University of Natal
Durban

Introduction

The simulation of dynamic systems plays an important part in engineering and technological studies. This requires the solution of sets of ordinary differential equations of the form

$$\dot{y}_i = f_i(\underline{y},t) \qquad i = 1,2,\dots,N \qquad (1)$$

In this paper it is assumed that the functions f_i are smooth and well-behaved such that $\partial f_i/\partial x_j$ exists for all i and j.

These sets of equations can be solved using

(i) An Analog Computer
(ii) A Digital Computer
(iii) A Digital Differential Analyser.

The analog computer is theoretically the most satisfactory device because of speed, however, limited accuracy and inflexibility make the digital computer a much more attractive proposition. The digital computer suffers from the major disadvantage of its slow speed. Even very fast machines such as the CDC7600 could take several hours for the simulation of a process which could, in theory, be solved in minutes on the analog computer. It is the purpose of this paper to show that modern numerical methods allow the digital to approach the speed of the analog computer. The digital differential analyzers are hardware integrating devices that may in the future prove to be a replacement for software.

Stiff differential equations

As we assumed that the functions would be smooth well-behaved functions, we can illustrate stiffness by local linearisation of our differential equations.

$$\begin{aligned}\dot{\underline{y}} &= \underline{F}(\underline{y},t)\\ &\doteq \underline{F}\left(y(t_0),t_0\right) + \frac{\partial \underline{F}(\underline{y},t)}{\partial \underline{y}}\,\underline{\Delta y}\\ &+ \frac{\partial \underline{F}(y,t)}{\partial t}\,\Delta t\end{aligned}$$

Let $A = \dfrac{\partial F}{\partial y}$ and the eigenvalues of A will determine the characteristics of the solution.

The solution will be given by an approximation to the exponential matrix, exp(Ah)

$$\exp(Ah) \doteq \sum_{i=0}^{k} (Ah)^n/_{n!}$$

where k is the order of the method used for integrating the differential equations.

Accuracy (1) is given by $\varepsilon > ||(Ah)||^{k+1}/k!$ where ε is some small number and $||Ah|| \leq k$.

Stability is given by $\left|\left| \sum_{i=0}^{k} (Ah)^i/i! \right|\right| < \xi$

where ξ is 2 for a first order method and slowly increases for higher order methods (1).

A very simple problem posed by Rosenbrock and Storey (2) very clearly indicates the numerical difficulties

$$\begin{pmatrix}\dot{y}_1\\ \dot{y}_2\end{pmatrix} = \begin{pmatrix}-1000 & 0\\ 0.999 & -1\end{pmatrix}\begin{pmatrix}y_1\\ y_2\end{pmatrix}$$

$$\underline{y}^T(0) = \left(1.000,\ 0,999\right)$$

The solution is given by

$$y_1(t) = \exp(-1000t)$$
$$y_2(t) = e^{-t} - \left(\exp(-1000t)\right)/1000$$

After t = 0.002 the second term of y_2 makes a negligible contribution to the solution.

The accuracy criterion can be re-written $||\lambda h||^{k+1}/k!$ and for the above problem

$\lambda = -1$. Let $k = 1$, then $h^2 < \varepsilon < 10^{-3}$ for 0.1 % accuracy ($h = 0.03$)

but the stability requires that $1000h < 2$ and $h < 0.002$ i.e. 15 times smaller than the accuracy requirements warrant.

Conventional Numerical methods

The usual explicit methods are the Euler method which as a first-order method is extremely simple to program, the Runge-Kutta which may be summarized as

$$\underline{k}_1 = h\underline{F}(\underline{y}_{n-1}, t_{n-1})$$

$$\underline{k}_2 = h\underline{F}(y_{n-1} + \underline{k}_{1/2}, t_{n-1} + h/2)$$

$$\underline{k}_3 = h\underline{F}(y_{n-1} + k_{2/2}, t_{n-1} + h/2)$$

$$\underline{k}_4 = h\underline{F}(y_{n-1} + \underline{k}_3, t_{n-1} + h)$$

$$\underline{y}_n = \underline{y}_{n-1} + (\underline{k}_1 + 2\underline{k}_2 + 2\underline{k}_3 + \underline{k}_4)/6$$

and the Kutta Merson (3)

$$\begin{aligned}
\underline{k}_1 &= h\underline{F}(\underline{y}_{n-1}, t_{n-1}) \\
\underline{k}_2 &= h\underline{F}(\underline{y}_{n-1} + \underline{k}_{1/3}, t_{n-1} + h/3) \\
\underline{k}_3 &= h\underline{F}(\underline{y}_{n-1} + \underline{k}_{1/6} + \underline{k}_{2/6}, t_{n-1} + h/3) \\
\underline{k}_4 &= h\underline{F}(\underline{y}_{n-1} + \underline{k}_{1/8} + 3\underline{k}_{3/8}, t_{n-1} + h/2) \\
\underline{k}_5 &= h^{F}(\underline{y}_{n-1} + \underline{k}_{1/2} - 3\underline{k}_{3/2} + 2\underline{k}_4, t_{n-1} + h) \\
\underline{y}_n &= \underline{y}_{n-1} + (\underline{k}_1 + 4\underline{k}_4 + \underline{k}_5)/6 \\
\underline{e}_n &= (2\underline{k}_1 + 9\underline{k}_3 + 8k_4 - k_5)/30
\end{aligned}$$

This latter method is particularly interesting because it allows for a calculation of the error, hence automatic step control is possible.

It should be noted that for the linear system

$$\dot{\underline{y}} = A\underline{y}$$

$$\underline{y}_n = \left(I + Ah + \frac{A^2h^2}{2} + \frac{A^3h^3}{3!} + \frac{A^4h^4}{4!} + \left(\frac{5}{6}\right) \frac{A^5h^5}{5!}\right) y_{n-1}$$

and $$\underline{e}_n = \left(\frac{A^5h^5}{5!}\right) /_6$$

so that as well as an error estimate the method should be slightly better than a fourth order method. These latter two methods are particularly useful if it is necessary to write your own integration routine.

A number of unusual methods have been generated in attempts to deal with the stiff problem, for example in the linear case it is possible to formulate the equations as

$$\dot{\underline{y}}_i = -a_{ii}y_i + \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{ij}y_j$$

and then a numerical approximation can be arrived at using the concepts of a discrete time Markov process (4). Attempts to use this method for the general problem were not successful.

A fairly successful procedure was produced by Treanor (5) based on the assumption that the solution will be of an exponential form.

$$\begin{aligned}
\dot{y} &= f_{(y,t)} \doteqdot - P(y-y_{n-1}) + A + B(t-t_{n-1}) \\
&\quad + \tfrac{1}{2}C(t-t_{n-1})^2 \\
y_n &= y_{n-1} + h(AF_1 + BhF_2 + Ch^2F_3) \\
F_o &= e^{-Ph};\ F_n = \frac{F_{n-1} - {}^1/(n-1)!}{(-Ph)}
\end{aligned}$$

This method may well be significantly better than the Runge-Kutta or the Kutta-Merson.

Implicit methods are far more popular today, largely because they allow for a reasonable error prediction and they will allow for a somewhat larger increment of the independent variable. The simplest predictor corrector has the form

Predictor $\quad \underline{y}_{n_1(o)} = \underline{y}_{n-1} + h\underline{F}(\underline{y}_{n-1}, t_{n-1})$

Corrector $\quad \underline{y}_{n_1}(m+1) = \underline{y}_{n-1} + hF\{(1-\theta)\underline{y}_{n-1} + \theta\underline{y}_n(m), t_n + h\theta\}$

If $\theta = \frac{1}{2}$ we have the Crank-Nicolsen procedure, for $\theta > \frac{1}{2}$ the method should be completely stable.

The procedure is, for the linear case, equivalent to the following expansion

$$\begin{aligned}\exp\,(Ah) &= \exp\{(1-\theta)Ah\}\exp\{-(-Ah\theta)\} \\ &\doteq \{I + (1-\theta)Ah\}\{I - \theta Ah\}^{-1}\end{aligned}$$

Distefano (6) has tested a large number of numerical methods and shown that the explicit and implicit methods give much the same results. (A slight discrepancy between the theoretical and calculated stability limits is due to the use of an incorrect physical model.)

<u>Gear's method</u>

A good method for the numerical solution of ordinary differential equations should have the following properties

(i) Ability to handle stiff problems

(ii) Automatic error estimation and control

(iii) Ability to change step length easily and to print out at convenient values of t.

(iv) Easy to use.

Gear (7,8) developed a routine which satisfies most of the requirements and it is worth looking at the essential features of his method. The ordinary and stiff methods are formulated as given below.

Predictor $\quad \underline{y}_{n,(o)} = \underline{y}_{n-1} + \beta_1 h\dot{\underline{y}}_{n-1} + \dots + \beta_k h\dot{\underline{y}}_{n-k}$

Corrector $\quad \underline{y}_{n},(m+1) = \underline{y}_{n-1} + \beta_0^* h\underline{F}(y_n(m), t_n) + \beta_1^* h\dot{y}_{n-1} + \dots$
$\quad + \beta_{k-1}^* h\dot{y}_{n-k+1}$

and the truncation error is given by $e_{k+1}^A\, h^{k+1}\, y^{(k+1)}$.

The formulation for stiff methods is similar

Predictor $\quad \underline{y}_{n,(o)} = \alpha_1\underline{y}_{n-1} + \dots + \alpha_k y_{n-k} + \eta_1 h\dot{\underline{y}}_{n-1}$

Corrector $\quad \underline{y}_{n',(m+1)} = \alpha_1^*\underline{y}_{n-1} + \dots + \alpha_k^* y_{n-k} + \eta_0^* h\underline{F}(y_{n_1}(m), t_{n_1})$

The truncation error is $h^{k+1}y^{(k+1)}(t_n)/(k+1)$

In order to obtain a formulation of the predictor corrector equations suitable for changing the step length it is convenient to represent the equations in matrix form.

The procedure is as follows for the Adams Moulton predictor-corrector. Subtract the corrector equation from the predictor equation and define

$$\delta_i = (\beta_i - \beta_i^*)/\beta_0^* \qquad \text{with } \beta_k^* = 0$$

then

$$y_{n,(1)} = y_{n,(o)} + \beta_0^*\{hf(y_{n,(o)}, t_n - h\dot{y}_{n,(o)})\}$$

where

$$h\dot{y}_{n,(o)} \overset{\Delta}{=} \delta_1 h\dot{y}_{n-1} + \dots + \delta_k h\dot{y}_{n-k}$$

and if

$$h\dot{y}_{n,(m)} \overset{\Delta}{=} hf(y_{n,(m-1)}, t_n) \qquad m \geq 1$$

Then the corrector equation can be written as

$$y_{n,(m+1)} = y_{n,(m)} + \beta_0^*\{hf(y_{n,(m)}, t_n) - h\dot{y}_{n,(m)}\}$$

The predictor can now be written in the following matrix form

$$\begin{pmatrix} y_{n,(o)} \\ h\dot{y}_{n,(o)} \\ h\dot{y}_{n-1} \\ \vdots \\ h\dot{y}_{n-k+1} \end{pmatrix} = \begin{pmatrix} 1 & \beta_1 & \beta_2 & \cdot & \cdot & \cdot & \beta_{k-1} & \beta_k \\ 0 & \delta_1 & \delta_2 & \cdot & \cdot & \cdot & \delta_{k-1} & \delta_k \\ 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots \\ 0 & 0 & 0 & & & & 1 & 0 \end{pmatrix} \begin{pmatrix} y_{n-1} \\ h\dot{y}_{n-1} \\ h\dot{y}_{n-2} \\ \vdots \\ h\dot{y}_{n-k} \end{pmatrix}$$

or defining the vectors $\underline{V}_{n,(m)}$ and $\underline{V}_{n-1}$ and the matrix as B

$$\underline{V}_{n,(o)} = B\,\underline{V}_{n-1}$$

and the corrector equation combined with the equation

$$\begin{aligned} h\dot{y}_{n,(m+1)} &= hf\left\{y_{n,(m)}, t_n\right\} \\ &= h\dot{y}_{n,m} + \left\{hf(y_{n,(m)}, t_n) - h\dot{y}_{n,(m)}\right\} \end{aligned}$$

can be written in terms of the vector $\underline{V}$ as

$$\underline{V}_{n,(m+1)} = \underline{V}_{n,(m)} + \underline{c}F\left\{\underline{V}_{n,(m)}\right\}$$

where $\underline{c} = (\beta_0^*, 1, 0, \ldots 0)^T$ and

$$F\left\{\underline{V}_{n,(m)}\right\} = hf\left\{y_{n,(m)}, t_n\right\} - h\dot{y}_{n,(m)}$$

After m iterations set $\underline{V}_n = V_{n,(m)}$

For stiff methods we get much the same formulation by using the above procedure and now the predictor takes the form

$$\begin{pmatrix} y_{n,(o)} \\ h\dot{y}_{n,(O)} \\ y_{n-1} \\ y_{n-2} \\ \vdots \\ y_{n-k+1} \end{pmatrix} \qquad \begin{pmatrix} \alpha_1 & \eta_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{k-1} & \alpha_k \\ \gamma_1 & \delta_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_{k-1} & \gamma_k \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} y_{n-1} \\ h\dot{y}_{n-1} \\ y_{n-2} \\ y_{n-3} \\ \vdots \\ y_{n-k} \end{pmatrix}$$

The corrector takes the same form with

$$\underline{c} = (\eta_0^*, 1, 0, \ldots, 0)^T$$

But the predictor formulae are equivalent to fitting a kth degree polynomial through the known information carried in

$\underline{V}_{n-1}$. And, instead of saving information in this form, we will make a linear transformation Q such that

$$\underline{Z}_{n-1} = Q\underline{y}_{n-1}$$

The transformation Q is chosen so that k+1 components of $\underline{Z}_{n-1}$ are the function value y_{n-1} and the first k derivatives of the polynomials used in the prediction process. If the pth derivative in $\underline{Z}_{n-1}$ is scaled by $h^p/p!$, the matrix Q will be independent of h.

So

$$\underline{Z}_n = \left(y_n, h\dot{y}_n, \ldots, h^k y_n^{(k)}/k!\right)^T,$$

where $y_n^{(p)}$ is the pth derivative of the approximating polynomial.

$$\underline{Z}_{n,(o)} = Q\underline{y}_{n,(o)} = QBQ^{-1}\underline{Z}_{n-1}$$

$$\underline{Z}_{n,(m+1)} = Q\underline{V}_{n,(m+1)} = \underline{Z}_{n,(m)} + \ell F\left(Q^{-1}\underline{Z}_{n,(m)}\right)$$

where $\ell = Q\underline{c}$. Since both $\underline{Z}_n$ and $\underline{V}_n$ have y_n and $h\dot{y}_n$ as their first two components and F depends on these only

$$F(Q^{-1}\underline{Z}_n) = F(\underline{Z}_n)$$

$\underline{\ell}$ depends on the predictor corrector method used. The matrix QBQ^{-1} provides a kth order approximation to $\underline{Z}_{n,(o)}$ in terms of $\underline{Z}_{n-1}$; hence it is the Pascal triangle matrix for either method.

One difficulty remains in that the corrector only converges for small values of h and stiff methods require the use of large values of h.

In view of the equation

$$\underline{Z}_{n,(m+1)} = \underline{Z}_{n,(m)} + \ell F(\underline{Z}_n)$$

convergence is equivalent to solving the equation $F(\underline{Z}_n) = 0$ and

Newtons method may be applied. The successive approximations may be written as

$$\underline{Z}_{n,(m+1)} = \underline{Z}_{n,(m)} + \underline{\ell}\{-(\partial F/\partial Z)\cdot\underline{\ell}\}^{-1} F(\underline{Z}_{n,(m)}\}$$

where $\{-\partial F/\partial Z\cdot\underline{\ell}\}^{-1} = (\ell_1 - h\ell_0 \partial f/\partial y)^{-1}$

The error is controlled using a single parameter, ϵ, and the decision whether this is a relative or an absolute error control depends on the solution. If it is growing, a relative control is used; if it is decaying, an absolute control is used.

Gear's (8) program satisfies the first three criteria given above. It is not particularly easy to use, but there is no doubt that the program (8) is superior to any other available today.

A few minor alterations have been made to the program listed in (8). Considerable difficulty was found in the use of the array SAVE and this was converted to a one dimensional array.

The numerical differentiation was altered so that if the numbers are nearly the same the difference would be set to zero. This latter modification is particularly important if the original problem is unscaled.

Finally a completely revised version of the program has also been written incorporating a sparse matrix inversion routine so that up to 500 differential equations can be handled in a total of 80K core. (15)

Problems tackled using Gear's method.

(1) Parameter Estimation.

A simple problem presented by Chandler et al (9) involves the simulation of the reactions catalysed by bovine liver glutamate dehydrogenase. The object was to find values of the rate constants from experimental data using regression procedures. To make this approach viable, extremely fast solution of the set of differential equations was required. The reaction scheme is of the following form.

$$\begin{aligned}
E + A \quad &\underset{k_2}{\overset{k_1}{\rightleftharpoons}} \quad FA \\
EA + B \quad &\underset{k_4}{\overset{k_3}{\rightleftharpoons}} \quad EAB \\
EAB \quad &\underset{k_6}{\overset{k_5}{\rightleftharpoons}} \quad EP \\
EP \quad &\xrightarrow{k_7} \quad E + P
\end{aligned}$$

It is easy to show that there are four independent reactions and that three material balances must be satisfied. We define $(EP) \overset{\Delta}{=} y_1$; $(P) \overset{\Delta}{=} y_2$; $(EA) = y_3$; $(EAB) \overset{\Delta}{=} y_4$, then

$$\begin{aligned}
\dot{y}_1 &= -(k_6 + k_7)y_1 + k_5 y_4 \\
\dot{y}_2 &= k_7 y_1 \\
\dot{y}_3 &= k_1(E)(A) - k_2 y_3 - k_3 y_3 (B) + k_4 y_4 \\
\dot{y}_4 &= k_3 y_3 (B) - (k_4 + k_5) y_4 + k_6 y_1
\end{aligned}$$

$$\begin{aligned}
(A) &= (A_0) - y_1 - y_2 - y_3 - y_4 \\
(B) &= (B_0) - y_1 - y_2 - y_4 \\
(E) &= (E_0) - y_1 - y_3 - y_4
\end{aligned}$$

The properties of the problem can easily be seen from the Jacobian matrix

$$\frac{\partial \underline{F}}{\partial \underline{y}} = \begin{pmatrix} -(k_6 + k_7) & 0 & 0 & k_5 \\ k_7 & 0 & 0 & 0 \\ -k_1(A) & 0 & -\{k_1(A) + k_2 + k_3(B)\} & \{k_4 - k_1(A)\} \\ k_6 & 0 & k_3(B) & -(k_4 + k_5) \end{pmatrix}$$

In addition it turns out that (A) and (B) do not vary greatly and that using the values of the constants given by Chandler et al (9) the eigenvalues were found, using the Francis QR - algorithm.

$$\underline{\lambda} = (-5.12 \times 10^5, -8.8 \times 10^5, -2.38 \times 10^2, -6.43 \times 10^{-9})$$

As the solution must be found at a time equalling 0.196 the stiffness can be assessed as approximately $0.196/(8.8 \times 10^5)^{-1} \doteqdot 10^5$.

The effectiveness of Gear's routine can be seen from Table I below.

TABLE I

	Runge-Kutta	Treanor	Gear
No. of derivative evaluations on the interval t = 0,0.196	167681	6721	402

(2) Simulation of a flame reactor

Williams (10) proposed a partial kinetic scheme by means of which the partial combustion of methane or ethane may be modelled. Using Gear's method it becomes possible to judge the usefulness of the kinetic data given for the 35 reactions. Here the sheer weight of formulating the problem has to be considered and the data must be organised. We note that for a set of chemical reactions we have a number of molecular species which consist of a number of atomic species. These facts can be used to find a consistent set of compositions and the corresponding set of differential equations.

The procedure as outlined by Aris (11) is given below. Let A be a matrix of stoichiometric coefficients where a_{ij} refers to the ith reaction and the jth chemical species. For each reaction we can define an extent X_i and then the change in the number of moles is given by

$$\Delta N_j = \sum_i X_i a_{ij}$$

$$\underline{\Delta N}^T = \underline{X}^T A$$

and now the number of independent rows of the matrix A is found, by a linear transformation Q.

$$QA = \begin{pmatrix} I & c_{12} \\ \underline{0} & \underline{0} \end{pmatrix}$$

$$\underline{\Delta N}^T = \underline{X}^T Q^{-1} QA$$

Let $\underline{X}^T Q^{-1} = \underline{Z}$ and the number of moles of the independent components represented by the vector ΔN_I with the dependent components by ΔN_D

$$(\underline{\Delta N}_I^T, \underline{\Delta N}_D^T) = (z_I, z_I c_{12})$$

$$\underline{\Delta N}_D^T = \underline{\Delta N}_I^T c_{12}$$

So that we need only work with the independent subset of chemical species, ΔN_I. Except for the case of equilibrium, we <u>cannot</u> work with an independent subset of reactions. The statement by Aris that one can work with an independent subset of reactions is in fact incorrect.

The reaction rate set that we work with is given by

$$\dot{N}_I^T = \dot{\underline{Z}}_I^T = \dot{\underline{X}}^T Q^{-1} = \underline{r}^T Q^{-1}$$

The form of the detailed data given by Williams is shown in Table II.

TABLE II

No.	Reaction	$\log_{10} k_\infty$	E kJ mol^{-1}
1	$CH_4 \rightarrow CH_3. + H.$	15	436.8
2	$C_2H_6 \rightarrow 2CH_3.$	16.51	369.6
3	$2CH_3. \rightarrow C_2H_6$	11.12	0.0
4	$CH_3.+ O_2 \rightarrow CH_2O + OH.$	11.3	0.0
.	. .	.	.
.	. .	.	.
.	. .	.	.
35	$CH_3.+ O. \rightarrow CH_2O + H.$	14.11	8.4

Experimental evidence indicates that aldehydes (e.g. CH_2O in the above) promote the combustion reaction. Sufficient data exists in this paper to enable a comparison between experiment and theory to be made. At 600K the reaction rate constants are

$$(k_1, k_2, k_3, k_4, k_{35}) = (10^{-23}, 10^{-15.7}, 10^{11.12}, 10^{11.3}, 10^{12.5})$$

and from this it can be seen that the problem is a particularly stiff one. At time of writing no numerical results are available

(3) Simulation of large systems.

By large we are talking of systems with several hundred differential equations. Systems that come into this category are power station simulation, global economic and technological simulations (12), such as described in the book "Limits to Growth" and here we shall look at the simulation of distillation columns. The form of the equations are discussed by Holland (13).

The equations take the form

$$\dot{x}_{j,i} = \left(- (L_j + P_j)x_{j,i} - V_j y_{j,i} + L_{j-1}x_{j-1,i} + \right.$$
$$\left. + V_{j+1}\, y_{j+1,i} + F_j Z_{j,i}\right) / W_j$$

j is the number of stages and i is the number of components. The relationship between the vapour $(y_{j,i})$ and liquid $(x_{j,i})$ mole fractions is given by

$$y_{j,i} = \alpha_i\, x_{j,i} / \left(\sum_{k=1}^{r} \alpha_k\, x_{j,k}\right)$$

The modified version of Gear's program incorporating a sparse matrix inversion routine has been used here. By changing the number of components and the number of stages it is possible to alter the size of the problem.

TABLE III

No. of O.D.E.'s	Program 1 Storage	Time*	Program 2 Storage	Time	Program 3 Storage	Time
25	11K**	1	18K	1.4	12K	1.1
150	59K	34	36K	15	25K	12
300	-	-	80K	71	52K	47
500	-	-	-	-	82K	105+

* time is relative to the smallest

**storage given as 1000's of 24 bit words

+ actual computational time on a CDC 7600 was 40 seconds

Program 1	Gear's program as given in ref.(8)
Program 2	Modified Gear's program using sparse matrix inversion routine of Curtis & Reid (14)
Program 3	Modified Gear's program using Senior's (15) sparse matrix inversion routine.

Table III shows the storage requirements and relative speeds of different computer programs. Only the stiff method of Gear's program was used because the Adam's Moulton predictor corrector would be far too slow.

(4) Simulation of closed circuit grinding

In this case the objective was to look at the grinding operation in an attempt to check modifications in process design and look at different control strategies. In particular the model would be useful in the development of a multivariable control system.

The crushing process can be approximated by thinking of the mill as J consecutive stages in the direction of flow and N particle size classes in each stage of the mill. The dynamic equations for the solids flowing through the mill are

$$\dot{y}_{n,j} = F_{in} - F_{out} + Break_{in} - Break_{out}$$

$$F_{in} = G_{j-1} \frac{W_{n,j-1}}{\sum_n W_{n,j-1} + WW_{j-1}}$$

$$G_j = \alpha_j\left(\sum_n W_{n,j} + WW_j - \Sigma W_{n,j+1} - WW_{j+1}\right) \qquad j \neq J$$

$$\alpha_j = \alpha^0{}_j\left(1 - \left(\frac{\sum_n W_{n,j}}{\sum_n W_{n,j} + WW_j}\right)^\delta\right)$$

$$F_{out} = G_j \frac{W_{n,j}}{\sum_n W_{n,j} + WW_j}$$

The outflow from the final stage is mechanically controlled and assuming that the hold-up exceeds some quantity, W_{MIN}

$$Q = \Sigma W_{n,J} - W_{MIN} + WW_J$$

$$G_J = \alpha_J(2 \times 10^{-2}Q^2 - 2Q)$$

$$\text{Break}_{in} = \sum_{k=n+1}^{N} \delta_k B_{n,k} W_{k,j}$$

$$\text{Break}_{out} = \delta_n W_{n,j}$$

$B_{n,k}$ is the breakage matrix and must satisfy the condition that

$$\sum_{n=1}^{n=k-1} B_{n,k} = 1$$

$$\delta_n = 0.054(D_{n-1}/180.)^{(4.89 - 4\times\delta_{N-n})}$$

The equations for the water flowing through the mill are

$$\dot{y}_{N+1,j} = G_{j-1}\frac{WW_{j-1}}{\Sigma W_{n,j-1} + WW_{j-1}} - G_j\frac{WW_j}{\Sigma W_{n,j} + WW_j}$$

Some of the material is recycled and before returning to the mill it is classied in a cyclone so that only the large particles are returned to the mill. The equations describing the behaviour of the cyclone are given by Lynch (16).

With 5 stages and 5 particle size classes only 30 differential equations arise but using Gear's program in the original version resulted in an increase of speed of 20 to 25 times, so that 1 hr of mill simulation took approximately ½ an hour on a CDC 1700. The detailed description of the simulation is given in (17).

Conclusions

Gear's program has been run satisfactorily on some five different makes of computer and when used properly always given a considerable increase in speed for simulations in which we approach some sort of equilibrium or steady-state. A wide range of problems have been discussed in this paper. For the smaller problems single precision word-length of six figures has been found adequate. For the larger problems such as the distillation case a longer word length is required as referred to by Gear (8). There is no doubt that this routine enables the user to approach the speed of an analog computer with a considerable improvement in accuracy.

References

1. B.A. Buffham and H.W. Kropholler. "Evaluation of the Exponential Matrix" 1971 Joint Symposium I.Chem.E. - Brit. Comp. Society.

2. H.H. Rosenbrock and C. Storey. " Computational Techniques for Chemical Engineers" Pergamon Press, Oxford 1966.

3. L.A. Fox and D.F. Mayers. "Computing Methods for Scientists and Engineers" Clarendon Press, Oxford 1968.

4. L.G. Gibilaro, H.W. Kropholler and D.J. Spikins. Chem.Eng.Sci. 1967, <u>22</u>, 517.

5. C.E. Treanor. Maths of Comp. 1966, <u>20</u>, 39 - 45.

6. L.G. Distefano. A.I.Ch.E. J. 1968, <u>14</u>, 190 - 199.

7. C.W. Gear. Comm. ACM 1971, <u>14</u>, 176 - 179.

8. C.W. Gear. ibid 1971, <u>14</u>, 185 - 190.

9. J.P. Chandler, D.E. Hill and H.O. Spivey. Comp. and Biomedical research 1972, <u>5</u>, 515 - 534.

10. Williams. "Flame and Plasma Reactors" Trans. Instn. Chem.Engrs. 1973, <u>51</u>, 225.

11. R. Aris. "Elementary Chemical Reactor Analysis" Prentice-Hall 1969, Ch.2 p.8 - 27.

12. D.H. Meadows, D.L. Meadows, J. Randers, W.W. Behrens III "The limits to growth" Pan Books Ltd. 1974.

13. C.D. Holland. "Unsteady State processes with applications in Multicomponent Distillation".

14. A.R. Curtis and J.K. Reid. "Fortran subroutines for the Solution of large sparse sets of Linear Equations" UKAE Res. Group Rept R6844.

15. P.R. Senior. Ph.D. Loughborough University, 1975.

16. A.J. Lynch and T.C. Rao. "Modelling and scale-up of hydrocyclone classifiers" Tech. Rep. Julius Kruttschnitt Min. Res. Centre, Univ. of Queensland, 1 - 29 Jan - June 1974.

17. R.K. Jaspan, H.W. Kropholler, T. Mika and E. Woodburn. "An analysis of closed circuit wet-grinding mill control characteristics by simulation" to be read at 4th European Symposium on Comminution, September 1975.

AUTOMATIC STEP-SIZE CONTROL IN PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS

D.P. Laurie
National Research Institute for Mathematical Sciences,
CSIR, P.O. Box 395, Pretoria, 0001

ABSTRACT

When using a suitable implicit method to solve a parabolic equation, the step-size need not be restricted to satisfy stability requirements. As the faster components die out, the step-size can be increased without loss of accuracy. Using arguments from approximation theory, formulas are derived for finding suitable step-sizes to be used with various time-stepping methods. The global error is shown to be related to the stiffness ratio of the semi-discrete form of the parabolic equation. Numerical results are given.

This work forms part of research towards a Ph.D. degree at the University of Dundee under the direction of Prof. A.R. Mitchell.

1. INTRODUCTION

The question of automatic step-size control when solving a parabolic partial differential equation (PDE), arises only when implicit methods are used, since the inherent stability restriction on the step-size for an explicit method is usually also severe enough to guarantee accuracy that is commensurable with the spatial discretization error.

The problem can be thought of as related to adaptive quadrature and to the automatic solution of ordinary differential equations (ODEs). In the case of the former, we can learn a lesson from Rice [1975], who cautions against a proliferation of heuristic techniques and advocates a "metalgorithm", in which the problem is broken up into several independent sections, each of which can be analysed separately.

With this end in view, we separate the solution of the PDE into two phases by semi-discretization, following Varga [1962]. The application of one's favourite technique for elliptic equations to the space variables (e.g. finite differences, finite elements, parametric models) leaves one with a system of ODEs, typically of the following form

$$(1) \qquad BU' + AU = F, \qquad U(0) = U_o;$$

where B and A are n by n matrices, and U and F are vector-valued functions of the time variable t. Most of the classical difference formulas given by Richtmyer and Morton [1967] can be obtained

from (1) by applying a simple ODE formula such as the forward or backward Euler method or the trapezoidal rule.

One may ask why (1) cannot be solved by an automatic ODE solver such as that of Gear [1971]. If the problem is small enough and the computer large enough, such packages can in fact be used; see e.g. Murphy [1975]. When, however, the matrix B is large, sparse, and not diagonal (as occurs in finite element discretizations), the explicit inversion of B to bring (1) into canonical form should be avoided.

The a posteriori technique of error control that is used in many ODE routines could still be applied with any method that does not explicitly invert B. This technique consists of performing the calculation twice at each point of time, using first a single step of size Δt and then two steps of size $\frac{1}{2}\Delta t$. With the aid of asymptotic error analysis, one can now decide whether the error is too large, acceptable or too small, and the step-size can then be decreased, kept constant or increased, according to some (usually heuristic) strategy. This technique obviously has an overhead of at least 50%, but this is still less than the work required to obtained a priori error estimates in the general ODE case.

In the case of the system (1), however, we know that the exact solution is

$$(2) \qquad U(t) = \exp(-tB^{-1}A)[U_o + \int_o^t \exp(\tau B^{-1}A)B^{-1}F(\tau)d\tau].$$

When the order n of the matrices B and A is small, this formula can indeed also be used for computing U, by using the identity (applicable to diagonalizable matrices)

$$\exp(M) = Q^{-1}\mathrm{diag}(\exp \lambda_i)Q \tag{3}$$

where λ_i are the eigenvalues of M and Q is a non-singular matrix such that $M = Q^{-1}\mathrm{diag}(\lambda_i)Q$. In the parabolic case, we usually know that the eigenvalues of $B^{-1}A$ are simple and positive: upon suitable ordering we may write

$$0 < \lambda_1 < \lambda_2 < \ldots < \lambda_n. \tag{4}$$

It is clear that we have quite detailed information about the solution, and it becomes plausible to look at a priori error estimates.

The problem has now been essentially reduced to a problem in approximation theory, since we shall show in the next section that the approximation of a matrix exponential by a rational function is equivalent to the simultaneous approximation of the exponential function at each of the eigenvalues of the matrix. If we approximate the matrix exponential to a given absolute accuracy ε, we have in a sense approximated the solution to a relative accuracy ε, since the matrix exponential is multiplied by the given vectors U_o and F. The aim of the error analysis is to find an estimate for the error E as a function of the time step Δt. For the automatic step-size control strategy, this relation is then inverted to yield Δt as a function of the desired accuracy ε.

2. THE APPROXIMATION PROBLEM

We consider a rational approximation $r(t) = p(t)/q(t)$ to $\exp(-t)$, and define

$$r(M) = \{q(M)\}^{-1}p(M) \tag{5}$$

where M is as before. Clearly

$$\begin{aligned} r(M) &= \{q\{Q^{-1}\mathrm{diag}(\lambda_i)Q\}\}^{-1}p\{Q^{-1}\mathrm{diag}(\lambda_i)Q\} \\ &= \{Q^{-1}\mathrm{diag}\{q(\lambda_i)\}Q\}^{-1}\ Q^{-1}\mathrm{diag}\{p(\lambda_i)\}Q \\ &= Q^{-1}\mathrm{diag}\{r(\lambda_i)\}Q \end{aligned}$$

We thus obtain

THEOREM 1

$$\exp(-M) - r(M) = Q^{-1}\mathrm{diag}(\exp(-\lambda_i) - r(\lambda_i))Q.$$

Using the fact that $\|Q^{-1}\|\ \|Q\| = 1$, we obtain

COROLLARY

$$\begin{aligned} \|\exp(-M) - r(M)\| &\leq \min_i |\exp(-\lambda_i) - r(\lambda_i)| \\ &\leq \min_{\lambda\in[\lambda_1,\lambda_n]} |\exp(-\lambda) - r(\lambda)|\ . \end{aligned}$$

An interesting choice of r was made by Cody, Meinardus and Varga [1969]. They minimize the maximum error over the entire half-axis, which makes it unnecessary to estimate λ_1 and λ_n. The only drawback of their approach is that accuracy can only be increased by increasing the degree of r; the approximation cannot be used for time-stepping.

For a time-stepping method we need an approximation that will yield stable solutions. Such an approximation is called acceptable by Lambert [1973]. There are many different levels of acceptability, but for our purpose the following will suffice:

Definition

A rational approximation $r(t)$ to $\exp(-t)$ is called acceptable if it satisfies the following three criteria:

(7a) $$|\exp(-t) - r(t)| \leqslant 1, \qquad t \in [0,\infty);$$

(7b) $$|r(t)| \leqslant 1 \quad , \qquad t \in [0,\infty);$$

(7c) for all $\varepsilon > 0$ there exists $h > 0$ such that
$$|\exp(-t) - r(t)| \leqslant \varepsilon, \qquad t \in [0,h] .$$

In order to examine the use of a time-stepping approximation, we proceed as follows:

(8) $$\exp(-(t+\Delta t)M) = \exp(-tM)\ \exp(-\Delta tM)$$

$$\Rightarrow \ \|\exp(-(t+\Delta t)M) - \exp(-tM)r(\Delta tM)\|$$

$$\leqslant \|\exp(-tM)\|\ \|\exp(-\Delta tM) - r(\Delta tM)\|$$

(9) $$\leqslant \max_{\lambda \in [\lambda_1,\lambda_n]} \exp(-t\lambda)|\exp(-\Delta t\lambda) - r(\Delta t\lambda)| .$$

To make the righthand side of (9) less than ε, we put

(10) $$\tilde{\lambda} = \min(\lambda_n, \frac{\ell n(\frac{1}{\varepsilon})}{t})$$
$$t = \frac{h}{\tilde{\lambda}} .$$

This requires an estimate of the largest eigenvalue λ_n, which is usually easy to obtain by Gerschgorin's theorem or numerically by the power method.

We now have the following: given $\varepsilon > 0$ and any acceptable rational approximation $r(t)$, and knowing $\exp(-tM)$, we can find $\Delta t > 0$ such that $\exp(-(t+\Delta t)M)$ can be approximated to an accuracy of ε. The reader will have noted that the requirement of consistency has not yet been mentioned. Since lack of consistency is essentially a non-local phenomenon, which does not show up in a single time step, we postpone discussion of it to the next section.

3. GLOBAL ERROR

Let e_k be the computed approximation to $\exp(-t_kM)$. Then

$$\|\exp(-(t_k+\Delta t_k)M) - e_k\ r(\Delta t_kM)\|$$

$$\leqslant \|\exp(-(t_k+\Delta t_k)M \ - \exp(-t_kM)r(\Delta t_kM)\|$$

$$+ \|\exp(-t_kM) - e_k\|\ \|r(\Delta t_kM)\| \ .$$

By (7b), $\|r(\Delta t_kM)\| \leqslant 1$. Since the first term is less than ε, it follows by induction that

$$\|\exp(-t_kM) - e_k\| \leqslant k\ \varepsilon.$$

In the strategy (10) we take $\tilde{\lambda} = \lambda_n$ until $t \geqslant \frac{\ell n(\frac{1}{\varepsilon})}{\lambda_n}$. The total number of steps taken is

(11) $$k_o = \frac{\ell n(\frac{1}{\varepsilon})}{h} \quad .$$

Thereafter, the time-step taken is

$$\Delta t_k = \frac{h\ t_k}{\ell n(\frac{1}{\varepsilon})} = \frac{t_k}{k_o} \quad .$$

Hence

$$t_{k+1} = (1 + \frac{1}{k_o})t_k$$

leading to

(12) $$t_k = \left(1 + \frac{1}{k_o}\right)^{k-k_o} t_{k_o} \quad .$$

We regard the computation as finished when $\exp(-t_K\lambda_1) < \varepsilon$, i.e.

$$t_K = \frac{\ell n(\frac{1}{\varepsilon})}{\lambda_1} \quad .$$

Comparison with (12) yields

(13) $$K-k_o = \frac{\ell n(\lambda_n/\lambda_1)}{\ell n(1 + \frac{1}{k_o})}$$

since $$t_{k_o} = \frac{\ell n(\frac{1}{\varepsilon})}{\lambda_n} \quad .$$

If k_o in (11) or K in (13) does not evaluate to an integer, we take, of course, the next highest integer.

In order to add (11) to (13), we use the approximation (reasonable for large k_o)

$$\ell n(1 + \frac{1}{k_o}) \doteq \frac{1}{k_o}$$

arriving at the estimate for the total number of steps

$$K \doteq k_o\{1 + \ell n(\lambda_n/\lambda_1)\}$$

We thus have a quantitative expression for the qualitative fact that more steps are needed when the stiffness ration λ_n/λ_1 of the problem is large.

Example

Most of the ordinary approximations have local error estimates of the form ch^{p+1}, where p is an integer. Thus the global error in such an approximation is estimated by

(14) $$K\varepsilon \doteq \{1 + \ell n(\lambda_n/\lambda_1)\}\ \ell n(\frac{1}{\varepsilon})ch^p.$$

This reflects the well-known fact that a given method globally loses one order of accuracy, and also shows that for the error to go to zero with h, p must be at least 1. This is where the consistency requirement shows up.

The presence of the factor $\ell n(\frac{1}{\varepsilon})$ seems to conflict with the known fact that the global error in a p-th order method is $O(h^p)$. However, we are calculating not up to a constant t_K, but up to the point where $\exp(-t_K\lambda_1) < \varepsilon$, so that t_K becomes larger as ε becomes smaller. This is essentially the reason why this factor occurs.

4. CONCRETE REALIZATION AND NUMERICAL EXAMPLES

The preceding discussion applies to any acceptable approximation r(t). The choice of r(t) is however severely limited in practice, since the formation of higher powers of A and B gradually fills up the matrix appearing in the linear system. We therefore confine ourselves to the consistent (1,1) approximation, acceptable when $\frac{1}{2} \leqslant \theta \leqslant 1$,

$$r(t) = \frac{1-(1-\theta)t}{1+\theta t} .$$

For $\theta = \frac{1}{2}$ we have the estimate $|e^{-t} - r(t)| \leqslant \frac{h^3}{12}$ for $t \in [0,h]$, leading to

$$h \doteq (12\varepsilon)^{1/3}.$$

This choice of θ gives rise to a non-negative strictly monotonically increasing error, of the same sign throughout [0,h], and greatest at h. Loosely, accuracy is "wasted" since the approximation is equally good over [-h,0] where no eigenvalues can occur. We see from Fig. 1 how θ can be chosen to be optimal for a given ε in the sense of giving the longest interval in which (7b) holds. The numerical examples have been calculated both for $\theta = \frac{1}{2}$ and for this optimal value of θ.

The test problem used was the following:

$$u_t = u_{xx}; \quad u(0,t) = u(1,t) = 0$$
$$u(x,0) = \tfrac{1}{2} - |x - \tfrac{1}{2}| .$$

The space discretization used was the fourth-order difference formula

$$U'_m + \frac{1}{12}(U'_{m-1} - 2U'_m + U'_{m+1}) = \frac{1}{h^2}(U_{m-1} - 2U_m + U_{m+1}) .$$

An indication of the accuracy of this formula can be gained by comparing the eigenvalues of the semi-discrete form above with those of the continuous version. (See Table 1).

The error norm used was

$$\text{error} = \frac{\|U-\tilde{U}\|}{\|U\|} ,$$

where $\tilde{U}$ is the computed solution and $\| \ \|$ is the ordinary vector norm. In Tables 2 and 3 we give the total error as well as the number of steps taken for two choices of θ, namely $\theta = \frac{1}{2}$ and θ = 0,51674. The latter is optimal for ε = 0,0001, as we have shown elsewhere [Laurie (1975)].

Semi-discrete	Continuous
- 9,8692	- 9,8696
- 39,461	- 39,478
- 88,621	- 88,826
-156,74	-157,91
-242,18	-246,74

Table 1: Comparison of the five smallest eigenvalues of the semi-discrete form for h = 1/11 with those of the continuous version.

t	Number of steps	Relative error
0,01	46	0,00007
0,02	92	0,00014
0,05	172	0,00024
0,10	233	0,00035
0,20	294	0,00046
0,50	374	0,00097

Table 2: Numerical results for test problem with $\theta = \frac{1}{2}$.

t	Number of steps	Relative error
0,01	22	0,00003
0,02	44	0,00004
0,05	82	0,00005
0,10	111	0,00005
0,20	140	0,00047
0,50	178	0,00271

Table 3: Numerical results for test problem with $\theta = 0{,}51674$ (optimal for $\varepsilon = 0{,}0001$)

5. CONCLUDING COMMENTS

We have derived a step-size control strategy applicable to any space discretization coupled with any rational approximation in time. The estimate (14) for the global error contains one factor

$$c\ h^p\ \ell n(\tfrac{1}{\varepsilon})$$

related to the characteristics of the rational approximation, and one factor

$$1 + \ell n(\lambda_n/\lambda_1)$$

related to the stiffness of the space discretization.

In the numerical results quoted, the error per step has been much smaller than predicted. We attribute this phenomenon to the fact that the initial values contain a dominant component of the eigenfunction corresponding to λ_1, and it is intended to take into account the initial values in future refinements of the present technique.

REFERENCES

[1] CODY, W.J., MEINARDUS, G. & VARGA, R.S. Chebyshev rational approximations to exp(-x) on $[0,\infty)$ and applications to heat-conduction problems. J. Approx. Theory, vol. 2, 1969, pp. 50-65.

[2] GEAR, C.W. Numerical initial value problems in ordinary differential equations. Prentice-Hall, Englewood Cliffs, 1971.

[3] LAMBERT, J.D. Computational methods in ordinary differential equations. Wiley, New York, 1973.

[4] MURPHY, W.D. Cubic spline Galerkin approximations to parabolic systems with coupled non-linear boundary conditions. Int. J. Numer. Meth. Eng., vol. 9, 1975, pp. 63-71.

[5] RICE, J.B. A metalgorithm for adaptive quadrature. J. Assoc. Comp. Mach., vol. 22, 1975, pp. 61-82.

[6] RICHTMYER, R.D. & MORTON, K.W. Difference methods for initial-value problems. Interscience, New York, 1967.

[7] VARGA, R.S. Matrix iterative analysis. Prentice-Hall, Englewood Cliffs, 1962.

[8] LAURIE, D.P. A note on the θ-method for parabolic equations. CSIR Special Report WISK 165, Pretoria, 1975.

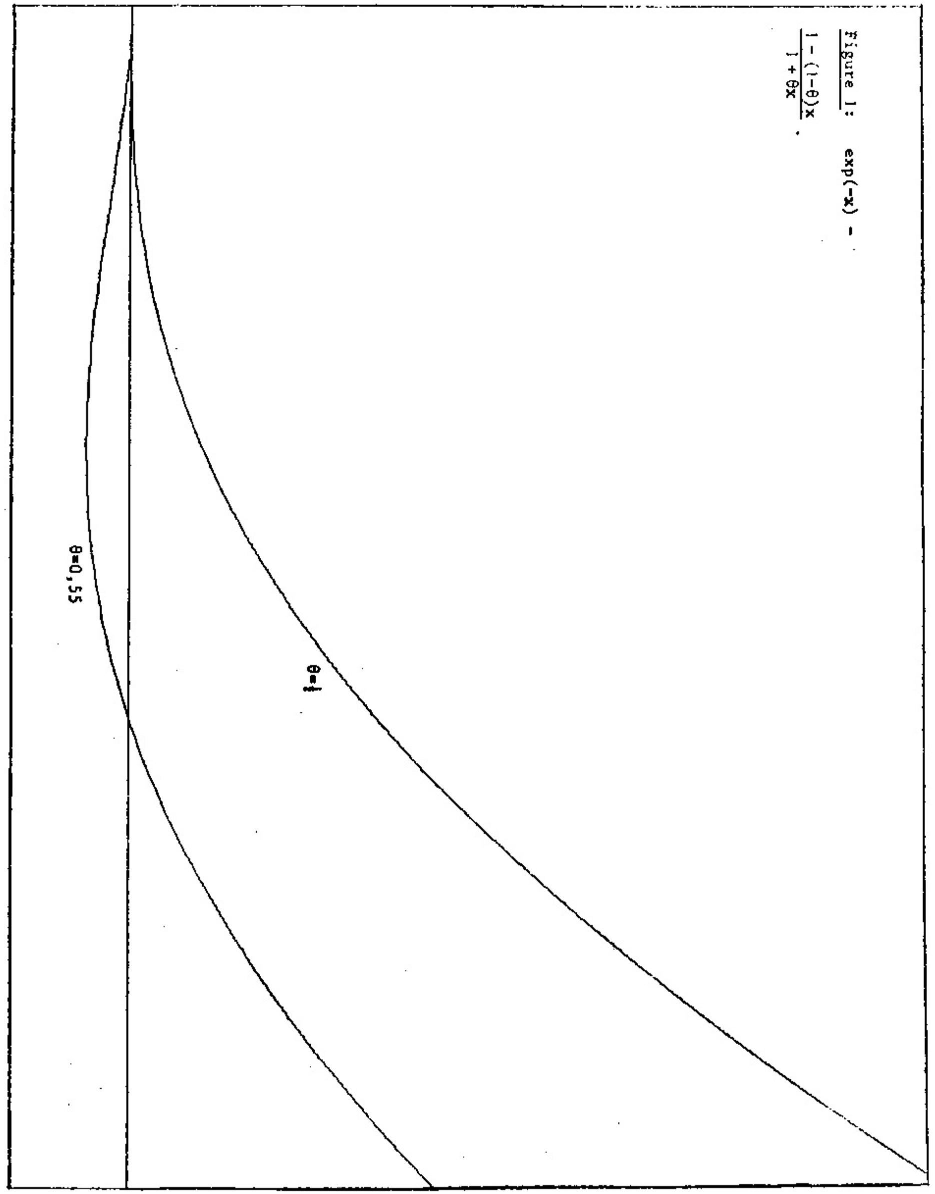

Figure 1: $\exp(-x) - \frac{1-(1-\theta)x}{1+\theta x}$.

NUMERICAL PROCEDURES FOR SOLVING A FOURTH-ORDER PARABOLIC PARTIAL DIFFERENTIAL EQUATION

G. R. Joubert
Department of Computer Science
University of Natal
Durban

1. Introduction

In [13] it has been shown that in the case of the numerical solution of a second-order parabolic partial differential equation improved explicit difference approximations can be constructed by use of a smoothing technique. In this paper the aim is to investigate the possibility of extending this method to the numerical solution of a higher-order parabolic partial differential equation which can be written as a system of simultaneous equations.

An example of such an equation, one for which a large number of different difference analogs have been proposed, is the fourth-order equation

(1.1) $u_{tt} + u_{xxxx} = 0,\ 0 \leqslant x \leqslant 1,\ t > 0.$

It will be assumed that initial values

(1.2) $$\begin{aligned} &(x,0) = g_0(x) \\ &u_t\ (x,0) = g_1(x) \end{aligned}$$

for $0 \leqslant x \leqslant 1$, and boundary values

(1.3) $$\begin{aligned} u(0,t) &= f_0(t) \\ u(1,t) &= f_1(t) \\ u_{xx}(0,t) &= p_0(t) \\ u_{xx}(1,t) &= p_1(t) \end{aligned}$$

for $t > 0$, are given.

The transformation

(1.4) $$\begin{aligned} \phi &= u_t \\ \psi &= u_{xx} \end{aligned}$$

transforms (1.1) into the system of simultaneous equations

(1.5) $$\begin{aligned} \phi_t &= -\psi_{xx} \\ \psi_t &= \ \ \phi_{xx} \end{aligned}$$

$0 \leqslant x \leqslant 1,\ t > 0$, which can be written as

(1.6) $\omega_t = A\omega_{xx}$

with

(1.7) $$\omega = \begin{bmatrix} \phi \\ \psi \end{bmatrix},\ A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

2. Difference approximations of the initial boundary value problem.

In order to construct finite difference approximations to (1.6) a rectangular difference grid with mesh widths Δx and Δt in the x and t directions respectively will be used, with $\Delta x,\ \Delta t > 0$. The co-ordinates of the grid points are given by $(j\Delta x,\ k\Delta t)$, and j and k non-negative integers and $M\Delta x = 1$.

Furthermore, with $x = j\Delta x,\ t = k\Delta t$,

$$\begin{aligned} \phi(x,t) &= \phi(j\Delta x,\ k\Delta t) \simeq \Phi_{j,k} \\ \psi(x,t) &\simeq \Psi_{j,k}, \\ \omega(x,t) &\simeq \Omega_{j,k}. \end{aligned}$$

The mesh ratio is given by $r = \dfrac{\Delta t}{(\Delta x)^2}$.

A large number of difference approximations of the initial boundary value problem given in §1 have been published. These either take the form of direct analogs of (1.1), see e.g. [1, 2, 3, 4, 5, 6, 7, 15], or approximations of (1.6).

In the latter case approximations of the scalar equation

(2.1) $v_t = v_{xx}$

are adapted to approximate (1.6). Examples of such methods were published by:

(a) Evans [11].

In this the Du Fort-Frankel difference analog [10] of (2.1) is used to approximate (1.6), the resulting approximation can be written as:

(2.2) $\Omega_{j,k+1} - \Omega_{j,k-1} = 2rA[\Omega_{j-1,k} + \Omega_{j+1,k} - \Omega_{j,k+1} - \Omega_{j,k-1}].$

The truncation error is similar to that of the Du Fort-Frankel equation, viz.

(2.3) $E = O((\Delta t)^2 + (\Delta x)^4 + (\frac{\Delta t}{\Delta x})^2).$

This approximation is explicit and stable for all $r > 0$, but starting values on two consecutive time layers are required.

(b) Fairweather and Gourlay [12].

Lees' method [14] for (2.1) applied to (1.6) gives:

(2.4) $\Omega_{j,k+1} - \Omega_{j,k} = rA[\Omega_{j-1,k+1} + \Omega_{j+1,k} - \Omega_{j,k+1} - \Omega_{j,k}].$

In this case the truncation error is

(2.5) $E = O(\Delta t + (\Delta x)^2 + \frac{\Delta t}{\Delta x}),$

the equation is also explicit and stable for all $r > 0$.

(c) Richtmyer [16].

The Crank-Nicolson method [8] for (2.1) applied to (1.6) gives:

$$(2.6)\qquad (I + \frac{r}{2} A\delta_x^2)\,\Omega_{j,k+1} = (I + \frac{r}{2} A\delta_x^2)\,\Omega_{j,k},$$

δ_x^2 being the central difference operator in the x direction. The truncation error is

$$(2.7)\qquad E = O((\Delta t)^2 + (\Delta x)^2).$$

(2.6) is implicit and stable for all $r > 0$.

(d) Fairweather and Gourlay [12].

The high order correct difference method of Douglas [9] applied to (1.6) gives:

$$(2.8)\qquad \begin{aligned}&(10I + 12rA)\,\Omega_{j,k+1} + (I-6rA)\left[\Omega_{j+1,k} + \Omega_{j-1,k+1}\right]\\ &= (10I-12rA)\,\Omega_{j,k} + (I+6rA)\left[\Omega_{j-1,k} + \Omega_{j+1,k}\right].\end{aligned}$$

The truncation error is

$$(2.9)\qquad E = O((\Delta t)^2 + (\Delta x)^4),$$

the approximation is implicit and stable for all $r > 0$.

3. A smoothing method.

In [13] a number of explicit difference methods employing a smoothing technique were developed. The simplest of these, when adapted to approximate (1.6), gives the difference analog:

$$(3.1)\qquad \Omega_{j,k+1} = \Omega_{j,k} + A \sum_{p=-2}^{2} b_p\, \Omega_{j+p,k}$$

or

$$(3.2)\qquad \begin{aligned}\Phi_{j,k+1} &= \Phi_{j,k} - \sum_{p=-2}^{2} b_p\, \Psi_{j+p,k}\\ \Psi_{j,k+1} &= \Psi_{j,k} + \sum_{p=-2}^{2} b_p\, \Phi_{j+p,k}\,.\end{aligned}$$

Under the assumption that

$$(3.3)\qquad b_{-i} = b_i,\ i=1,2$$

the consistency conditions are given by

$$(3.4)\qquad \begin{aligned}b_0 + 2b_1 + 2b_2 &= 0\\ b_1 + 4b_2 &= r.\end{aligned}$$

The truncation error is

$$(3.5)\qquad E = O(\Delta t + (\Delta x)^2).$$

If, in addition to (3.4), the relation

$$(3.6)\qquad b_2 = -\frac{r}{12}\,(6r-1)$$

holds, the truncation error is

$$(3.7)\qquad E = O((\Delta t)^2 + (\Delta x)^4).$$

Using the results from [13] it can easily be shown that the approximation (3.1) is stable for all $0 < r \leqslant 2$ if b_2 is chosen such that if

$$(3.8)\qquad \begin{aligned}&0 < r \leqslant 1 \text{ then } \frac{2r-1}{8} \leqslant b_2 \leqslant \frac{r}{4}\,,\\ &\text{else if}\\ &1 \leqslant r \leqslant 2 \text{ then } \frac{r^2}{8} \leqslant b_2 \leqslant \frac{r}{4}\,.\end{aligned}$$

The choice (3.6) for b_2 satisfying (3.8) is possible for all $0 < r \leqslant {}^2/3$.

The difference approximation (3.1) is not defined for all points of the difference grid used. It can however, see [13] , be written as a combination of a basic difference equation and a smoothing formula, both of which are defined for all grid points.

An obvious choice for the basic difference equation is the explicit approximation

$$(3.9)\qquad \Omega_{j,k+1} = \Omega_{j,k} + rA(\Omega_{j-1,k} - 2\Omega_{j,k} + \Omega_{j+1,k}),$$

which is stable for $0 < r \leqslant \frac{1}{2}$.

As has been shown previously a number of smoothing formulas can normally be constructed in each case. One possibility is the following:

$$(3.10)\qquad \begin{aligned}\Omega'_{j,k+1} &= a_{0,0}\,\Omega_{j,k} + a_{1,0}\,(\Omega_{j-1,k} + \Omega_{j+1,k})\\ &\quad + a_{0,1}\,\Omega_{j,k+1} + a_{1,1}\,(\Omega_{j-1,k+1} + \Omega_{j+1,k+1}).\end{aligned}$$

In (3.10) the value Ω' on the left-hand side is the smoothed value. If (3.9) is used to compute the values Ω on the right-hand side of (3.10), then the combination of (3.9) and (3.10) is equivalent to the computational procedure (3.1), if the following relations hold:

$$(3.11)\qquad \begin{aligned}a_{1,1} &= \frac{1}{r}\,b_2\\ a_{0,0} &= 2a_{1,1}\\ a_{1,0} &= -a_{1,1}\\ a_{0,1} &= 1-a_{0,0}.\end{aligned}$$

The computational procedure using (3.9) and (3.10) is the following: Using the prescribed initial and boundary values, values are computed for the first time layer by use of (3.9). These computed values, together with the known values on the previous time layer are then used in (3.10) to compute smoothed values Ω' for the first time layer. Using these smoothed values and the prescribed boundary values, values are then computed for the second time layer by use of (3.9). The smoothing procedure is then applied to these computed and the previously smoothed values to compute smoothed values Ω' for the second time layer. The whole process is repeated to compute smoothed values for subsequent time layers.

4. Example.

Consider the differential equation (1.1) with initial conditions

$$u(x,0) = \frac{x}{12}(2x^2 - x^3 - 1),$$

$$u_t(x,0) = 0, \quad 0 \leqslant x \leqslant 1$$

and boundary conditions

$$u(0,t) = u(1,t) = u_{xx}(0,t) = u_{xx}(1,t) = 0, \quad t \geqslant 0.$$

With the transformation (1.4) these conditions give

$$\omega(x,0) = \begin{bmatrix} 0 \\ x-x^2 \end{bmatrix}$$

$$\omega(0,t) = \omega(1,t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

In the following table results for this problem published in [12] are compared with results computed with the smoothing procedure described in §3. The differences between the analytic and approximate solutions are given for t = 0,02, computed with Δx = 0,05.

x =	0,05	0,20	0,35	0,50
Analytical soln.	-0,003 999 73	-0,015 048 33	-0,022 841 22	-0,025 659 28
(2.2) with $r = \frac{1}{2}$	0,000 004 89	0,000 014 16	0,000 011 86	-0,000 011 95
(2.4) with $r = \frac{1}{2}$	-0,000 016 75	-0,000 061 93	-0,000 060 64	-0,000 013 35
(2.6) with $r = \frac{1}{2}$	0,000 117 45	0,000 367 12	0,000 388 87	0,000 335 31
(2.8) with $r = \frac{1}{2}$	0,000 000 07	0,000 000 29	0,000 000 28	-0,000 000 17
(3.1) with $r = \frac{1}{2}$, b_2=0,083 333 33	0,000 000 58	0,000 022 80	0,000 048 29	0,000 009 17
(3.1) with $r = \frac{3}{2}$ b_2 = 0,3	0,000 035 85	-0,000 045 73	0,000 008 12	-0,000 039 45

REFERENCES

[1] ALBRECHT, J.: Zum Differenzenverfahren bei parabolischen Differentialgleichungen. Z. angew. Math. und Mech. 37, 202-212 (1957).

[2] COLLATZ, L.: Über das Differenzenverfahren bei Anfangswertprobleme partieller Differentialgleichungen. Z. angew. Math. und Mech. 16, 239 (1936).

[3] COLLATZ, L.: Zur Stabilität des Differenzenverfahrens bei der Stabschwingungsgleichung. Z. angew. Math. und Mech. 31, 392-393 (1951).

[4] CONTE, S.D.: A stable implicit finite-difference approximation to a fourth-order parabolic equation. J. Assoc. Comp. Machinery 4, 18-23 (1957).

[5] CONTE, S.D. & W.C. ROYSTER: Convergence of finite-difference solutions to a solution of the equation of the vibrating rod. Proc. Amer. Math. Soc. 7, 742-749 (1956).

[6] CRANDALL, S.H.: Numerical treatment of a fourth-order parabolic partial differential equation. J. Assoc. Comp. Machinery 1, 111-118 (1954).

[7] CRANDALL, S.H.: Optimum recurrence formulas for a fourth-order parabolic partial differential equation. J. Assoc. Comp. Machinery 4, 467-471 (1957).

[8] CRANK, J. & P. NICOLSON: A practical method for numerical evaluation of solutions of partial differential equations of heat conduction type. Proc. Cambridge Philos. Soc. 43,50-67 (1947).

[9] DOUGLAS, J.: The solution of the diffusion equation by a high-order correct difference equation. J. Math. Phys. 35, 145-151 (1956).

[10] DU FORT, E.C. & S.P. FRANKEL: Stability conditions in the numerical treatment of parabolic differential equations. Math. Tables and other Aids to Computation 7, 145-152 (1953).

[11] EVANS, D.J.: A stable explicit method for the finite-difference solution of a fourth-order parabolic partial differential equation. Comput. J. 8,280-287 (1965).

[12] FAIRWEATHER, G. & A.R. GOURLAY: Some stable difference approximations to a fourth-order parabolic partial differential equation. Maths. of Comp. 21, 1-11 (1967).

[13] JOUBERT, G.R.: Explicit difference approximations of the one-dimensional diffusion equation, using a smoothing technique. Numer. Math. 17, 409-430 (1971).

[14] LEES, M.: A priori estimates for the solutions of difference approximations to parabolic partial differential equations. Duke Math. J. 27, 297-312 (1960).

[15] NISHIMURA, T.: A numerical solution of one-dimensional heat conduction problems, thermal conductivity being function of temperature and situation. Mem. Fac. Engng. Nagoya Univ. 6, 30-33 (1954).

[16] RICHTMYER, R.D. & K.W. MORTON: Difference methods for initial-value problems, 2nd edition. New York: Interscience Publishers 1967.

A LANCZOS-TYPE ALGORITHM FOR THE EIGENVALUES OF AN ARBITRARY MATRIX

C. J. Wright
Department of Applied Mathematics
University of the Witwatersrand
Johannesburg

ABSTRACT

Lanczos in 1950 proposed a method of minimized iterations for reducing a matrix to tridiagonal form. However, because of cancellation errors which occur, modifications have to be made to the algorithm which make it extremely cumbersome to use when the given matrix is sparse. A computational variant is proposed which goes a long way toward solving these difficulties. An error analysis is also presented. Further the resulting tridiagonal matrix is automatically equilibrated. Also, only a submatrix of the full tridiagonal matrix need be utilized under some conditions if only some of the extreme eigenvalues are required.

NON-WELL POSED PROBLEMS IN ANALYSIS

Fritz John
Courant Institute of Mathematical Sciences
New York

ABSTRACT

For a problem that is well-posed in the sense of Hadamard we are presented with a space F of data f, a space U of solutions u, and a continuous mapping T : F → U. In an ill-posed problem T either ceases to be defined or to be continuous. For the class of problems discussed here we assume that T is defined but not continuous on F. Continuity, however, is essential for any numerical scheme that would permit to obtain approximations to the solution u from approximate data f. In many ill-posed problems continuity can be restored by restricting the domain of T to a suitable subset F_M of F. For that purpose we first restrict the solutions u to a subset U_M of the space U of admitted solutions; (usually U_M consists of those u ∈ U which together with a certain number of their derivatives have M as an upper bound for their absolute values). Subsequently F_M is defined as the set of f ∈ F for which Tf ∈ U_M. The set U_M must be chosen in such a way that the restriction of T to F_M is continuous.

A realistic numerical scheme can only exist when the problem is "well-behaved" in the sense that T is Hölder continuous on F_M. Examples of well-behaved improperly posed problems discussed here are the problem of analytic continuation of functions, the Cauchy problem for the Laplace equation, the final value problem for the heat equation, the numerical initial value problem for the wave equation in 3-space, and the problem of determining a function in the unit disk from integrals over chords.

(The lecture will be published in full in an appendix to the Proceedings of a Symposium on Differential Equations- 2-4 April 1975, presented by the National Research Institute for Mathematical Sciences.)

ON THE NUMERICAL SOLUTION OF SYSTEMS OF NON-LINEAR DIFFERENTIAL DIFFERENCE EQUATIONS

J. D. Neethling
Department of Mathematics
University of Stellenbosch

ABSTRACT

Written in integral form the type of equation to be considered is

$$(1)\quad 0 = Tu = F(u(t),\ u(t-\alpha_1),\ \ldots,\ u(t-\alpha_n),\ t) + \int_0^t G(u(s),\ u(s-\alpha_1),\ \ldots,\ u(s-\alpha_n),s)\ ds$$

with u, F, G ε R^m, $u = \phi(t)$ ε R^m, a given function, when $\beta \leq t \leq 0$. $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is a sequence of positive numbers and $\beta = -\max(\alpha_1, \alpha_2, \ldots, \alpha_n)$. F and G possess continuous second derivatives with respect to all vector components. An attempt to solve (1) using Newton's method in Banach space usually fails owing to the non-availability of the inverse Frêchet derivative of T. In this paper Newton's method is modified by substituting the Frêchet derivative by an operator which may be inverted to yield an explicitly applicable iteration formula. It is then shown that the iterations found by this formula do not differ from Newton iterations. The method is applied to ecological examples and it is shown how difficulties arising in the implementation may be avoided.

To appear in Zeitschrift für Angewandte Mathematik und Mechanik.

A METHOD FOR DETERMINING THE EIGENVALUES AND EIGENVECTORS OF A PERTURBED MATRIX

C. A. Botsaris
Department of Applied Mathematics
University of the Witwatersrand
Johannesburg

ABSTRACT

A method is presented for the determination, to a first order approximation, of the eigenvalues and eigenvectors of a perturbed matrix, from those of the unperturbed one. The method is based on the dyadic expansion of a matrix and it allows the eigenvalues of the unperturbed matrix to be of any multiplicity.

1. Introduction

Let A be an n X n matrix, whose eigenvalues λ_i and eigenvectors u_i, $i = 1,2,\ldots, n$, are known and let us form the matrix

$$C = A + \varepsilon B \tag{1.1}$$

where B is a given n X n matrix and ε is a perturbation parameter. We want to know the effect of the perturbation on the eigenvalues and eigenvectors of C, for small ε.

Jacobi in his paper on the algebraic eigenvalue problem (C.G.J. Crelle's Journal, 30 (1846) 51 - 95) treated the above perturbation problem for symmetric matrices with distinct eigenvalues. Since then, various authors have treated the same problem but, as Wilkinson [2] notes, no rigorous method based on the classical perturbation theory and handling the case of multiple eigenvalues, can be found in the literature.

However, Wilkinson, using the theory of algebraic functions, notes that, if λ is an eigenvalue of A of multiplicity m, then there will be a set of m eigenvalues which, in general, will fall into groups which can be expanded in terms of fractional power series of ε. In his book Wilkinson, using no more the classical perturbation theory but the Gershgorin's Theorems, develops a method for the perturbation of the eigenvalues and eigenvectors, which handles the case of multiple eigenvalues.

Our method uses the classical perturbation theory and the dyadic expansion of a matrix. In the case of an eigenvalue of multiplicity m an m X m matrix is determined, whose eigenvalue problem must be solved, in order to evaluate the first order perturbed eigenvalues and eigenvectors of (1.1).

We restrict ourselves to symmetric matrices and symmetric perturbations, i.e., all matrices in (1.1) are taken as symmetric.

2. The dyadic expansion of a matrix

Definition 2.1

Let x and y be two vectors in R^n. The outer product of x and y is the n X n matrix xy^T. The specialized matrices formed by the outer product of two vectors are called dyads.

Let u_i, $i = 1, 2,\ldots, n$, be a complete set of orthonormal vectors in R^n and let A be an n X n matrix. The normalized dyads E_{ij} are defined as

$$E_{ij} = u_i ui_j^T \tag{2.1}$$

and they have the following two properties:

(i) Idempotency

$$E_{ii}^P = E_{ii} \tag{2.2}$$

where P is a positive integer

Indeed,

$$E_{ii}^2 = (u_i u_i^T)\ (u_i u_i^T) = (u_i^T u_i)\ u_i u_i^T \tag{2.3}$$

and since

$$u_i^T u_j = \delta_{ij} \tag{2.4}$$

where δ_{ij} is the Kronecker delta, it follows that

$$E_{ii}^2 = E_{ii} \tag{2.5}$$

and, by induction, any power of E_{ii} is the same as E_{ii}.

(ii) nilpotency

$$E_{ij}\ E_{ij} = 0 \quad , i \neq j \tag{2.6}$$

Indeed,

$$E_{ij}\ E_{ij} = (u_i u_j^T)(u_i u_j^T) = (u_j^T u_i)\ u_i u_j^T = \delta_{ji}\ E_{ij} \tag{2.7}$$

and since $i \neq j$, it follows that

$$E_{ij}\ E_{ij} = 0 \tag{2.8}$$

Now since the set u_i, $i = 1, 2,\ldots, n$, is complete, it is evident that the set E_{ij} is complete and can therefore be used as a basis for the algebra of n X n matrices. Therefore A can be expanded in terms of these dyads as

$$A = \sum_{i,j} c_{ij}\ E_{ij} \tag{2.9}$$

where c_{ij}, $i, j = 1, 2,\ldots, n$, are coefficients to be determined. Premultiplication of (2.9) by u_m^T and postmultiplication by u_n yields

$$u_m^T\ Au_n = \sum_{i,j} c_{ij}\ u_m^T\ u_i u_j^T u_n \tag{2.10}$$

We see, the, that the terms on the right vanish unless $i = m$ and $j = n$, Therefore,

$$c_{mn} = u_m^T\ Au_n \tag{2.11}$$

Hence,

$$A = \sum_{i,j} (u_i^T\ Au_j)\ u_i u_j^T \tag{2.12}$$

3. The method

Let A be an n X n symmetric matrix. Let the eigenvalues of A fall into m isolated groups and let n_i be the number of equal eigenvalues within the ith group, i = 1, 2,..., m. Obviously,

$$n_1 + n_2 + \dots + n_m = n \tag{3.1}$$

Let us denote by

$$\overbrace{\lambda_1 = \lambda_2 = \dots = \lambda_{n_1}}^{n_1}, \ \overbrace{\lambda n_1 + 1 = \dots \lambda_{n_1 + n_2}}^{n_2}, \dots,$$

$$\overbrace{\lambda_{n_1 + n_2 + \dots + n_{m-1} + 1} = \dots = \lambda_{n_1 + n_2 + \dots + n_m}}^{n_m}$$

The eigenvalues of A.

Since A is symmetric it has a complete set of orthonormal eigenvectors $u_1, u_2, \dots, u_{n_1+1}, \dots, u_{n_1+n_2},$
$\dots, u_{n_1+n_2+\dots+n_{m-1}+1}, \dots, u_{n_1+\dots+n_m},$
where

$$\overbrace{u_{n_1 + n_2 + \dots + n_{i-1} + 1}, \dots u_{n_1 + n_2 + \dots + n_{i-1} + n_i}}^{n_i}$$

are the n_i eigenvectors corresponding to the n_i equal eigenvalues of the ith group,

$$\overbrace{\lambda_{n_1 + n_2 + \dots + n_{i-1} + 1}, \dots, \lambda_{n_1 + n_2 + \dots + n_{i-1} + n_i}}^{n_i} .$$

Let,

$$C = A + \varepsilon B \tag{3.2}$$

where B is a symmetric n X n matrix and ε is a perturbation parameter. We want to know the first-order effect on the pth eigenvalue and eigenvector, p = 1, 2,..., n.

Let us suppose that λ_p is equal to the eigenvalues of the kth group, k = 1, 2,..., m, i.e. of multiplicity n_k.
Then the eigenvectors corresponding to λ_p are u_i, i = $n_1 + n_2 + \dots + n_{k-1} + 1, \dots, n_1 + n_2 + \dots n_{k-1} + n_k$.

We want to find u_p' and λ_p' such that

$$Cu_p' = \lambda_p' u_p' \tag{3.3}$$

to a first order approximation at least.
Since every linear combination of the u_i's, i = $n_1 + n_2 + \dots + n_{k-1} + 1, \dots, n_1 + n_2 + \dots + n_{k-1} + n_k$, is an eigenvector of A with the same eigenvalue λ_p, it is not unreasonable to set

$$u_p' = \sum_{i = n_1 + \dots + n_{k-1} + 1}^{n_1 + \dots + n_{k-1} + n_k} \alpha_{pi} u_i + \varepsilon v_p \tag{3.4}$$

We also set

$$\lambda_p' = \lambda_p + \varepsilon\mu_p \tag{3.5}$$

We now seek to determine v_p, μ_p and the α_{pi}'s,
i = $n_1 + \dots + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k$.
Since the set u_i, i = 1, 2,..., n, is complete we can expand v_p as

$$\lambda_p = \sum_{i=1}^{n} b_{pi} u_i \tag{3.6}$$

If we expand A and B in terms of the $E_{ij} = u_i u_j^T$ dyads we get

$$A = \sum_{i,j=1}^{n} (u_i^T Au_j) u_i u_j^T = \sum_{i,j=1}^{n} \lambda_j (u_i^T u_j) u_i u_j^T = \sum_{i=1}^{n} \lambda_i u_i u_i^T \tag{3.7}$$

$$B = \sum_{i,j=1}^{n} (u_i^T Au_j) u_i u_j^T \tag{3.8}$$

Substituting (3.4) and (3.5) into (3.3) and using (3.2) we get

$$(A + \varepsilon B) \left(\sum_{i = n_1 + \dots + n_{k-1} + 1}^{n_1 + \dots + n_{k-1} + n_k} \alpha_{pi} u_i + \varepsilon v_p \right) = (\lambda_p + \varepsilon\mu_p) \left(\sum_{i = n_1 + \dots + n_{k-1} + 1}^{n_1 + \dots + n_{k-1} + n_k} \alpha_{pi} u_i + \varepsilon v_p \right) \tag{3.9}$$

or $$\sum_{i = n_1 + \dots + n_{k-1} + 1}^{n_1 + \dots + n_k} \alpha_{pi} Au_i + \varepsilon Av_p + \varepsilon \sum_{i = n_1 + \dots + n_{k-1} + 1}^{n_1 + \dots + n_{k-1} + n_k} \alpha_{pi} Bu_i + \varepsilon^2 Bv_p =$$

$$\sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \lambda_p \alpha_{pi} u_i + \varepsilon\lambda_p v_p + \varepsilon \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \mu_p u_i + \varepsilon^2 \mu_p v_p \tag{3.10}$$

But, $Au_i = \lambda_p u_i$, $i = n_1 + \dots + n_{k-1} + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k$. (3.11)

Therefore, in (3.10), the first term of the left is equal to the first term on the right. Hence,

$$\varepsilon A v_p + \varepsilon \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} Bu_i + \varepsilon^2 Bv_p = \varepsilon\lambda_p v_p + \varepsilon \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \mu_p u_i + \varepsilon^2 \mu_p v_p \tag{3.12}$$

Collecting from (3.12) the terms in the first order in ε we get

$$Av_p + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} Bu_i = \lambda_p v_p + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi}\mu_p u_i \tag{3.13}$$

If we now use the expansions for v_p, A and B from (3.6), (3.7) and (3.8) respectively we have,

$$\left(\sum_{i=1}^{n} \lambda_i u_i u_i^T\right)\left(\sum_{j=1}^{n} b_{pj} u_j\right) + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \text{ multiplied by}$$

$$\left(\sum_{m,j=1}^{n} (u_m^T Bu_j)\, u_m u_j^T\right) u_i = \lambda_p \left(\sum_{i=1}^{n} b_{pi} u_i\right) + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \mu_p u_i \tag{3.14}$$

Using the orthogonality relation $u_i^T u_j = \delta_{ij}$ we get from (3.14),

$$\sum_{i=1}^{n} {}_i\, b_{pi} u_i + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} {}_{pi} \sum_{m=1}^{n} (u_m^T Bu_i)\, u_m =$$

$$\lambda_p \sum_{i=1}^{n} b_{pi} u_i + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \mu_p u_i \tag{3.15}$$

Recognizing that

$$\sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}} \alpha_{pi} \sum_{m=1}^{n} (u_m^T Bu_i) u_m = \sum_{i=1}^{n} \sum_{j=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pj} (u_i^T Bu_j)\, u_i \tag{3.16}$$

we have from (3.15)

$$\sum_{i=1}^{n} \lambda_i b_{pi} u_i + \sum_{i=1}^{n} \sum_{j=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pj} (u_i^T Bu_j)\, u_i = \lambda_p \Sigma b_{pi} u_i + \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pi} \mu_p ui \tag{3.17}$$

Since the u_i's, i=1, 2, .., n, are linearly independent the coefficient of each vector in (3.17) must separately vanish.

Hence,

$$\lambda_i b_{pi} + \sum_{j=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pj} (u_i^T Bu_j) = \lambda_p b_{pi} \tag{3.18}$$

for $i \in \{ 1, 2, \dots, n_1 + n_2 + \dots + n_{k-1}, n_1 + n_2 + \dots + n_{k-1} + n_k + 1, \dots, n\}$

and

$$\sum_{j=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pj} (u_i^T Bu_j) = \alpha_{pi} \mu_p \tag{3.19}$$

for $i \in \{ n_1 + \dots + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k\}$.
Equation (3.19) is now written, in a matrix form, as follows:

$$Q_p \alpha_p = \alpha_p \mu_p \tag{3.20}$$

where Q_p is an n_k X n_k matrix, whose (i,j) element is $u_i^T Bu_j$, for all $i,j \in \{n_1 + \dots + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k\}$ and α_p the n_k X 1 vector

$$\alpha_p = (\alpha_{p,n_1 + \dots + n_{k-1} + 1}, \dots, \alpha_{p,n_1 + \dots + n_{k-1} + n_k})^T \tag{3.21}$$

Note that Q_p is symmetric due to the symmetry of B. We can easily recognize (3.20) as the eigenvector equation for the matrix Q_p. Therefore μ_p is an eigenvalue and α_p the corresponding eigenvector of Q_p. Note that Q_p is symmetric and hence has a complete set of eigenvectors.

From (3.18) we can now evaluate the coefficents b_{pi}.
We find,

$$b_{pi} = \frac{\sum\limits_{j=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_{k-1}+n_k} \alpha_{pj}\,(u_i^T Bu_j)}{\lambda_p - \lambda_i} \tag{3.22}$$

for all $i \in \{1, 2, \dots, n_1 + \dots + n_{k-1}, n_1 + \dots + n_{k-1} + n_k + 1, \dots, n\}$

We observe that the coefficients of $u_{n_1 + \dots + n_{k-1} + 1}, \dots, u_{n_1 + \dots + n_{k-1} + n_k}$ in the expansion of v_p (3.6) are indeterminate. This is to be expected however. Obviously $\varepsilon\, b_{pi}\, u_i$ perturbs the magnitute of $\alpha_{pi}\, u_i$, $i=n_1 + \dots + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k$, without changing its direction. Since we can change the length of a vector without disturbing the fact that it is an eigenvector of A, we should not expect these coefficients to be determined. The choice of the b_{pi}'s, $i=n_1 + \dots + n_{k-1} + 1, \dots, n_1 + \dots + n_{k-1} + n_k$, affects the second order terms, but may be chosen arbitrarily as far as the first order terms are concerned.

We now summarize the previous analysis for an eigenvalue λ of multiplicity m :

(i) find the m eigenvectors with eigenvalue λ.
Let us denote, for simplicity, these eigenvectors

$$u_1, u_2, \dots, u_m$$

(ii) form the matrix $Q = (u_i^T Bu_j)$, $i, j = 1,2,\dots, m$, and find its eigenvalues $\mu_1, \mu_2, \dots, \mu_m$ and associated normalized eigenvectors $\alpha_1, \alpha_2, \dots, \alpha_m$ where $\alpha_k = (\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{km})^T$, $k=1, 2, \dots, m$ (3.23)

(iii) set,

$$\lambda_k' = \lambda + \varepsilon\mu_k$$

$$u_k' = \sum_{i=1}^{m}\alpha_{ki}\, u_i + \varepsilon \sum_{i=1}^{m} b_i u_i + \varepsilon \sum_{i=m+1}^{n} \frac{\sum\limits_{j=1}^{m}\alpha_{kj}(u_i^T Bu_j)}{\lambda-\lambda_i}\, u_i \tag{3.25}$$

K = 1, 2,..., m. The coefficients b_i, i=1, 2,..., m, are arbitrary.

Remarks

(i) If the eigenvalues of Q are all distinct, then the perturbation has split the degeneracy for the particular eigenvalue. Indeed, if $\mu_k \neq \mu_p$ for $k,p \in \{1, 2, \dots, m\}$, $k \neq p$, then $\lambda_k' \neq \lambda_p'$

(ii) Since the coefficients b_i, i=1, 2,..., m, are indeterminate we can choose them to be zero. Equation (3.25) is then written as

$$u_k' = \sum_{i=1}^{m}\alpha_{ki} u_i + \varepsilon \sum_{i=m+1}^{n} \frac{\sum\limits_{j=1}^{m}\alpha_{kj}(u_i^T Bu_j)}{\lambda-\lambda_i}\, u_i, \tag{3.26}$$

k = 1, 2,..., m.

(iii) If we set

$$D = \varepsilon B \tag{3.27}$$

then (3.2) is written as

$$C = A + D \tag{3.28}$$

Since μ_k is an eigenvalue of $Q = (u_i^T Bu_j)$ with associated eigenvector α_k, $\varepsilon\mu_k$ is an eigenvalue of $\varepsilon Q = \varepsilon(u_i^T Bu_j) = (u_i^T \varepsilon Bu_j) = (u_i^T Du_j)$ with the same associated eigenvector. i.e. for the perturbation problem (3.28),

a) form the matrix $Q = (u_i^T Du_j)$

b) find its eigenvalues μ_k and associated eigenvectors α_k, k = 1, 2,.., m, where

$$\alpha_k = (\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{km})^T \tag{3.29}$$

c) set

$$\lambda_k' = \lambda + \mu_k' \tag{3.30}$$

$$u_k' = \sum_{i=1}^{m}\alpha_{ki} u_i + \sum_{j=m+1}^{n} \frac{\sum\limits_{j=1}^{m}\alpha_{kj}\,(u_i^T Du_j)}{\lambda-\lambda_i}\, u_i \tag{3.31}$$

k = 1, 2,..., m.

(iv) If λ_p is a simple eigenvalue with associated eigenvector u_p then Q is the 1 × 1 matrix $u_p^T Qu_p$ i.e. a scalar. Setting

$$\mu_p = u_p^T Qu_p \tag{3.32}$$

equations (3.24) and (3.25) are now written as

$$\lambda_p' = \lambda_p + \varepsilon\mu_p \tag{3.33}$$

$$u_p' = \alpha_p u_p + \varepsilon b_p u_p + \varepsilon \sum_{\substack{i=1 \\ i \neq p}}^{n} \frac{\alpha_p (u_i^T \, B u_p)}{\lambda_p - \lambda_i} \, u_i \tag{3.34}$$

We can easily recognize (3.33) and (3.34) as the well known relations for the first order perturbation of the eigenvalues and eigenvectors in the case of distinct eigenvalues.

4. Conclusion

We have presented a simple, but rigorous, method which permits us to determine, to a first order approximation, the eigenvalues and eigenvectors of the matrix

$$C = A + \varepsilon B \tag{4.1}$$

from those of the matrix A, where the matrices A and B are both symmetric. In the case of an eigenvalue of multiplicity m, the eigenvalue problem of an m × m matrix has to be solved. We therefore expect the method to be more efficient for m<<n.

Acknowledgements

I would like to thank warmly Prof. D.H. Jacobson for his comments.

References

1. M.C. PEASE, III "Methods of Matrix Algebra"
   Academic Press, New York and London, 1965.

2. J.H. WILKINSON "The algebraic eigenvalue Problem"
   Oxford University Press, 1965.