NUMERICAL SOLUTIONS OF A PARTIAL DIFFERENTIAL EQUATION

ADAM H. WALKER

SUMMARY OF A THESIS SUBMITTED FOR THE M.Sc. IN COMPUTING

BY ADAM H. WALKER

NUMERICAL SOLUTIONS OF A PARTIAL DIFFERENTIAL EQUATION

The subject of the thesis is the numerical integration of the Partial Differential Equation :

$$\frac{\partial f}{\partial t} = f^{p} \frac{\partial^{2} f}{\partial x^{2}} \quad \text{for} \quad 0 \leq x \leq 1.$$

In particular, the case $p = 1$ is dealt with under the conditions :

$$f(0,t) = c_1 \; ; \; f(x,0) = c_2 \quad \text{for } 0 < x \leq 1 \; ; \left.\frac{\partial f}{\partial x}\right]_{x=1} = 0$$

Also, several other cases are treated in order to bring out various aspects of the techniques used. The case $p = -1$ is treated in order to obtain comparisons with known analytical solutions. In the case $p = 0$ the techniques are entirely linear and one method of integration used demands the inversion of a tri-diagonal matrix. Other values of p are chosen in order to illustrate various aspects of the stability theory and demonstrate the generality of the methods.

In the first chapter, an analytic approach is adopted and the equati is classified by its non-linearity and by the fact that it is Parabolic. The equation is non-linear in the sense that it contains a product of the dependent variable and a partial derivative. These observations determine the approach to the numerical solution. In this chapter an attempt is made to find analytical solutions but it appears that such solutions are only useful for highly speciali initial and boundary conditions. Even when such solutions exist, they are usual implicit and very unwieldy. The exception to this is the case $p = -1$ which can be seen to have a simple analytical solution :

$$f(x,t) = \tfrac{1}{2} \tan \left(\pi/4 + \frac{x}{2} + t\right)$$

for suitable boundary and initial conditions.

In the second chapter, three numerical methods of integration are discussed in detail. These are :

1. A simple explicit method.
2. A semi-explicit method which requires a special starting procedure. The method used is known as the DuFort-Frankel method.
3. An implicit method based on the well-known Crank-Nicolson technique. This method reduces to the solution of sets of simultaneous non-linear equations

The stability of the three methods is dealt with empirically by comparison with the linear heat-conduction equation. The results obtained may be stated briefly as follows.

1. For the explicit method we must have :

$$f^p \frac{\delta t}{(\delta x)^2} < \tfrac{1}{2} \; ; \quad f^p > 0.$$

2. For the DuFort-Frankel method :

$$f^p > 0 \; ;$$

$$\delta t/(\delta x)^2 \text{ is unrestricted.}$$

3. For the Crank-Nicolson method :

$$f^p > 0 \; ,$$

providing that the parameter r used to combine the forward difference and backward difference representations of $\frac{\partial f}{\partial t}$ is greater than ½. δt and δx are the step-lengths in t, x respectively. The second method is found to exhibit the phenomenon of inconsistency unless $\delta t/ \delta x$ is kept small. Truncation errors and the treatment of the singularity are mentioned briefly in this chapter. The main portion of the chapter is devoted to the development of the Crank-Nicolson simultaneous equations and methods for the solution of these equations. Iterativ methods for the solution of the equations are treated in detail. Explicit analyt methods for the solution of the equations are ignored, since they are clumsy to programme. Two iterative processes are given. One is an extension of Newton-Raphson iteration and the other is a generalisation of direct functional iteratio Three separate methods are investigated for the generation of first approximations for the iterative processes. These are :

(a) The explict formula mentioned earlier.

(b) The Newton Backward Difference extrapolation formula.

(c) Use of $f(x,t - \delta t)$ as a first approximation to $f(x,t)$.

In the third chapter, the main results are presented and discussed. It is found that the empirical stability theory given in the second chapter gives agreement with the numerical results. The DuFort-Frankel method is seen to give inconsistency for values of $\delta t/(\delta x)^2$ higher than those allowed for stability in the simple explicit method. The numerical results obtained by the explicit and Crank-Nicolson methods are found to agree fairly closely. The Crank-Nicolson method gives good agreement with the analytical solution for $p = -1$; usually the agreement is much closer than 1 %. The Newton-Raphson iterative method is found tobbe much superior to the direct iteration process and it is found that extremel fast convergence may be obtained by making the parameter r just greater than ½. At the end of the chapter, a review of the investigation is given and some conclusions of a general nature are drawn.

The appendices contain a brief account of the programmes used and some of the numerical results obtained, as well as a short list of the books fou useful in the solution of the problem.

I wish to thank all the staff of the Computing Laboratory for their help with this thesis. I am particularly grateful to Dr. Gilles for his assistance and encouragement.

CONTENTS

NUMERICAL SOLUTIONS OF A PARTIAL DIFFERENTIAL EQUATION

Chapter 1

1.1. Introduction

The development of high speed digital computers during the last twenty years has made possible the solution of many difficult problems in numerical analysis; in particular, these problems which involve tedious processing of large amounts of data. The solution of Partial Differential Equations by numerical methods, usually leads to the treatment of large numbers of variables and therefore, this field was one of the first to undergo extensive investigation. To begin with, much research was done on the solution of linear Partial Differential Equations. This was due to two things. Firstly, the mathematical techniques required had been developed to a high degree in the years immediately prior to, and during the war. Two main topics, Stability and Matrix Iteration had been studied for some time. Secondly, the problems which arose were usually linear or could be suitably linearised. However, work has been done on non-linear equations, for example by Blanch and also by Crank and Nicolson. In general, attempts to solve non-linear equations have developed on a semi-empirical approach to the important question of stability. For example, the Partial Differential Equation :

$$\frac{\partial u}{\partial t} = \frac{\partial^2}{\partial x^2} (u^n)$$

may be written in the form :

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(nu^{n-1} \frac{\partial u}{\partial x} \right)$$

and this may be treated via its similarity to the heat-conduction equation with diffusion constant nu^{n-1}.

The semi-empirical approach yields results which are useful but not rigorous and it is used in the problem dealt with here.

1.2. Statement of the Problem

We wish to find numerical solutions to the Partial Differential Equation :

$$\frac{\partial f}{\partial t} = f^{p} \frac{\partial^{2} f}{\partial x^{2}}$$

with the conditions :

$$f(x,0) = c, \quad 0 < x \leq 1 \ ;$$

$$f(0,t) = k, \quad 0 \leq t < \infty, \text{ where } k \neq c;$$

$$\left.\frac{\partial f}{\partial x}\right|_{x=1} = 0, \quad 0 \leq t < \sigma \ .$$

In this investigation we shall take

$$k = 2,$$

$$c = 1,$$

$$p = 1.$$

except when it is necessary to check particular points in the theory. Hence the following values of the parameters will also be used :

$p = 1$		$p = 2$
$k = -2$	and	$k = -2$
$c = -1$		$c = -1$

These values of the parameters are chosen for convenience in checking by hand. The reason for the choice of negative values for k and c will emerge later. The case $p = -1$ is also treated for special initial and boundary conditions.

1.3. Analytical Discussion

To begin with, we observe that the equation is non-linear and does not contain mixed partial derivatives. Next, we examine the characteristics of the equation. The characteristics are given by the auxiliary equation :

$$f^p \lambda^2 = 0 \quad \text{where} \quad \lambda = \frac{dt}{dx}$$

that is $\lambda = 0$ (twice) for all x, t, p.

Hence the equation is of parabolic type for all values of x and t since it has only one real characteristic.

Since the equation is non-linear, it cannot be reduced to simpler form as is the case with certain linear and quasilinear equations.

The absence of mixed derivatives such as $\frac{\partial^2 f}{\partial x \partial t}$ introduces the possibility of explicit methods, for if $\frac{\partial f}{\partial t}$ is replaced by $\frac{(E_t-1)f_{xt}}{\delta t}$ and $\frac{\partial f}{\partial x}$ by $\frac{(E_x-1)f_{xt}}{\delta x}$ then $\frac{\partial^2 f}{\partial x \partial t}$ may be expressed as follows :

$$\frac{\partial^2 f}{\partial x \partial t} = (E_x-1)(E_t-1)f_{xt}/\delta t \delta x$$

$$= \left[f(x+1,t+1) - f(x+1,t) - f(x,t+1) + f(x,t)\right]/\delta t \delta$$

and this last expression involves simultaneous determination of two unknowns $f(x,t+1)$ and $f(x+1,t+1)$ in the solution of the equation at time t+1.

It is worth mentioning that analytical solutions can be given under special circumstances.

The equation is

$$\frac{\partial f}{\partial t} = f^p \frac{\partial^2 f}{\partial x^2}$$

We now look for solutions of the form

$$f(x,t) = X(x)T(t)$$

Then
$$XT' = X^pT^pX''T$$

where the dashes denote differentiation with respect to the independent variabl

Rearranging gives

$$\frac{T'}{T^{p+1}} = X''X^{p-1}$$

Now $\frac{T'}{T^{p+1}}$ is a function of t alone and $X''X^{p-1}$ is a function of x alone.

Hence, each of these functions equals some constant. That is

$$X^{p-1}\,X'' = \frac{T'}{T^{p+1}} = \alpha$$

where α is some arbitrary constant.

In the case p = 0 we have simply solutions of the form :

$$e^{\alpha t}\,(Ae^{\sqrt{\alpha}x} + Be^{-\sqrt{\alpha}x})$$

which may be fitted to suitable boundary conditions.

In the case p = 1 we have :

$$X'' = \alpha \; ; \qquad \frac{T'}{T^2} = \alpha ,$$

and these may be integrated directly to give :

$$X = \frac{\alpha x^2}{2} + Ax + B,$$

$$T = \frac{-1}{(\alpha t + c)}$$

where A, B, C are constants of integration.

Hence the general solution of this form is :

$$f(x,t) = - \frac{(\frac{\alpha x^2}{2} + Ax + B)}{(\alpha t+c)}$$

The condition $\left.\frac{\partial f}{\partial x}\right|_{x=1} = 0$ gives $A = -\alpha$.

Therefore

$$f(x,t) = - \frac{(\alpha x(x/2 -1) + B)}{(\alpha t+c)}$$

This analysis shows that if

$$f(x,0) = \frac{\alpha x^2}{2} - \alpha x + B$$

then there is a solution

$$f(x,t) = - \frac{(\alpha x(x/2 - 1) + B)}{(\alpha t-1)}$$

This solution will have a singularity at $t = \frac{1}{\alpha}$ unless $\alpha \leq 0$.

It is not possible to fit a solution of this type to the initial conditions given in the problem considered here.

It is of interest to look for solutions of the equation :

$$\frac{\partial f}{\partial t} = f^n \frac{\partial^2 f}{\partial x^2}$$

which are of the form $f(x,t) = g(x-vt)$ where v is a constant and g is some function to be determined. Solutions of this type are similar to travelling wave solutions of the linear wave equation. We now seek to determine the form of g. We have

$$\frac{\partial f}{\partial t} = -vg'(\xi)$$

where $\xi = x - vt$ and the dash denotes differentiation with respect to ξ .

Also

$$\frac{\partial^2 f}{\partial x^2} = f''(\xi)$$

Substituting in the partial differential equation gives :

$$-vg' = g^n g''$$

Now

$$g'' = \frac{d}{dg}(\tfrac{1}{2}p^2) \quad , \text{ where } p \equiv g'.$$

Hence,

$$-vp = g^n \overset{p}{p} \frac{dp}{dg} + v) = 0$$

that is

$$p(g^n \frac{dp}{dg} + v) = 0.$$

Therefore,

$$p = 0 \quad \text{or} \quad v + g^n \frac{dp}{dg} = 0.$$

But p = 0 just gives the simple solution : g = constant.

The other solution is :

$$v + g^n \frac{dp}{dg} = 0$$

or

$$-\frac{vdg}{g^n} = dp.$$

Now suppose $n \neq 1$.

Then

$$-\frac{vdg}{g^n} \equiv \frac{v}{(n-1)} d\left(\frac{1}{g^{n-1}}\right)$$

Therefore

$$\frac{v}{(n-1)} d\left(\frac{1}{g^{n-1}}\right) = dp$$

thus

$$\frac{v}{(n-1)} \frac{1}{n-1} = p + c_1$$

where c_1 is an arbitrary constant.

Substituting $p = \frac{dg}{d\xi}$ gives :

$$\frac{dg}{\left(\frac{v}{(n-1)} \frac{1}{g^{n-1}} - c_1\right)} = d\xi .$$

Therefore

$$\int \frac{dg}{\left(\frac{v}{(n-1)} \frac{1}{g^{n-1}} - c_1\right)} = \xi + c_2 .$$

where c_2 is another arbitrary constant.

Now

$$\int \frac{dg}{\left(\frac{v}{(n-1)} \frac{1}{g^{n-1}} - c_1\right)}$$

may be expressed in terms of elementary functions for special values of n. We now consider some of these special solutions.

(a) <u>The case n = 0</u>

We have

$$- vg' = g''$$

This leads to the solution

$$f(x,t) = k_1 e^{-v(x-vt)} + k_2 .$$

(b) <u>The case n = 1</u>

Here we obtain

$$v + g\frac{dp}{dg} = 0$$

that is
$$v \log g + p = c_1$$

where c_1 is an arbitrary constant.

Therefore

$$\int \frac{dg}{(c_1 - v \log g)} = \xi + c_2$$

where c_2 is an arbitrary constant.

Put $u = c_1 - v \log g$. $du = \frac{-v}{g}\, dg$. $g = e^{-(u-c_1)/v}$.

Therefore

$$\int - \frac{du e^{-(u-c_1)/v}}{vu} = \xi + c_2.$$

This latter integral may be reduced, by simple transformations, to

$$K \int \frac{e^{-q}}{q}\, dq$$

This integral is of the same form as the Exponential-Integral

$$e\, i(x) = \int_x^{\infty} \frac{e^{-u}}{u} \ ; \quad x > 0.$$

Integrals of this type often appear in non-linear problems. The analytical solution for n=1 cannot be taken further.

(c) <u>The case n=-1</u>

In this case we have

$$-\int \frac{dg}{(c_1 + \frac{v}{2} g^2)} = \xi + c_2$$

where c_1 and c_2 are arbitrary constants.

Therefore

$$-\int \frac{2}{v} \frac{dg}{\left(\frac{c_1 2}{v} + g^2\right)} = \xi + c_2$$

Integrating gives

$$-\frac{2}{v}\left[\sqrt{\frac{2c_1}{v}}\right]^{-1} \tan^{-1} x/\sqrt{\frac{2c_1}{v}} = \xi + c_2 \text{ (if } \frac{c_1}{v} > 0).$$

Without loss of generality we may put $K_1 = \sqrt{\frac{2c_1}{v}}$

then

$$-\frac{2K_1^{-1}}{v} \tan^{-1} x/K_1 = \xi + c_2.$$

Thus

$$\tan^{-1} g/K_1 = -\frac{v}{2K_1^{-1}} \xi + K_2 \quad \text{where} \quad K_2 = \frac{-c_2 v}{2K_1^{-1}}$$

Therefore $$g/K_1 = \tan(K_2 - \frac{v}{2K_1^{-1}} \xi)$$

This expression for g is simple and explicit.

(d) <u>The case n = 2</u>

The general expression reduces to:

$$\int \frac{dg}{\left(\frac{v}{g} - c_1\right)} = \xi + c_2$$

where c_1 and c_2 are arbitrary constants

$$\int \frac{dg}{\left(\frac{v}{g} - c_1\right)} = \int \frac{g\,dg}{(v - c_1 g)}$$

$$= \int -\frac{1}{c_1} \left(\frac{v - c_1 g}{v - c_1 g} \right. dg + \int \frac{v}{c_1} \cdot \frac{dg}{(v - c_1 g)}$$

$$= - \frac{g}{c_1} - \frac{v}{c_1^2} \log (v - c_1 g)$$

Hence

$$- \frac{g}{c_1} - \frac{v}{c_1^2} \log (v - c_1 g) = \xi + c_2$$

This formula gives g implicitly as a function of x and t. The usefulness of this special analytical approach is severely restricted, since we must have :

$$f(x,0) = g(x,0).$$

In effect, these solutions are only suitable for highly specialised initial values, and in general, it is impossible to fit boundary conditions which contain a derivative. However, it is of interest to note that the problem examined here can be related to another problem which has been studied in some detail. The equation :

$$\frac{\partial f}{\partial t} = f^n \frac{\partial^2 f}{\partial x^2}$$

may be written in the form :

$$\frac{1}{(-n+1)} \frac{\partial}{\partial t} (f^{-n+1}) = \frac{\partial^2 f}{\partial x^2} \quad , \text{ for } n \neq 1$$

Employing the change in the dependent variable defined by $f^{-n+1} = u$ gives :

$$\frac{+1}{(-n+1)} \frac{\partial u}{\partial t} = \frac{\partial^2}{\partial x^2} \left(u^{\frac{1}{(-n+1)}} \right)$$

Results are given by Richtmyer (see reference) for the non-linear equation :

$$\frac{\partial u}{\partial t} = \frac{\partial^2}{\partial x^2} (u^5)$$

using the method which has just been discussed (for suitably chosen initial values) and the theoretical answers are compared with those found by means of a linearised Crank-Nicolson method. The agreement is found to be good.

1.4. General Notes

We may observe that the equation under investigation bears a strong resemblanse to the linear diffusion equation :

$$\frac{1}{k}\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2} \quad \text{where k is a constant.}$$

The resemblance is exploited later.

In the numerical treatment, care has been taken to ensure that $f(x,0)$ is never zero. The reason for this is as follows.

The equation :

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{f}\frac{\partial f}{\partial t}$$

may be written in the form :

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial t}\ (\log f + g(x))$$

where $g(x)$ is an arbitrary function of x.

If $f(x,t)$ is zero for some x,t then $\log f$ takes the value $-\infty$; in other words a singularity occurs. In order to avoid such a possibility, K and c are chosen to be non-zero.

The numerical solution to the problem was found using the English Electric KDF 9 Digital Computer. The programming language used was Algol.

Chapter 2

2.1. Review of General Techniques

The choice of methods of solving the given equation is determined by the fact that the equation is parabolic and non-linear. The general position may besummarised briefly as follows.

Elliptic and hyperbolic partial differential equations are extreme types but can usually be solved by well-defined methods. Elliptic equations have no real characteristics and in general, such equations are solved by implicit methods. At the other extreme, hyperbolic equations have real distinc characteristics and can oftem be solved conveniently using one of three methods Parabolic equations cannot be solved using the method of characteristics but can usually be solved by explicit or implicit methods.

Many of the methods which have been used for linear equations may be extended to non-linear equations. An extensive account of explicit and implicit methods is given by Richtmyer in connection with the linear heat conduction equation and some of these methods may easily be adapted to non-linear problems. Here we shall use three methods, two explicit and one implicit. From time to time in the investigation, combinations of these methods will be used. As we have noted earlier, purely explicit methods are available, since the equation does not contain any mixed partial derivatives. In the first instance we shall simply give the representations used, and deal with the details later.

(a) Explicit Method

In this method we replace $\frac{\partial f}{\partial t}$ by the finite difference expression

$$\frac{(E_t - 1)f}{\delta t}$$

and $\frac{\partial^2 f}{\partial x^2}$ by

$$\frac{\delta_x^2 f}{(\delta x)^2}$$

where E_t and δ_x have the usual meanings and δt and δx are the step-lengths in t and x respectively. In these representations we have ignored higher differences. Hence the equation :

$$\frac{\partial f}{\partial t} = f^D \frac{\partial^2 f}{\partial x^2}$$

is replaced by

$$(E_t - 1)f = f^D \frac{(\delta t)}{(\delta x)^2} \quad \delta_x^2 x^2 f$$

and it is convenient to write $s = \delta t/(\delta x)^2$.

(b) Semi-Explicit Method

The method used here is due to Dufort and Frankel and is an adaption of Richardson's Formula. In this method, $\frac{\partial f}{\partial t}$ is represented by :

$$\frac{f(x,t+1) - f(x,t-1)}{2\delta t}$$

and $\frac{\partial^2 f}{\partial x^2}$ is represented by

$$\frac{f_{x+1,t} - f_{x,t+1} - f_{x,t-1} + f_{x-1,t}}{(\delta x)^2}$$

again ignoring higher differences.

The quantities in the suffices represent steps of δt and δx. The original equation now becomes :

$$f_{xt+1} - f_{xt-1} = 2sf^p_{xt}\,(f_{x+1,t} - f_{x,t+1} - f_{x,t-1} + f_{x-1,t})$$

where $s = \frac{\delta t}{(\delta x)^2}$ as before.

This method is semi-explicit since it requires a starting procedure. In order to evaluate f_{xt+1} it is necessary to know f_{xt} and f_{xt-1}. This means that if $f_{x,0}$ is given for all x then $f_{x,1}$ must be calculated by some other process before $f_{x,2}$ can be found from the DuFort-Frankel method. The above formula may be put into the more suggestive form :

$$f_{xt+1} = (2sf^p_{xt}\ (f_{x+1t} - f_{xt-1} + f_{x-1t}) + f_{xt-1})/(1 + f^p_{xt}\ 2s).$$

(c) Crank-Nicolson Implicit Method

If higher differences are ignored we have

$$(E_t - 1)f = sf^p \delta^2_x f$$

using a forward difference expression for $\frac{\partial f}{\partial t}$

and

$$(1-E_t^{-1})f = sf^p \delta_x^2 f$$

using a backward difference expression for $\frac{\partial f}{\partial t}$.

The second of these expressions may be written as :

$$(E_t-1)f = sE_t(f^p \delta_x^2 f)$$

by operating on both sides of the equation with E_t.

We now introduce a number r such that

$$0 \leq r \leq 1$$

and form a linear combination of the two representations as follows :

$$(1-r)(E_t-1)f = s(1-r)f^p \delta_x^2 f$$

$$r(E_t-1)f = s\, r\, E_t(f^p \delta_x^2 f)$$

Adding the equations gives :

$$(E_t-1)f = s(r\, E_t + (1-r))\, f^p \delta_x^2 f$$

This gives us the well known Crank-Nicolson formula.

2.2. Representation of Boundary Condition

The boundary condition $\frac{\partial f}{\partial x}\Big|_{x=1} = 0$ may be incorporated as follows. Ignoring terms of higher order

$$\frac{\partial f}{\partial x} = \frac{f_{x+1,t} - f_{x-1,t}}{2\delta x}$$

If the x-axis is divided into steps of equal length then if x = 1 is the mth point, we must introduce a virtual point outside x = 1 at which f(x,t) is represented by $f_{m+1,t}$. Hence at the boundary we have :

$$\frac{\partial f}{\partial x} = \frac{f_{m+1,t} - f_{m-1,t}}{2\delta t} = 0$$

that is

$$f_{m+1,t} = f_{m-1,t}$$

This enables us to find $\delta_x^2 f$ for all x in the formulae derived above.

2.3. The Singularity

In the initial conditions given for the problem, a singularity occurs at x = 0, t = 0. In the neighbourhood of x = 0, t = 0, the finite difference representation is of doubtful value. The difficulty may be avoided in two ways, both of which are described here. The first method utilises the change of variables to X,T given by :

$$X = xt^{-\frac{1}{2}} \qquad T = t^{\frac{1}{2}}$$

Then the equation

$$\frac{\partial f}{\partial t} = f^p(x,t)\frac{\partial^2 f}{\partial x^2}$$

is transformed to

$$\frac{T}{2}\frac{\partial f}{\partial T} = \frac{X}{2}\frac{\partial f}{\partial X} + f^p(X,T)\frac{\partial^2 f}{\partial x^2}$$

which is still parabolic.

In the new system, the point $x = 0$, $t = 0$ becomes the line $T = 0$. The line $t = 0$ becomes the point at infinity on $T = 0$. The boundary line $x = 0$ becomes the line $X = 0$. In effect, the origin has been transformed into a line. The singularity in f then becomes a gradual change. On $T = 0$ we have for the initial values

$$f^p(X,0)\frac{\partial^2 f}{\partial X^2} + \frac{X}{2}\frac{\partial f}{\partial X} = 0$$

or

$$u^p u'' + \frac{X}{2}u' = 0 \quad \text{where} \quad u(X) = f(X,0).$$

The difficulties involved in this approach are obvious. Not only is the new equation more complicated than the original equation, but the solut of the ordinary differential equation for the initial values is not a trivial problem. Even when results have been obtained for the transformed equation, a difficult interpolation process has to be carried out in order to give value of f on a rectangular grid in the original x-t plane.

The second method involves the use of a fine mesh near the singulari and transformation to a coarser mesh when the effects of the singularity are sufficiently small. The chief drawback in this approach is that it involves a change in mesh lengths, which may be inconvenient.

2.4. Truncation Error and Stability

Before proceeding with the discussion of stability for the non-linea case it is necessary to review some standard results for the linear case $p = 0$. It is convenient to consider the linear diffusion equation in the form :

$$\frac{\partial f}{\partial t} = k \frac{\partial^2 f}{\partial x^2}$$

with initial and boundary conditions as before. k is a constant introduced for the purposes of analysis.

In the stability discussion of linear Partial Differential Equations two techniques are widely used. In the first of these it is assumed that the error at any time can be expanded in a Fourier series in x. That is

$$e(t_{0,x}) = \sum_{\beta} T_{0\beta} e^{i\pi\beta x}$$

Since the equation is linear, it is possible to deal with the above series term by term and look for stability conditions for each β - component. This method is extremely convenient for linear equations with initial values given. However, this approach is unsuitable when the boundary conditions involve derivatives. For problems which involve a derivative in some simple way, it is usually possible to apply matrix techniques. We now indicate how this latter method may be applied by considering briefly the Explicit Forward Difference representation.

The Explicit Forward Difference formula for the linear diffusion equation is

$$(E_t - 1)f = ks\delta_x^2 f \quad \text{where} \quad s = \frac{\delta t}{(\delta x)^2}$$

and k is the constant introduced earlier.

It is convenient to re-introduce the subscripts which were dropped previously. We have :

$$f_{x,t+1} = f_{x,t} + ks(f_{x-1,t} - 2f_{x,t} + f_{x+1,t}).$$

This sytem of equations may be expressed concisely in the following matrix formula :

$$\underline{f}_{t+1} = B\,\underline{f}_t + \underline{c}_t$$

where

$$\underline{f}_t = \begin{bmatrix} f_{1t} \\ f_{2}t \\ \\ f_{nt} \end{bmatrix} ; \quad \underline{c}_t = \begin{bmatrix} s\, f_{0s} \\ 0 \\ 0 \\ \\ 0 \end{bmatrix}$$

B is the matrix.

$$\begin{bmatrix} 1-2ks & ks & & & \bigcirc \\ ks & 1-2ks & ks & & \\ & & & & \\ \bigcirc & & ks & 1-2ks & ks \\ & & & 2ks & 1-2ks \end{bmatrix}$$

B is in fact a tri-diagonal matrix.

We now suppose that each $\underline{f}_t$ is subject to error $\underline{e}_t$

Then $$\underline{f}_{t+1} + \underline{e}_{t+1} = B(\underline{f}_t + \underline{e}_t) + \underline{c}_t$$

and $$\underline{f}_{t+1} = B\,\underline{f}_t + \underline{c}_t$$

Therefore $$\underline{e}_{t+1} = B\,\underline{e}_t$$

Also, we suppose that B has n distinct vectors $u_1 \ldots . u_n$.

We may write

$$\underline{e}_o = a_1\underline{u}_1 + a_2\underline{u}_2 + \ldots + a_n\underline{u}_n$$

where $a_1, a_2 \ldots . a_n$ areconstants.

This expansion is the matrix equivalent of the Fourier Expansion done earlier.

Now

$$\underline{e}_{t+1} = B^t \underline{e}_o$$

$$= (\lambda_1)^t a_1\underline{u}_1 + (\lambda_2)^t a_2\underline{u}_2 + \ldots + (\lambda_n)^t a_n\underline{u}_n$$

where $\lambda_1, \lambda_2 \ldots \lambda_n$ are the latent roots of B.

Now

$$\lim_{t \to \infty} \underline{e}_{t+1} = 0$$

can only be true if $|\lambda_i| < 1$.

The error is bounded if $|\lambda_i| = 1$.

Hence the process is stable if the moduli of the latent roots of B are less than, or equal to unity. The latent roots of B are given by

$$|B - \lambda I| = 0.$$

It may be shown by consideration of the roots of the characteristic equation of B that the conditions for stability are :

$$k > 0; \quad s\,k \leqslant \tfrac{1}{2}.$$

Similar discussions may be produced for the other two formulae given previously. Here we shall merely summarise the results obtained. For the DuFort-Frankel formula, the stability conditions are :

$k > 0$, s unrestricted.

For the Crank-Nicolson process the stability conditions are :

$k > 0$,

$s\ k \leq \frac{1}{2(1-2r)}$ for $0 \leq r < \frac{1}{2}$,

s unrestricted for $\frac{1}{2} \leq r \leq 1$

where r is the parameter introduced earlier in the discussion.

The truncation error involved in the three representations may be examined by writing out the series for E_t and δ_x^2 in full.

This gives :

$$(E_t - 1)f = \left\{ \delta t D_t + \frac{(\delta t)^2}{2!} D_t^2 + \frac{(\delta t)^2}{3!} D_t^3 + \ldots + \ldots \right\} f$$

$$\delta_x^2 f = \left\{ (\delta x)^2 D_x^2 + \frac{2}{4!} (\delta x)^4 D_x^4 + \frac{2}{6!} (\delta x)^6 D_x^6 + \ldots \right\} f$$

where $D_x \equiv \frac{\partial}{\partial x}$ and $D_t \equiv \frac{\partial}{\partial t}$

Using these series it may be shown that for the Explicit Forward Difference formula

$$\frac{(E_t - 1)f}{\delta t} - \frac{k \delta_x^2 f}{(\delta x)^2} = \left[\frac{k(\delta x)^2}{2!} \left(\frac{k \delta t}{(\delta x)^2} - \frac{1}{6} \right) D_x^4 + \frac{1}{6}(k(\delta x)^4) \left(\frac{k^2 (\delta t)^2}{(\delta x)^4} - \frac{2}{5!} \right) D_x^6 + \ldots + . \right.$$

Thus the truncation error for the Explicit Forward Difference Formula is :

$$O(\delta t) + O\left[(\delta x)^2\right]$$

If we choose $\frac{\delta t}{(\delta x)^2}$ so that

$$\frac{k \delta t}{(\delta x)^2} = \frac{1}{6}$$

then the system is stable and the truncation error is $O[(\delta x)^4]$. Application of series for E_t and δ_x^2 to the DuFort-Frankel formula shows that the error is :

$$e = O(\delta t) + O[(\delta x)^2] + O\left[(\frac{\delta t}{\delta x})^2\right]$$

An interesting feature emerges from this analysis of the truncation error for the DuFort-Frankel method. The term :

$$k(\frac{\delta t}{\delta x})^2 \frac{\partial^2 f}{\partial t^2}$$

appears in the error expression. A finite difference representation is said to be consistent with the corresponding differential equation if the truncatio error tends to zero as δt and δx tend to zero. It is clear that in this metho consistency demands that δt tends to zero faster than δx. Certainly, we must have $\frac{\delta t}{\delta x} < 1$. If this is not true and

$$\lim_{\delta t, \delta x \to 0} (\frac{\delta t}{\delta x})^2 = \text{const}$$

then the finite difference representation is consistent with an equation of the form :

$$\frac{\partial f}{\partial t} - k\frac{\partial^2 f}{\partial x^2} + (\text{const}) \frac{\partial^2 f}{\partial t^2} = 0$$

and this equati n is hyperbolic. The practical outcome of this behaviour is that the ratio $\frac{\delta t}{\delta x}$ must be kept small. In fact, this destroys some of the advantage of unrestricted stability.

Finally, for the Crank-Nicolson formula, it may be shown that by using the sameseries as before, the truncation error is given by :

$$e = O(\delta t) + O\left[(\delta x)^2\right]$$

Although the use of truncated series may involve some error unless δt and δx are taken to be vanishingly small, in this case, it avoids the possibility of increasing spurious solutions. Increasing spurious solutions sometimes appea when the finite difference representation of a differential equation includes differences of higher order than the corresponding derivatives in the differential equation.

In the past, attempts to treat the stability of non-linear Partial Differential Equations by rigorous analytical methods have generally failed. However, it is often possible to develop an empirical model for stability by suitable linearisation of the equation under investigation. This is, in fact, the procedure adopted here, but before this is dealt with, a more rigorous discussion on instability is presented for the Explicit Forward Difference formula.

The finite difference scheme is :

$$(E_t-1)f = s\, f^p \delta_x^2 f$$

Let $f_{x,t}$ be the true solution of the above equation. Let $g_{x,t}$ be the rounded solution.

We define

$$e_{x,t} = f_{x,t} - g_{x,t}$$

Then

$$(E_t-1)f_{x,t} = s\, f_{x,t}^p\, \delta_x^2\, f_{x,t}$$

$$(E_t-1)g_{x,t} = s\, g_{x,t}^p\, \delta_x^2\, g_{x,t}.$$

Dropping subscripts and subtracting the second equation from the first gives :

$$(E_t-1)e = s\, f^p \delta_x^2\, e + s(f^p-g^p)\delta_x^2\, g$$

Now, if p is a positive integer we have :

$$\frac{f^p-g^p}{f-g} = f^{p-1} + f^{p-2}g + f^{p-3}g^2 + \dots + f^2 g^{p-3} + f\, g^{p-2} + g^{p-1}$$

and we may dmnote the right hand side of this by $B(f,g)$.

Hence

$$(E_t-1)\, e = s\, f^p \delta_x^2\, e + s\, e\, B(f,g)\delta_x^2 g$$

Unfortunately, there seems to be no way in which this partial difference equation may be either solved or simplified. This serves to show that the error varies from point to point in a complicated way.

We now turn to the empirical discussion of stability. It has already been noted that the equation under consideration bears a strong resemblance to the linear diffusion equation :

$$\frac{\partial f}{\partial t} = k\frac{\partial^2 f}{\partial x^2}$$

for which stability criteria have been derived rigorously. The obvious approach is to replace k by f^p in the appropriate expressions on the understanding that it may be necessary to choose the parameters s and r pessimistically in order to ensure stability for all values of f^p likely to be encountered in the solution of the problem. When this is done the following results emerge.

For the Forward Difference formula

$$f^p > 0,$$

$$f^p s < \tfrac{1}{2}$$

For the DuFort-Frankel method

$$f^p > 0$$

For the Crank-Nicolson method

$$f^p > 0,$$

$$s\, f^p < \frac{1}{2(1 - 2r)} \quad \text{for } 0 \leq r < \tfrac{1}{2}$$

$$s \text{ unrestricted for } \tfrac{1}{2} \leq r \leq 1.$$

Now an interesting possibility emerges if p is odd and f takes negative values. Under these circumstances the linearised theory suggests that all three methods are unstable. Otherwise we should hope to find stability if s and r are chosen suitably with f^p given its maximum value in the inequalitie Of course, it must be emphasised that the linear model is only really meaningful in regions where f^p varies so slowly that the non-linear equation behaves like a linear diffusion equation.

The truncation error involved at each step in the integration of the non-linear equation may be examined by methods similar to those used for the linear case.

For the Explicit Forward Difference Representation we have :

$$(E_t - 1)f = s\, f^p \delta_x^2 f$$

Expanding the left hand side by Taylor series gives :

$$(E_t-1)f = \delta t \; \frac{\partial f_{x,t}}{\partial t} + \frac{(\delta t)^2}{2!} \; \frac{\partial^2 f_{x,t}}{\partial t^2} + \dots..$$

and in the right hand side expension of $\delta_x^2 f$ gives :

$$\delta_x^2 f = f_{x+1t} - 2f_{xt} + f_{x-1t}$$

$$= (\delta x)^2 \frac{\partial^2 f_{xt}}{\partial x^2} + \frac{2}{4!} (\delta x)^4 \; \frac{\partial^4 f_{xt}}{\partial x^4} + \dots.$$

By substituting these expressions in the finite difference representation we find that the truncation error e is given by :

$$e = + \frac{(\delta t)}{2} \; \frac{\partial^2 f}{\partial t^2} - \frac{2f^p}{4!} (\delta x)^2 \; \frac{\partial^4 f}{\partial x^4} + \dots.$$

This expression shows that δx must be kept small if the factor f^p in the second term is large. In the non-linear case it is not possible to choose s in a special way in order to make the predominant terms in the truncation error of the fourth order.

In the DuFort-Frankel representation

$$\frac{f_{xt+1} - f_{xt-1}}{2 \, \delta t} = \frac{f^p_{xt}}{(\delta x)^2} (f_{x+1t} - f_{xt+1} - f_{xt-1} + f_{x-1t})$$

By applying the same techniques as before, we find that the truncation error e is given by :

$$e = \frac{(\delta t)^2}{3!} \frac{\partial^3 f}{\partial t^3} - \frac{2}{4!} f^p (\delta x)^2 \frac{\partial^4 f}{\partial x^4} + (\frac{\delta t}{\delta x})^2 f^p \; \frac{\partial^2 f}{\partial t^2} + \dots.$$

Again the phenomenon of inconsistency appears. While δt and δx may both be small enough to make the first two terms in the expression for e negligible, the ratio $\frac{\delta t}{\delta x}$ may be large enough to make the behaviour of the third term significant. In cases where this occurs, the DuFort-Frankel representation would be consistent with

$$\frac{\partial f}{\partial t} - f^p \frac{\partial^2 f}{\partial x^2} + \left(\frac{\delta t}{\delta x}\right)^2 f^p \frac{\partial^2 f}{\partial t^2} = 0$$

The Crank-Nicolson representation is :

$$\frac{(E_t - 1)}{\delta t} = \frac{(rE_t + (1-r))\, f^p \delta_x^2 f}{(\delta x)^2}$$

The truncation error is found to be :

$$e = \frac{\delta t}{2} \frac{\partial^2 f}{\partial t^2} - \frac{2}{4!} (\delta x)^2 f^p \frac{\partial^4 f}{\partial x^4} - r(\delta t)\, f^p \frac{\partial^3 f}{\partial x^2 \partial t} - r \delta t \frac{\partial (f^p)}{\partial t} \cdot \frac{\partial^2 f}{\partial x^2} + \ldots$$

In this method the representation is consistent but, as usual, the truncati error may vary significantly as f^p changes.

This concludes the discussion of stability and truncation error.

2.5. Treatment of the Crank-Nicolson Method in General

The details of the two explicit methods are trivial as far as programming is concerned. The advantage of explicit methods lies in the fact that each step in the calculation only requires the evaluation of an arithmetic expression.

On the other hand, the Crank-Nicolson implicit method involves the solution of simultaneous non-linear equations for $p \neq 0$. The Crank-Nicolson formula may be written in the form :

$$E_t(f_{x,t} - srf^p_{x,t}\delta^2_x f_{x,t}) = f_{x,t} + s(1-r)f^p_{x,t}\,\delta^2_x\, f_{x,t}$$

Since the right hand side of this equation is always known it may be replaced by $\alpha_{x,t}$. The equation may be reduced to :

$$f_{x,t+1} - srf^p_{x,t+1}(f_{x+1,t+1} - 2f_{x,t} + f_{x-1,t+1}) = \alpha_{x,t}$$

Substituting y for $f_{x,t+1}$ for convenience gives

$$y + 2\,sry^{p+1} - sry^p(f_{x+1,t+1} + f_{x-1,t+1}) = \alpha_{x,t}$$

For any value of t there are m such equations in m quantities of the type $f_{x,t+1}$. Hence we have to solve m equations in m unknowns with the special condition $f_{x+1,t+1} = f_{x-1,t+1}$ at the boundary, though this last condition is just a computational detail.

If p is an integer, then the typical equation is similar to a polynomial with one undetermined coefficient.

At this point we may consider various ways of handling the simultane equations which appear in the Crank-Nicolson method. The first possibility is the use of linearisation. This technique has been used successfully in the solution of the Partial Differential Equation :

$$\frac{\partial f}{\partial t} = \frac{\partial^2}{\partial x^2}\,(f^5)$$

The linearisation is carried out as follows, using Taylor Series :

$$f^{p}_{x,t+1} \approx f^{p}_{x,t} + \delta t \frac{\partial}{\partial t}(f^{p}_{x,t})$$

$$\approx f^{p}_{x,t} + \delta t \; p \; f^{p-1}_{x,t} \frac{\partial f_{xt}}{\partial t}$$

$$\approx f^{p}_{x,t} + p \; f^{p-1}_{x,t} \; (f_{x,t+1} - f_{x,t})$$

where terms of order $(\delta t)^2$ have been neglected. The solution of the simultaneous linear equations may then be effected by standard matrix methods. This method is not used here for the following reasons. Firstly, the truncated Taylor series introduces additional error and in some cases the linearisation may destroy the advantage of unrestricted stability. Secondly, linear methods have been treated exhaustively and the object of this investigation is to retain the non-linearity for as long as possible.

Another method for the solution of non-linear systems uses a generalisation of Newton-Raphson Iteration. If the system of equations is

$$\phi_i(x) = 0$$

where x is the vector of unknowns ξ_j; then if each function ϕ_i is well-behaved in the neighbourhood of x_o there exists an iterative process :

$$x_n = x_{n-1} - f_x^{-1}(x_{n-1}) f(x_{n-1})$$

In this process, f is the vector whose elements are the ϕ_i and $f_x = \left[\frac{\partial \phi_i}{\partial \xi_j}\right]$ which is just the Jacobian matrix of the ϕ_i. This process may be shown to be convergent, but we avoid it here because repeated calculation of the matrix f_x^{-1} is undesirable. In the problem considered here f_x is a tri-diagonal matrix but even for this simple form of matrix, exact inversion is not trivial and the use of a sub-iteration for the inverse only complicates programming.

We now consider a third approach to the solution of the simultaneou non-linear equations of the Crank-Nicolson method. In this method it is assumed that the undetermined coefficient $(f_{x+1,t+1} + f_{x-1,t+1})$ in the typical equation :

$$y + 2ary^{p+1} - ary^{p}(f_{x+1,t+1} + f_{x-1,t+1}) = \alpha_{x,t}$$

can be found approximately. This means that we require some process for generating approximate values for all the $f_{x,t+1}$. Assuming that approximate values of $f_{x,t+1}$ can be found for all x and that p is an integer, we are faced with the solution of sets of polynomial equations. It is tempting to consider the use of explicit formulae for the solution of these polynomials in the cases p = 1, 2, 3, but this idea may be rejected for two reasons.

Firstly, analytical formulae for p = 1, 2, 3 give all the roots of the equation and this involves selection of the "correct" root. Even when this can be done logically, further complication has been introduced into the programming, and this is undesirable. The difficulty emerges at once in the simple case p = 1 where the resulting quadratic equations have solutions of the type :

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

using standard notation for the coefficients.

Secondly, using the improved values for $f_{x,t+1}$ it will be necessary then to set up the simultaneous equations onee more using the improved value for the undetermined coefficient. Clearly, there is no advantage to be gained by solving the equations exactly at each stage. It is actually easier to use iterative methods for the solution of the equations and this approach is developed in the next section.

2.6. Iterative Methods for the Crank-Nicolson Formula

In this section we construct a detailed scheme for the programming of the Crank-Nicolson method, but before dealing with the general case we find it instructive to look at the case p = 0. The case p = 0 illustrates simply many of the points which emerge in the general case. For p = 0 the Partial Differential Equation is linear and the Crank-Nicolson scheme becomes

$$(E_t - 1)f = s(rE_t + 1 - r)\delta_x^2 f$$

Thus we have a set of simultaneous linear equations for $f_{x,t}$ and this set may be expressed in matrix form as Ax = b where x is the vector of unknowns $f_{x,t}$ and A is the matrix

$$\begin{bmatrix} 1+2rs & -rs & & & \bigcirc \\ -rs & 1+2rs & -rs & & \\ & \ddots & \ddots & \ddots & \\ \bigcirc & & -rs & 1+2rs & -rs \\ & & & -2rs & 1+2rs \end{bmatrix}$$

A is, in fact, a tri-diagonal matrix and is also symmetric if boundary values are specified. A may be written in the form :

$$A = D + u$$

where D is a diagonal matrix and u is a matrix with zeros on its principal diagonal. Then we have :

$$(D + u)x = b$$

or $$Dx + ux = b$$

Now we consider the iterative matrix scheme represented by :

$$Dx_{n+1} + ux_n = b$$

This may be written in the form :

$$x_{n+1} = D^{-1}b - D^{-1}ux_n$$

and we note that the inverse of D is obtained simply by inverting its non-zero elements. Also, if x is the actual solution of Ax = b then

$$x = D^{-1}b - D^{-1}ux$$

We now define $\varepsilon_n = x - x_n$, where ε_n is just the error at the nth iteration.

By subtracting x_{n+1} from x we have :

$$x - x_{n+1} = -D^{-1}u(x - x_n)$$

That is

$$\varepsilon_{n+1} = -D^{-1}u\,\varepsilon_n$$

Continued application of this last relationship gives :

$$\varepsilon_{n+1} = (-1)^{n+1}\,(D^{-1}u)^{n+1}\varepsilon_o$$

and ε_o is the error in the initial approximation to x. If $D^{-1}u$ has m distinct latent roots $\lambda_1, \dots \lambda_m$ and corresponding latent vectors $u_1, \dots u_m$ then it is possible to write

$$\varepsilon_o = a_1 u_1 + a_2 u_2 + \dots + a_m u_m$$

where $a_1, \dots a_m$ are constants.

This gives :

$$\varepsilon_{n+1} = (-1)^{n+1}(D^{-1}u)^{n+1}(a_1 u_1 + a_2 u_2 + \dots + a_m u_m)$$

$$= (-1)^{n+1}(a_1 \lambda_1^{n+1} u_1 + a_2 \lambda_2^{n+1} u_2 + \dots + a_m \lambda_m^{n+1} u_m)$$

Obviously, the condition for the convergence of the iterative scheme is that $|\lambda_i| < 1$ for all i. Multiplying u by D^{-1} gives :

$$D^{-1}u = \begin{bmatrix} 0 & \frac{-r s}{1 + 2 r s} & & & \\ \frac{-r s}{1 + 2 r s} & 0 & \frac{-r s}{1 + 2 r s} & & \huge 0 \\ & & \ddots & & \\ \huge 0 & & & & \frac{-r s}{1 + 2 r s} \\ & & & \frac{-2 r s}{1 + 2 r s} & 0 \end{bmatrix}$$

and the roots of $|D^{-1}u - \lambda I| = 0$ may be found analytically and it emerges that the conditions for convergence of the iterative process are s, r ≥ 0. These conditions are always satisfied. An important feature concerning r may be seen. The error $\mathcal{E}_{n+1}$ is determined by the rate at which the elements of $(D^{-1}u)^{n+1}$ approach zero. The non-zero elements of $(D^{-1}u)^{n+1}$ consist of products which depend on the magnitude of

$$\frac{-rs}{(1+2rs)}$$

Now if $0 \leq r \leq 1$ then

$$\frac{rs}{(1+2rs)} \leqslant \frac{s}{(1+2s)}$$

Thus we expect faster convergence of the iterations for $r < 1$, although the Crank-Nicolson stability condition demands that $r > \frac{1}{2}$ for unrestriced stability. The typical equation in the linear case is :

$$(1+2rs)f_{x,t+1} - (f_{x+1,t+1} + f_{x-1,t+1})rs = \alpha_{x,t}$$

In the iterative process considered here we know the factor

$$(f_{x+1,t+1} + f_{x-1,t+1})$$

approximately and we wish to find improved values of $f_{x,t+1}$ for all x. Clearly, in order to do this, we have to solve simple linear equations of the type :

$$ax_j - b(x'_{j+1} + x'_{j-1}) = c$$

where the dash denotes approximate values. We may apply Newton-Raphson iterati to each equation in turn giving

$$x_j^{(r+1)} = x_j^{(r)} - \frac{(ax_j^{(r)} - b(x'_{j+1} + x'_{j-1}) - c)}{a}$$

$$= \left[b(x^{r}_{j+1} + x^{r}_{j-1}) + c\right]/a$$

r denotes the rth iteration. It is quite clear that we may replace the dash by (r) to give the iterative process :

$$x_j^{(r+1)} = \left[b(x^{(r)}_{j+1} + x^{(r)}_{j-1}) + c\right]/a$$

This is just the same result as we obtained earlier using matrix notation. The significant difference is that we have dealt with each equation individually, and the iterative process merely corrects one coefficient in the linear relation. This idea is the basis of the method used in the general non-linear case. The technique is worthwhile as long as the term to be corrected is fairly small compared to definitely known terms and terms containing the iterated quantity. That this condition is reasonable can be seen at once from the form of the simultaneous equations.

We now consider the details of the iterative process used to solve the Crank-Nicolson simultaneous equations. The basic steps in the process may be summarised as follows. Assume that :

(a) There is some systematic procedure for finding a first approximation to $f_{x,t+1}$ for all x.

(b) We treat each equation of the type :

$$y + 2\,a\,r\,y^{p+1} - a\,r\,y^{p}(f_{x+1,t+1} + f_{x-1,t+1}) = \alpha_{x,t}$$

as an equation in the unknown y, with the coefficient $(f_{x+1,t+1}+f_{x-1,t+1})$ determined by the first approximation.

(c) Each equation in y may be processed by some method to obtain a closer approximation to each $f_{x,t+1}$.

(d) The closer approximation may be used to redetermine each unknown coefficient of the form $(f_{x+1,t+1} + f_{x-1,t+1})$.

(e) Steps (b), (c) and (d) are repeated until the successive approximations differ by less than some prescribed amount. This is the basis of the implicit method. The details of the programming are dealt with later. We now give detailed consideration to steps (a) to (e).

Let $b = (f_{x+1,t+1} + f_{x-1,t+1})/2$

$a = 2\,s\,r$

and drop all suffices. We are now considering the solution by iteration of equations of the type :

$$y + ay^{p+1} - aby^{p} - \alpha = 0$$

Two iterative processes for the solution of this equation suggest themselves immeditely.

(a) Newton-Raphson iteration.

(b) Simple functional iteration.

Since the above equation may be written in many different forms, any one of which can be treated by the two methods indicated above, we shall simply choose two forms which are easy to handle.

(1) In the Newton-Raphson formula

$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}$$

choose $$f(y) = y + ay^{p+1} - aby^{p} - \alpha$$

whence $$f'(y) = 1 + a(p+1)y^{p} - abpy^{p-1}$$

(2) In the functional iteration process the equation is written in the form :

$$y(1 + ay^{p} - aby^{p-1}) = \alpha$$

and for the iterative process we take :

$$y_{n+1} = \frac{\alpha}{(1 + ay_n^{p} - aby_n^{p-1} \quad)}$$

This deals with assumption (c).

There are two important considerations for step (a). Firstly, given a reasonably good approximation to $f_{x,t+1}$, the iteration must converge. Secondly, a process for generating good approximations to $f_{x,t+1}$ must be found for large S. This last point is one which can only be investigated numerically. The question of convergence of the iterations is examined in some detail below.

Suppose $g(x)$ has no pole which coincides with a root of $f(x) = 0$. If

$$h(x) = x - g(x)\,f(x)$$

then any root of $f(x) = 0$ satisfies $x = h(x)$. If x_1 is a root of $f(x) = 0$ and if $g(x)$ is analytical and non-zero in the neighbourhood of x_1 then x_1 is the only root of

$$h(x) = x - g(x).f(x)$$

in this neighbourhood of x_1. This suggests the possibility of choosing

$g(x)$ so that the sequence

$$x_{i+1} = h(x_i)$$

converges to x_1 provided that the initial point x_0 is sufficiently close to x_1. In fact, it may be shown that the condition

$$\left| h'(x_1) \right| < 1$$

is sufficient to ensure the existence of some neighbourhood of x_1 in which $x_{i+1} = h(x_i)$ converges.

It may also be shown that convergence is more rapid for smaller $\left| h'(x_1) \right|$. Clearly it would be an advantage to make $h'(x_1) = 0$. When this is so, the sequence $x_{i+1} = h(x_i)$ is said to have second order convergence. We may apply these results to the two iterative methods outlined earlier. In the Newton-Raphson formula we have :

$$h(x) = x - \frac{f(x)}{f'(x)}$$

Therefore

$$h'(x) = 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{[f'(x)]^2}$$

$$= \frac{f(x)f''(x)}{[f'(x)]^2}$$

whence $$h'(x_1) = \frac{f(x_1)f''(x_1)}{[f'(x_1)]^2} = 0$$

assuming $f'(x_1) \neq 0$.

The assumption $f'(x_1) \neq 0$ is the same as saying that $f(x)$ must not have a repeated root at x_1.

This discussion shows that the Newton-Raphson method is Second Order in general, and hence the particular Newton-Raphson process used here is Second Order. If $f'(x_1) = 0$, that is if there is a repeated root of $f(x)$ at x_1, then $\frac{f(x_1)}{f'(x_1)}$ is indeterminate. Thus for equations with repeated or nearly equal roots the Newton-Raphson process is ill-conditioned

In the direct iteration we have :

$$y_{n+1} = \frac{\alpha}{(1 + ay_n^p - aby_n{}^{p-1})}$$

and here

$$h(y) = \frac{1}{(1+ay_n^p-aby^{p-1})}$$

therefore

$$h'(y) = \frac{y^{p-2}(apy - ab(p-1))}{(1+ay - aby^{p-1})^2}$$

If x_1 is a root of the equation represented by this iterative process then $h'(x_1) \neq 0$ except in pathological cases and hence the process is first order. Also, this process is well-conditioned for multiple roots. To sum up this analytical discussion of the two iterative methods we may say that although the convergence of the Newton-Raphson process is of higher order than the direct iteration, it is arithmetically more complicated and sometimes ill-conditioned. Moreover, in the direct process the numerator is constant at each stage in the iteration and this fact is extremely convenient for hand calculations. It is also useful in machine computation since the quantity α need only be found once for each x and t. The question of the existence of real roots of the equations cannot be dealt with in advance. The only control is to ensure that initial approximations are reasonably good. The use of accelerative devices such as Aitken's δ^2 process and iterative methods of third or higher orders introduces more powerful convergence at the cost of simple programming and has been avoided here. It is conceivable that such techniques are useful for large values of S.

At this point it is worth mentioning that if the equation had been quasi-linear and if instead of f^p there had been some explicit function of x and t, then the solution of the system of equations which emerges in the Crank-Nicolson method would have been much easier. In fact, such a system could be handled by iterative matrix methods. This is a brief indication of the increase in complexity introduced by having a non-linear factor.

First approximations for use in the Crank-Nicolson method may be generated in three ways :

1. By means of the explicit formula. This method is ideal for use when testing the convergence of the iterations, subject to the stability restrictions on the explicit formula.
2. By using an extrapolation formula. The most convenient formula is the Newton Backward Difference formula :

$$f_{x,t+1} = f_{x,t} + \nabla f_{x,t} + \nabla^2 f_{x,t} + \ldots\ldots$$

In this formula $f_{x,t+1}$ is estimated from a knowledge of $f_{x,t}$, $f_{x,t-1}$, $f_{x,t-2}$, ...

Since the convergence of this formula is generally poor, it is enough to take just a few terms in the series. No great advantage is gained by considering differences higher than the third or fourth. This formula provides an independent means of testing the Crank-Nicolson method. It is free from the stability restrictions of explicit formulae, but it is clearly a poor approximation when compared with stable explicit representations.

3. The simplest method is to approximate $f_{x,t+1}$ by $f_{x,t}$. Obviously, this method requires a powerful iteration process.

There is another way of looking at the iterative scheme used here. The scheme may be seen to be a particular case of the generalised Newton-Raphson iteration when the Newton-Raphson process is applied to the equations individually. The generalised Newton-Raphson process which was briefly mentioned earlier is

$$x_{n+1} = x_n - f_x^{-1}(x_n)f(x_n)$$

where

x is the vector of unknowns

f(x) is the vector of equations

f_x^{-1} is the inverse of the Jacobian matrix of f(x)

x_{n+1} is a closer approximation to the roots of f(x) than x_n provided that f_x^{-1} can be found sufficiently closely and assuming good initial approximations.

There is really no point in finding f_x^{-1} by exact methods since x_{n+1} need only be a closer approximation. It is more in keeping with the spirit of iteration to use a good approximation to f_x^{-1}. This point is similar to the one we discussed earlier regarding the exact solution of the individual equations by explicit analytical formulae; such an approach is usually laborious without giving any marked advantages. The special form of the simultaneous equations in the Crank-Nicolson method allows a considerable simplification. The matrix f_x^{-} is actually tri-diagonal in form. Assuming that the off-diagonal entries are small compared with the diagonal entries, it should be possible to approximate f_x^{-1} by D_x^{-1} where D_x is the matrix of diagonal entries. The scheme may still converge successfully even when this condition is not satisfied, since we are not interested in finding f_x^{-1} to a high degree of precision by iteration. We only require that the Newton-Raphson process gives successive approximations which increase in accuracy. This is just the scheme which has been used in this investigation. The possibility of using D_x^{-1} in this way makes the adoption of a non-linear approach worthwhile. The advantages to be gained by retaining the non-linearity and using the Newton-Raphson iteration method are quite clear. They are as follows :

(a) There are no linear approximations at an early stage in the computation. While we have introduced an approximation in the form $D_x^{-1} \approx f_x^{-1}$ it matters less in a convergent iterative process where there is no explicit truncation error involved. The only significant drawback in this approximation is that the rate of convergence may be slightly less.

(b) The strong convergence of the Newton-Raphson iteration avoids the exact inversion of matrices. The main disadvantage introduced by non-linear iteration is that there is some increase in the complexity of the programming.

This concludes the discussion of iterative schemes for the solution of the Crank-Nicolson simultaneous equations.

Chapter 3

3.1. General

In this chapter, three aspects of the problem are discussed. This section deals with the practical details of the investigation. The next section contains a summary of the numerical results obtained. In the last section the results are discussed in detail and some general features are examined.

In the programming of the methods discussed in Chapter 2, the approach was experimental. The objectives of the study were as follows :

(a) To find numerical solutions of the problem using the three formulae developed earlier and to test these formulae in as many ways as possibl

(b) To test the usefulness of the empirical stability theory.

(c) To compare a numerical solution with an analytical solution.

In the first programme, the simple explicit method was mechanised and solutions were found for the case p = 1 for various values of S and for other values of p and S with different initial conditions. This establishes stability conditions and enables us to programme the other two methods. As soon as stability conditions were established, the results of the explicit method were used to start the DuFort-Frankel formula. The DuFort-Frankel method was investigated for various values of $\delta t/(\delta x)$ but could not be run with p odd and negative function values since no stable starting procedure was known. Finally, the Crank-Nicolson method was investigated under a wide variety of circumstances. The stability theory for this method was checked in considerable detail, for different values of p and various function values. The two iterative processes were programmed and compared. A simple method for speeding up convergence near the singularity was attempted. This consisted of using the result of the current iteration for (m-1)th equation to correct the term $f_{x-1,t+1}$ in the mth equation. All three initial approximation procedures were examined in fair detail. The programmes were tested with as many combinations of the parameters s, p and r as possible.

We now consider in detail the method whereby the analytical solutio was obtained. In the first chapter various analytical solutions were discuss and were found to be too complicated to programme quickly. However, it was found that the case p = -1 gives a simple analytical solution. It may be verified by differentiation that a solution of :

$$f\,\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2} \text{ in } 0 \le x < 1$$

is

$$f(x,t) = \tfrac{1}{2}\tan(\pi/4 + \frac{x}{2} + t)$$

with the initial conditions

$$f(x,0) = \tfrac{1}{2}\tan(\pi/4 + \frac{x}{2})$$

and with time-varying boundary conditions

$$f(0,t) = \tfrac{1}{2}\tan(\pi/4 + t)$$

$$f(1,t) = \tfrac{1}{2}\tan(\pi/4 + \tfrac{1}{2} + t)$$

The main Crank-Nicolson programme was adapted to solve this equation by making the following essential modifications :

1. The boundary condition $\frac{\partial f}{\partial x}\Big|_{x=1} = 0$ was removed.
2. The programme was adjusted to calculate $f(x,0)$ for points in $0 \leqslant x \leqslant 1$ using the expression

$$\tfrac{1}{2} \tan \left(\pi/4 + \frac{x}{2} \right)$$

   and to use these values as initial data.
3. The boundary values $f(0,t)$ and $f(1,t)$ were similarly calculated and imposed automatically.

The rest of the programme was left substantially intact, and suitable values of s and r were used as input. Arrangements were made to print out the calculated theoretical solution beside the numerical solution, obtained by iteration. The values of x and t were suitably restricted in order to avoid the discontinuity which appears at

$$\pi/4 + \frac{x}{2} + t = \pi/2$$

in the theoretical solution. Such a process as we have just described assumes that the solution of the Partial Differential Equation is unique, given the initial and boundary conditions mentioned. The comparison of the theoretical and numerical solutions is important in this study, since the main problem has no analytical solution and difficult features such as singularities and truncation error, have not been adequately handled. Also, it is of interest to see how well the Crank-Nicolson method works when it is free from the effects of the singularity and the stability can be checked in isolation.

3.2. Summary of Results

In this section we summarise the qualitative findings of the investigation. Numerical results are given in the Appendix, and the various methods and techniques are compared in detail there.

For the simple explicit method, the stability conditions are found to be in accord with those suggested by the empirical theory. For $f(x,t) > 0$ stability is obtained provided

$$fs < \tfrac{1}{2} \quad \text{for} \quad p = 1$$

For values of $f(x,t)$ less than zero, the stability condition

$$f^p s < \tfrac{1}{2}$$

holds if p is an even integer. For negative values of $f(x,t)$ and $p = 1$, the numerical solution is violently unstable. It is found that when the stability condition is not satisfied then instability occurs.

The DuFort-Frankel method gives some results which disagree considerably with those obtained by the other two methods. Violent instability is not found to appear, but the numerical results are found to differ quite considerably for larger values of s.

In the Crank-Nicolson method it is found that the stability conditions are the same as those obtained empirically, namely

$$f^p > 0$$

$$sf^p < \frac{1}{2(1-2r)} \qquad \text{for } 0 \leq r \leq \tfrac{1}{2}$$

* For $r \underset{\approx}{>} \tfrac{1}{2}$ the method is found to give stability for all the values of s examined. The convergence of the iterative process is particularly good

for values of s just greater than ½. Of the three methods used to generate first approximations for the iterative processes, the explicit formula is the best, assuming that it is used stably. It is also found that the Newton-Raphson iteration converges much faster than the direct iteration and that the direct iteration process fails to converge for large values of s. The attempt to speed up the convergence of the iterations is only slightly successful. The comparison of analytical and numerical results for the case $p=-1$ shows that the general iterative scheme is reasonable. The results obtained for various values of p with the explicit and Crank-Nicolson methods show fair agreement, and are in accord with the empirical stability theory as far as they are investigated.

3.3. Discussion of Results and General Conclusions

The main results which emerge from the investigation are quite straightforward; they are listed below.

1. The empirical stability theory gives a good picture of the stability of the three methods used.
2. The DuFort-Frankel method is awkward to programme and gives unreliable results; the other two methods are in good agreement when used under stable conditions.
3. The analytical solution and the numerical solution for $p=-1$ are in good agreement, considering that the initial values of $f(x,t)$ are represented by only a few points along the x-axis.

We now discuss a few more general points. In the first place, it may be asked why implicit methods are used at all when a simple explicit method is available. It is true that for small values of p the explicit method is much simpler, but for higher values of p and larger function values the stability condition

$$r^p s < \frac{1}{2}$$

becomes too restrictive. The Crank-Nicolson method is easily arranged to avoid this stability restriction. The chief drawback encountered in the use of the Crank-Nicolson method is the difficulty of generating first approximations for larger values of s. In order to find first approximations, it is necessary to use either $f(x,t-\delta t)$ or the Newton Backward Difference Formula. It is also essential to use a value of r just greater than ½ in order to obtain rapid convergence. There is an incidental advantage in using the implicit method for a problem with a discontinuity such as we have here. After one step δt of the explicit method, only one function value has changed and usually many steps must be considered before all the function values are varying with time. In the iterative method, the discontinuity has time to propogate before the iterations stop and the boundary condition has time to take effect. For this reason, it is advisable to work to a fairly high precision when using the implicit method with rapid convergence, otherwise slow * propogation of the discontinuity occurs as with the explicit method.

It is found that the Crank-Nicolson method running times for the two iterative processes are about the same. This is not surprising, since the arithmetic involved in the direct iteration is much less than that required in the Newton-Raphson method, although the latter, being second order, converges much more quickly. The two effects roughly balance as far as time of running is concerned, but the Newton-Raphson method is much superior for large s.

The close agreement of theoretical and numerical results for the case p=-1, shows that the Crank-Nicolson method is very suitable for problems which contain no singular initial conditions. The fact that only a few points are taken to represent f(x,t) means that the computation in well-behaved cases need not be excessive in order to give a reasonable description of f(x,t).

An interesting possible method for the generation of initial approximations has not been dealt with here but is mentioned for completeness. The fundamental idea is to use the explicit method to start the DuFort-Frankel process to obtain approximations for f(x,t+ δ t). After f(x,t+ δ t) has been found by iteration it may be used, together with the initial conditions, to start the DuFort-Frankel method again. Subsequently, only the values of f(x,t) found by the Crank-Nicolson method need be used to give the next first approximation by insertion in the DuFort-Frankel formula. In this way, it may be possible to have the unrestricted stability of the DuFort-Frankel process without allowing inconsistency to develop.

Finally, we mention a problem which emerged in the investigation and which has not been dealt with. It was found when f(x,t) was negative and p was an odd integer, that the various methods available were

←
te "that"
mistake

unstable. A theoretical investigation of finite difference methods shows that all such methods require f^p to be positive. Thus, at present, there appears to be no finite difference technique for the solution of cases where f^p is negative.

APPENDIX 1

```
begin  comment This programme solves the eqn. by the Explicit method ;
integer m,n,a,b ;
real s,p ;
open(20) ; open(10) ;
a:=read(20) ; b:=read(20) ; s:=read(20) ; p:=read(20) ;
begin
real  array c[0:a,0:b+1] ;
for  m:=0  step  1  until  a  do
c[m,0]:=read(20) ;
for  n:=0  step  1  until  b do
for  m:=1  step  1  until  a  do
if  m<a  then
begin
c[0,n]:=c[0,0] ;
c[m,n+1]:=c[m,n]+c[m,n]↑p×s×(c[m+1,n]-2×c[m,n]+c[m-1,n]) ;
output(10,c[m,n+1])
end
else  if  m=a  then
begin  comment  this  deals  with  the derivative  boundary  condition  ;
c[0,n]:=c[0,0] ;
c[m,n+1]:=c[m,n]+c[m,n]↑p×s×2×(c[m-1,n]-c[m,n]) ;
output(10,c[m,n+1])
end ;
close(20) ; close(10)
end
end  end of programme
```

```
begin  comment This programme solves the eqn. by the Du Fort Frankel method
integer m,n,a,b ;
real s,p,x,y ;
open(20) ; open(10) ;
a:=read(20) ; b:=read(20) ; s:=read(20) ; p:=read(20) ; x:=read(20)
   y:=read(20) ;
begin
real array c[0:a,0:b+1] ;
for  m:=1  step  1  until  a  do
c[m,0]:=y ;  c[0,0]:=x ;
for  m:=0  step  1  until  a  do
c[m,1]:=read(20) ;
for  n:=1  step  1  until  b  do
begin
c[0,n]:=x ;
for  m:=1  step  1  until  a  do
begin
c[m,n+1]:=if  m<a  then
(2×s×c[m,n]↑p×(c[m+1,n]-c[m,n-1]+c[m-1,n])+c[m,n-1])/
                        (1+c[m,n]↑p×2×s)
else
(2×s×c[m,n]↑p×(2×c[m-1,n]-c[m,n-1])+c[m,n-1])/
                        (1+c[m,n]↑p×2×s) ;
output(10,c[m,n+1])
end  end ;
close(20) ; close(10)

end  end  end of programme
```

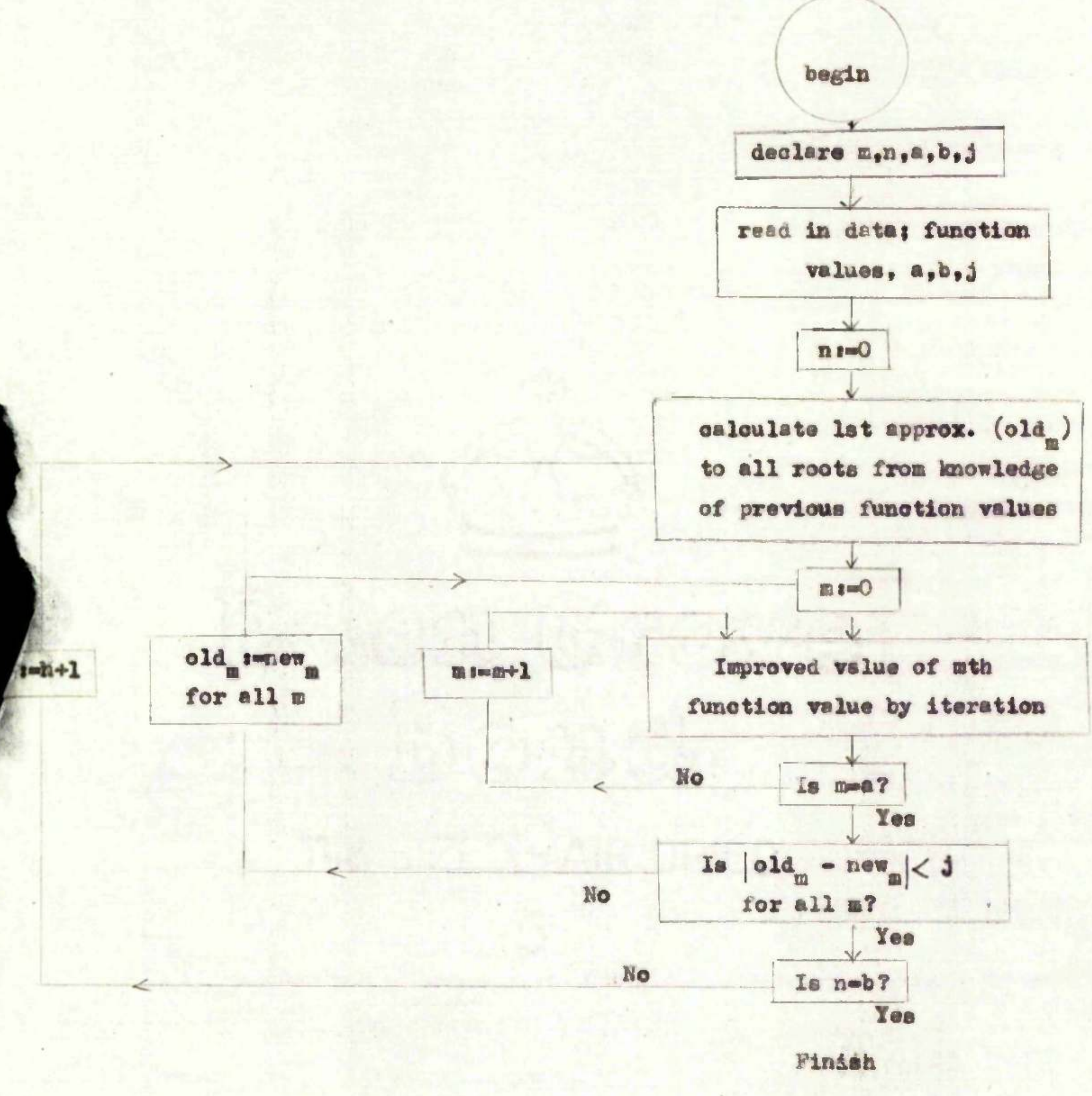
```
begin  comment This programme solves eqn by Crank-Nicolson method ;
integer m,n,a,b ;
real s,p,r,x,y ;
open(20) ; open(10) ;
a:=read(20) ; b:=read(20) ; s:=read(20) ; p:=read(20) ; r:=read(20)
   x:=read(20) ; y:=read(20) ;
begin
real array c1,c2[0:a,0:b+1] ;
for  m:=1  step  1  until  a  do
c1[m,0]:=y ;
c1[0,0]:=x ;
for  n:=0  step  1  until  b  do
begin
output(10,n) ;
for  m:=1  step  1  until  a  do
begin  comment initial guesses for Newton-Raphson iteration now found ;
c1[0,n+1]:=x ;
c1[m,n+1]:=if  m<a  then
c1[m,n]+c1[m,n]↑p×s×(c1[m+1,n]-2×c1[m,n]+c1[m-1,n])
else
c1[m,n]+c1[m,n]↑p×s×2×(c1[m-1,n]-c1[m,n])
end ; comment iterative process starts ;
```

```
1: for m:=1 step 1 until a do
c2[m,n+1]:=if m<a then
c1[m,n+1]-(2×r×s×c1[m,n+1]↑(p+1)
  -r×s×(c1[m+1,n+1]+c1[m-1,n+1])×c1[m,n+1]↑p+c1[m,n+1]
  -(c1[m,n]+s×(1-r)×c1[m,n]↑p×(c1[m+1,n]-2×c1[m,n]+c1[m-1,n])))/
  (2×r×s×(p+1)×c1[m,n+1]↑p
  -r×s×c1[m,n+1]↑(p-1)×(c1[m+1,n+1]+c1[m-1,n+1])+1)
else
c1[m,n+1]-(2×r×s×c1[m,n+1]↑(p+1)
  -2×r×s×c1[m-1,n+1]×c1[m,n+1]↑p+c1[m,n+1]
  -(c1[m,n]+s×(1-r)×2×c1[m,n]↑p×(c1[m-1,n]-c1[m,n])))/
  (2×r×s×(p+1)×c1[m,n+1]↑p-r×s×2×c1[m,n+1]↑(p-1)×c1[m-1,n+1]+1) ;
for m:=1 step 1 until a do
output(10,c2[m,n+1]) ;
for m:=1 step 1 until a do
if abs(c1[m,n+1]-c2[m,n+1]) >⏨-3 then
begin
for m:=1 step 1 until a do
c1[m,n+1]:=c2[m,n+1] ;
goto 1
end
end
end ;
close(20) ; close(10)
end  end of programme
```

```
begin  comment This programme solves eqn with p equal to minus one by
               the Crank-Nicolson method ;
integer m,n,a,b ;
real s,p,r;
open(20) ; open(10) ;
a:=read(20) ; b:=read(20) ; s:=read(20) ; p:=read(20) ; r:=read(20)
begin
real array c1,c2[0:a,0:b+1] ;
for  m:=0  step  1  until  a  do
c1[m,0]:=
0.5×sin(3.14159265/4+m/(a×2))/cos(3.14159265/4+m/(a×2))  ;
for  n:=0  step  1  until  b  do
begin  comment initial and boundary values calculated analytically ;
output(10,n) ;
c1[0,n+1]:=
0.5×sin(3.14159265/4+(n+1)×s/(a↑2))/
  cos(3.14159265/4+(n+1)×s/(a↑2))  ;
c1[a,n+1]:=
0.5×sin(3.14159265/4+1/2+(n+1)×s/(a↑2))/
  cos(3.14159265/4+1/2+(n+1)×s/(a↑2))  ;
for  m:=1  step  1  until  a-1  do
begin
c1[m,n+1]:=
c1[m,n]+1/c1[m,n]×s×(c1[m+1,n]-2×c1[m,n]+c1[m-1,n])
end ; comment Newton Raphson iteration now starts ;
```

```
1: for m:=1 step 1 until a-1 do
c2[m,n+1]:=
c1[m,n+1]-(2×α×s×c1[m,n+1]↑(p+1)
  -r×s×(c1[m+1,n+1]+c1[m-1,n+1])×1/c1[m,n+1]+c1[m,n+1]
  -(c1[m,n]+s×(1-r)×1/c1[m,n]×(c1[m+1,n]-2×c1[m,n]+c1[m-1,n])))/
  (2×α×s×(p+1)×1/c1[m,n+1]
  -r×s×1/c1[m,n+1]↑2×(c1[m+1,n+1]+c1[m-1,n+1])+1)    ;
for m:=1 step 1 until a-1 do
begin comment current iterated value and theoretical value printed together ;
output(10,c2[m,n+1]) ;
output(10,
0.5×sin(3.14159265/4+(n+1)×s/(a↑2)+m/(2×a))/
  cos(3.14159265/4+(n+1)×s/(a↑2)+m/(2×a)))
end ;
for m:=1 step 1 until a-1 do
if abs(c1[m,n+1]-c2[m,n+1]) >₁₀-3 then
begin
for m:=1 step 1 until a-1 do
c1[m,n+1]:=c2[m,n+1] ;
goto 1
end
end
end ;
close(20) ; close(10)
end end of programme
```

FLOW DIAGRAM FOR CRANK-NICOLSON METHOD



begin
declare m,n,a,b,j
read in data; function values, a,b,j
n:=0
calculate 1st approx. (old$_m$) to all roots from knowledge of previous function values
m:=0
:=n+1
old$_m$:=new$_m$ for all m
m:=m+1
Improved value of mth function value by iteration
No
Is m=a?
Yes
Is |old$_m$ - new$_m$|< j for all m?
No
Yes
No
Is n=b?
Yes
Finish

The boundary conditions are included in the blocks.

APPENDIX 2

All the results are for the initial and boundary conditions :

$$f(0,t) = 2.0$$

$$f(x,0) = 1.0 \quad \text{for } 0 \leq x \leq 1$$

except for the case p = -1, when the initial and boundary conditions are those described in the text. In all the Crank-Nicolson results the Newton-Raphson Iteration is carried out to a precision of 10^{-3}.

Explicit Method $p = 1.0$ $s = 0.24$

δx → δt ↓	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.240	1.000	1.000	1.000	1.000
2	2.000	1.395	1.058	1.000	1.000	1.000
3	2.000	1.484	1.129	1.014	1.000	1.000
4	2.000	1.541	1.194	1.038	1.003	1.000
5	2.000	1.582	1.249	1.068	1.011	1.002
6	2.000	1.614	1.295	1.100	1.023	1.006
7	2.000	1.640	1.334	1.131	1.038	1.014
8	2.000	1.661	1.367	1.161	1.055	1.026
9	2.000	1.679	1.396	1.189	1.074	1.040
10	2.000	1.694	1.421	1.215	1.095	1.057

Explicit Method $p = 2$ $s = 0.1$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.100	1.000	1.000	1.000	1.000
2	2.000	1.197	1.010	1.000	1.000	1.000
3	2.000	1.285	1.028	1.001	1.000	1.000
4	2.000	1.361	1.052	1.004	1.000	1.000
5	2.000	1.422	1.081	1.008	1.000	1.000
6	2.000	1.470	1.112	1.015	1.001	1.000

Explicit Method p = 3 s = 0.05

δx → δt ↓	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.050	1.000	1.000	1.000	1.000
2	2.000	1.102	1.003	1.000	1.000	1.000
3	2.000	1.156	1.007	1.000	1.000	1.000
4	2.000	1.209	1.015	1.000	1.000	1.000
5	2.000	1.262	1.024	1.001	1.000	1.000
6	2.000	1.312	1.036	1.002	1.000	1.000

Du Fort-Frankel Method $p = 1$ $s = 0.24$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.373	1.078	1.000	1.000	1.000
2	2.000	1.477	1.127	1.025	1.000	1.000
3	2.000	1.531	1.200	1.042	1.008	1.000
4	2.000	1.581	1.244	1.078	1.014	1.005
5	2.000	1.609	1.297	1.101	1.030	1.009
6	2.000	1.640	1.329	1.137	1.041	1.021
7	2.000	1.658	1.368	1.160	1.063	1.030
8	2.000	1.679	1.393	1.193	1.078	1.049
9	2.000	1.692	1.423	1.215	1.102	1.062
10	2.000	1.708	1.442	1.244	1.120	1.085

Du-Fort-Frankel Method   p = 1   s = 0.48

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.605	1.199	1.029	1.000	1.000
2	2.000	1.643	1.335	1.099	1.014	1.000
3	2.000	1.681	1.392	1.178	1.049	1.014
4	2.000	1.709	1.444	1.228	1.096	1.049
5	2.000	1.732	1.481	1.277	1.140	1.096
6	2.000	1.748	1.515	1.319	1.191	1.143
7	2.000	1.764	1.543	1.362	1.237	1.195
8	2.000	1.778	1.572	1.400	1.286	1.243
9	2.000	1.792	1.598	1.439	1.330	1.294
10	2.000	1.805	1.625	1.474	1.376	1.340

Du Fort-Frankel Method $p = 1$ $s = 0.96$

δx δt	0	1	2	3	4	5
0	2.000	1.000	1.000	1.000	1.000	1.000
1	2.000	1.940	1.460	1.175	1.046	1.014
2	2.000	1.815	1.703	1.324	1.123	1.061
3	2.000	1.803	1.628	1.517	1.246	1.159
4	2.000	1.813	1.637	1.492	1.426	1.316
5	2.000	1.827	1.665	1.539	1.477	1.542
6	2.000	1.844	1.707	1.611	1.595	1.557
7	2.000	1.868	1.760	1.709	1.638	1.622
8	2.000	1.901	1.833	1.746	1.702	1.680
9	2.000	1.944	1.859	1.799	1.753	1.743
10	2.000	1.946	1.893	1.839	1.809	1.792

Crank-Nicolson Method   Newton-Raphson Iteration

p = 1   s = 0.24   r = 0.75

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5	No. of Iterations required.
0	2.000	1.000	1.000	1.000	1.000	1.000	0
1	2.000	1.197	1.027	1.004	1.000	1.000	4
2	2.000	1.338	1.075	1.014	1.003	1.001	3
3	2.000	1.437	1.130	1.031	1.007	1.002	3
4	2.000	1.506	1.185	1.053	1.013	1.005	3
5	2.000	1.556	1.235	1.079	1.023	1.011	3
6	2.000	1.595	1.280	1.106	1.036	1.018	2
7	2.000	1.625	1.320	1.134	1.050	1.028	2
8	2.000	1.649	1.355	1.162	1.067	1.040	2
9	2.000	1.669	1.386	1.189	1.086	1.055	2
10	2.000	1.687	1.413	1.215	1.105	1.072	2

Crank-Nicolson Method   Newton-Raphson Iteration

$p = 1 \quad s = 0.48 \quad r = 0.75$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5	No. of Iterations required.
0	2.000	1.000	1.000	1.000	1.000	1.000	0
1	2.000	1.324	1.074	1.017	1.004	1.002	6
2	2.000	1.494	1.180	1.055	1.016	1.008	5
3	2.000	1.587	1.274	1.106	1.039	1.022	4
4	2.000	1.643	1.349	1.161	1.070	1.045	3
5	2.000	1.683	1.408	1.214	1.108	1.076	3
6	2.000	1.712	1.456	1.263	1.149	1.113	3

Crank-Nicolson Method Newton-Raphson Iteration

$p = 2 \quad s = 0.1 \quad r = 0.75$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5	No. of Iterations required.
0	2.000	1.000	1.000	1.000	1.000	1.000	0
1	2.000	1.098	1.006	1.000	1.000	1.000	2
2	2.000	1.191	1.020	1.001	1.000	1.000	2
3	2.000	1.274	1.040	1.004	1.000	1.000	3
4	2.000	1.345	1.064	1.008	1.001	1.000	3
5	2.000	1.404	1.092	1.014	1.002	1.000	3
6	2.000	1.452	1.121	1.022	1.003	1.001	3

Crank-Nicolson Method   Newton-Raphson Iteration

$p = 3 \quad s = 0.05 \quad r = 0.75$

δx → δt ↓	0	1	2	3	4	5	No. of Iterations required.
0	2.000	1.000	1.000	1.000	1.000	1.000	0
1	2.000	1.052	1.002	1.000	1.000	1.000	2
2	2.000	1.105	1.006	1.000	1.000	1.000	2
3	2.000	1.158	1.012	1.001	1.000	1.000	2
4	2.000	1.211	1.020	1.001	1.000	1.000	2
5	2.000	1.263	1.031	1.002	1.000	1.000	2
6	2.000	1.312	1.043	1.004	1.000	1.000	2

Crank-Nicolson Method **Newton-Raphson Iteration**

$p = -1.0 \quad s = 0.24 \quad r = 0.55$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5	No. of Iterations required.
0	0.583	0.675	0.787	0.927	1.110	1.362	0
1	0.589	0.682	0.795	0.938	1.125	1.384	1
2	0.595	0.689	0.804	0.949	1.140	1.406	1
3	0.601	0.696	0.813	0.961	1.156	1.428	1
4	0.607	0.704	0.822	0.973	1.172	1.451	1
5	0.613	0.711	0.832	0.985	1.188	1.475	1
6	0.619	0.719	0.841	0.997	1.205	1.499	1

Analytical Solution

$p = -1.0$ $s = 0.24$

$\delta x \rightarrow$ $\delta t \downarrow$	0	1	2	3	4	5
0	0.583	0.675	0.787	0.927	1.110	1.362
1	0.589	0.682	0.795	0.938	1.124	1.382
2	0.595	0.689	0.804	0.949	1.140	1.404
3	0.600	0.696	0.813	0.960	1.155	1.426
4	0.606	0.703	0.822	0.972	1.170	1.449
5	0.613	0.711	0.831	0.984	1.186	1.472
6	0.619	0.718	0.840	0.996	1.203	1.496

APPENDIX 3

REFERENCES

1. Fox, L. (Editor). Numerical Solutions of Ordinary and Partial Differential Equations. Oxford Summer School. Pergamon Press. London, 1962.

2. Hartree, D.R. Numerical Analysis. Second Edition. Clarendon Press. Oxford, 1964.

3. Householder, A.S. Principles of Numerical Analysis. McGraw-Hill Book Company Incorporated. 1953.

4. Milne, W.E. Numerical Solution of Differential Equations. John Wiley and Sons Incorporated. New York, 1953.

5. National Physical Laboratory. Modern Computing Methods. Second Edition. H.M.S.O., 1962.

6. Richtmyer, R.D. Difference Methods for Initial Value Problems. Interscience Publishers Limited. London, 1957.