1981

A Modified Crank-Nicolson Method

Ernest David Jordan Jr.

School of Arts and Sciences
Virginia Commonwealth University

This is to certify that the thesis prepared by Ernest David Jordan, Jr entitled A Modified Crank-Nicolson Method has been approved by his committee as satisfactory completion of the thesis requirement for the Master of Science degree in Mathematical Sciences.

[redacted]

Dr. Jerrold S. Rosenbaum
Director of Thesis

[redacted]

Dr. John R. Tucker
Committee Member

[redacted]

Dr. Robert H. Gowdy
Committee Member

[redacted]

Dr. John P. Mandeli
Chairman, Graduate Affairs Committee

[redacted]

Dr. William E. Haver
Department Chairman

5/15/81
Date

[redacted]

Dr. Elske P. Smith
Dean

A Modified Crank-Nicolson Method

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

Ernest David Jordan, Jr.

Director: Dr. Jerrold S. Rosenbaum

Assistant Professor of Mathematics

Virginia Commonwealth University

Richmond, Virginia

May, 1981

ACKNOWLEDGEMENTS

I wish to express my thanks and appreciation to Dr. Rosenbaum for his instruction and guidance throughout my graduate work. I thank my wife Linda for her loving patience throughout my education. I also wish to thank The Life Insurance Company of Virginia for giving me free use of its computer system.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

ODE	ordinary differential equation
PDE	partial differential equation
U(x,t)	U is a function of x and t
exp(x)	the exponential function e
$\partial U/\partial t$	the partial derivative of U with respect to t
dx/dt	the derivative of x with respect to t
f'	the first derivative of f with respect to its independent variable
f''	the second derivative of f
$\sum_{1}^{\infty} a_k$	summation of an infinite series $a_1 + a_2 + a_3 + \ldots$
$\int_{a}^{b} f(x)dx$	integral of f over the interval $[a,b]$
sin(x)	the sine of x
cos(x)	the cosine of x
Δx	mesh step size, $\Delta x = x_{k+1} - x_k$
n!	the factorial of n, n!= n(n-1)(n-2)...1
U_{ij}	$U(x_i,t_j)$
SCNM	the standard Crank-Nicolson method
SCNM2	the SCNM with space mesh 2h instead of h
ModCNM	the modified Crank-Nicolson method
r	the ratio k/(h*h)
det(A)	the determinant of A
diag($a_{11},\ldots,a_{nn}$)	the diagonal matrix with diagonal elements a_{11} through a_{nn}.

ABSTRACT

In order to obtain a numerical solution to the heat equation using finite differences, either implicit or explicit equations are used to formulate a solution. The advantage in an explicit formulation is its simplicity and minimal computer storage requirements while its disadvantage is its instability. The opposite is true for an implicit formulation such as the Crank-Nicolson method; although it is stable it is more difficult to implement and requires a much larger memory capacity. In this paper we examine the accuracy and stability of a hybrid approach, a modified Crank-Nicolson formulation, that combines the advantageous features of both the implicit and explicit formulations. This hybrid approach results in a 20% reduction in the amount of work required compared to the standard Crank-Nicolson solution if both methods use a special tridiagonal system solver. If Gaussian elimination is used, the modified Crank-Nicolson approach reduces the amount of work by 87%. Regardless of the linear system solver used, the modified Crank-Nicolson approach reduces by 50% the memory requirement of the standard Crank-Nicolson method.

CHAPTER I

INTRODUCTION

The partial differential equation which will be studied has the general form

$$aU_{xx} + bU_{xt} + cU_{tt} + dU_{x} + eU_{t} + fU + g = 0 \qquad (1)$$

where U is a function of two independent variables x and t. The subscript denotes differentiation with respect to that variable. The coefficients a, b, c, d, e, f, and g are also functions of x and t. As with ordinary differential equations (ODE), the problem is to find a solution function U(x,t) that satisfies the partial differential equation (PDE) described by (1). The solution function U(x,t) can sometimes be determined analytically but often one must resort to a numerical solution.

The order of a partial differential equation is the highest number of derivatives in a term of the equation. For example equation (1) is a second-order PDE because the highest number of derivatives in a single term is two. A partial differential equation is said to be linear if the coefficient functions depend only upon x and t. In this paper only the linear case will be considered. Also, all constants and functions are assumed to be real-valued.

A partial differential equation is said to be homogeneous if $g \equiv 0$. A special case of equation (1) is the following second-order linear homogeneous PDE called the "heat equation" :

$$U_{xx} - U_{t} = 0 \qquad (2)$$

This PDE governs the diffusion of heat in a thin solid rod of uniform density. For a derivation of the heat equation see references (6) or (9).

All second-order PDE's can be classified according to the coefficients a, b, and c as :

$b^2 - 4ac > 0$ hyperbolic

$b^2 - 4ac = 0$ parabolic

$b^2 - 4ac < 0$ elliptic .

Because the coefficients are functions of x and t, a given PDE may be elliptic in one region, hyperbolic in a second region, and parabolic in still a third region. In equation (2) the coefficients are constants so the heat equation is parabolic everywhere. The auxiliary conditions needed to find $U(x,t)$ differ markedly among the three; similarly the numerical methods among the classes also differ.

Before solving equation (2) some additional concepts and definitions must be developed.

An operator is a rule by which a function is associated with other functions. For example let $U(x,t) = \sin(\pi x)\exp(t^2)$ and define the operator $G = \partial/\partial t + \partial^2/\partial x^2$ and apply G to U :

$$GU = \partial U/\partial t + \partial^2 U/\partial x^2 = (2t - \pi^2)\sin(\pi x)\exp(t^2).$$

For the heat equation (2) the operator is $L = \partial^2/\partial x^2 - \partial/\partial t$ so the homogeneous form becomes $LU = \partial^2 U/\partial x^2 - \partial U/\partial t = 0$. An operator is said to be linear if for any n functions $F_1, F_2, \ldots, F_n$ and any n coefficients $C_1, C_2, \ldots, C_n$

$$L(C_1F_1 + \ldots + C_nF_n) = C_1L(F_1) + \ldots + C_nL(F_n) \quad .$$ See ref. (9).

The sum of n linear operators is another linear operator :

$$(L_1 + \ldots + L_n)(F) = L_1(F) + \ldots + L_n(F) \; .$$

Now define the linear operators $L_1(U) = \partial^2 U/\partial x^2$ and $L_2(U) = \partial U/\partial t$ so that the difference can be defined as the operator $L = L_1 - L_2$ which is linear also. This operator is the one previously defined for the heat equation in (2), hence it is also linear. The importance of linear operators is stated below in the two special cases of the Principle of Superposition :

> "If U_1, U_2,..., U_n are solutions of the homogeneous linear equation $Au = 0$, and if C_1, C_2,..., C_n are any constants, then $C_1U_1 + C_2U_2 + \ldots + C_nU_n$ is also a solution of the equation."
>
> "If U is a solution of $Au = f$ and V is a solution of $Av = 0$, then $W = U + V$ is a solution of $Aw = f$." See ref. (9).

As applied to non-homogeneous PDE's , one would first find all solutions of the homogeneous form and then form a linear combination with these solutions and the solution of the non-homogeneous case.

Auxiliary conditions are necessary to uniquely determine the solution to a PDE. Partial differential equations are similar to ordinary differential equations in this respect. An n-order ODE requires n auxiliary conditions to produce a unique solution from the family of solutions. The n conditions enable one to solve for the appropriate constants through relatively easy algebraic steps. By constrast, the n auxiliary conditions for PDE's are needed to solve for functions and not constants, thus the problem is infinitely more difficult. For the heat equation there are three derivatives so three auxiliary conditions are needed to uniquely determine $U(x,t)$. The three conditions are given by two boundary conditions and an initial condition. For the case of heat diffusion in a thin solid rod of uniform density, the boundary conditions specify the temperature at each end of the rod for any time t while the initial condition specifies the temperature distribution everywhere along the rod at some starting time t_0 .

To illustrate the preceding terms and definitions we state and solve the simplest form of the heat equation (which will be called the "basic heat problem").

Find $U(x,t)$ such that :

$$\text{(i)}\quad U_t = kU_{xx} \;,\quad 0 < x < 1 \text{ and } t > 0$$

$$\text{(ii)}\quad U(0,t) = U(1,t) = 0 \;,\quad t \geq 0 \text{ (boundary conditions)}$$

$$\text{(iii)}\quad U(x,0) = h(x) \;,\quad 0 < x < 1 \text{ (initial condition)}$$

The constant k is the coefficient of thermal conductivity associated with the material in the rod and the function $h(x)$ gives an initial temperature distribution along the rod. The classical solution to this problem begins by assuming that U has a particular form : $U(x,t) = k\, f(x)\, g(t)$. (2.5) Due to the assumed form of the solution, this method is called the Method of Separation of Variables . Differentiate (2.5) according to (i)

$$f\, g' = kg\, f'' \qquad \text{or}$$

$$g'/(kg) = f''/f \quad .$$

Because g is a function of t only and f depends only upon x, both sides must be a constant for any x or t, so call this constant $-c^2$. Continuing, there are two ordinary differential equations to solve :

$$g'/(kg) = -c^2 \quad\text{and}\quad f''/f = -c^2 \quad \text{which simplifies to}$$

$$g' = -kc^2 g \tag{3}$$

$$f'' = -c^2 f \quad . \tag{4}$$

To solve (4) requires two auxiliary conditions which are given by (ii) : $f(0) = f(1) = 0$. The general solution to (4) is $f(x) = A\cos(cx) + B\sin(cx)$. The first auxiliary condition requires $f(0) = A$ which implies $A = 0$ and the second requires $f(1) = B\sin(c) = 0$. The angles for which the sine is zero are $c = n\pi$ where $n = 1, 2, 3, \ldots$ thus there are an infinite number of solutions for $f(x)$:

$f_n(x) = B_n \sin(n\pi x)$ and the infinite sum is

$$f(x) = \sum_1^\infty B_n \sin(n\pi x) \quad .$$

Turning to g(t), an obvious general solution is

$g(t) = C \exp(-kc^2 t)$ where C is a constant to be determined from the auxiliary condition (iii). Having determined c from above one can substitute :

$$g_n(t) = C_n \exp(-kn^2\pi^2 t) \quad .$$

The solution becomes

$$U(x,t) = \sum_1^\infty A_n \exp(-kn^2\pi^2 t) \sin(n\pi x) \;, \quad (A_n = B_nC_n) \;.$$

The initial condition requires that

$U(x,0) = h(x) = \sum_1^\infty A_n \sin(n\pi x)$. This statement will be true if it can be shown that an arbitrary function h(x) can be represented by an infinite trigonometric series. In 1822 Joseph Fourier asserted that in the interval $(-\pi, \pi)$ an arbitrary function could be expressed by an infinite trigonometric series; he was able to prove his assertion for certain simple functions. It has subsequently been proven true under general conditions for a very general class of functions. (ref. 10). The coefficients A_n must satisfy the equations

$$A_n = 2 \int_0^1 h(x) \sin(n\pi x)dx \quad .$$

Thus the solution to the problem as posed is

$$U(x,t) = \sum_1^\infty \{(2 \int_0^1 h(x)\sin(n\pi x)dx) \exp(-kn^2\pi^2 t) \sin(n\pi x)\}.\text{(ref. 8)}$$

Obviously the solution will not be easy to evaluate at an arbitrary point (x,t) and that gives an important motivation to find a numerical solution. Also we have solved only one problem which is a relatively simple problem at that. Suppose the initial condition and boundary conditions were moderately "messy". The resulting attempt at solving the problem via separation of variables could be very difficult and perhaps impossible.

What are some other types of conditions ? There are three general classes :

(i) Dirichlet boundary conditions in which the boundary condition is a simple function of the form :

$$U(0,t) = F_1(t) \quad \text{and} \quad U(1,t) = F_2(t) \quad .$$

(ii) Neumann boundary conditions involving a derivative of U :

$$U_x(0,t) = F_1(t) \quad \text{and} \quad U_x(1,t) = F_2(t) \ .$$

(iii) Robin's boundary conditions which is a mixture of (i) and (ii) :

$$U(0,t) + kU_x(0,t) = F_1(t) \quad \text{and}$$
$$U(1,t) + cU_x(0,t) = F_2(t) \ . \text{ (ref. 9)}$$

The Dirichlet class has already been illustrated by the problem that was just solved. The Neumann conditions might arise in the case of a rod insulated at both ends so that the temperature loss in the direction of the x-axis is zero or near zero at both ends : $U_x(0,t) = U_x(1,t) = 0$. The Robin's conditions might apply in a situation where the ends are insulated (the derivative part of the condition) and there is some internal source generating the heat at the ends (the non-derivative part).

Thus far the x-axis has been restricted to the interval (0,1) instead of the more general interval (0,L). Likewise the temperature has been restricted to $[0,1]$ instead of $[t_0,t_1]$ where t_0 is the minimum temperature and t_1 is the maximum temperature in some domain. In order to use these restricted intervals a heat equation can be transformed via a change of variables so that (0,L) and $[t_0,t_1]$ are mapped to (0,1) and $[0,1]$ respectively. For example suppose one has the problem

$$U_t = kU_{xx} \text{ for } 0 < x < L , \quad t_0 \leq U(x,t) \leq t_1. \qquad (5)$$

Now introduce the two new variables z and w and the function V :

$$z = x/L \quad w = kt/L^2 \quad \text{and} \quad V = (U - t_0)/(t_1 - t_0) .$$

Differentiating one obtains

$$\frac{\partial V}{\partial z} = \frac{\partial}{\partial x}\left(\frac{U - t_0}{t_1 - t_0}\right)\frac{dx}{dz} = \left(\frac{L}{t_1-t_0}\right)U_x$$

$$\frac{\partial^2 V}{\partial z^2} = \left(\frac{L}{t_1-t_0}\right)\frac{\partial(U_x)}{\partial x}\frac{dx}{dz} = \left(\frac{L^2}{t_1-t_0}\right)U_{xx}$$

Substitute for U_{xx} in (5) :

$$U_t = \left(\frac{k(t_1-t_0)}{L^2}\right)V_{zz} \qquad (6)$$

Next find V_w :

$$\frac{\partial V}{\partial w} = \frac{\partial}{\partial t}\left(\frac{U - t_0}{t_1 - t_0}\right)\frac{dt}{dw} = \left(\frac{1}{t_1-t_0}\right) U_t \; (L^2/k)$$

Again substitute for U_t in (6) :

$$\left(\frac{k(t_1-t_0)}{L^2}\right)V_w = \left(\frac{k(t_1-t_0)}{L^2}\right)V_{zz}$$ which simplifies to equation (2) by replacing z, w, and V by x, t, and U respectively. Equation (5) is now said to be in dimensionless form. (ref. 1 and 3)

There are other transformations which can simplify the equation to be solved. In equation (1) the second term involves mixed partial derivatives. This term can be eliminated in a manner that is analogous to the technique through which the xy term in a general conic section (from analytic geometry) is eliminated through rotating the axes. Berg and McGregor state this fact in reference 9 :

> "By the introduction of new variables p, n, and w, every second-order equation $aU_{xx} + 2bU_{xy} + cU_{yy} + hU_x + kU_y + eU = f(x,y)$, where a, b, c, h, k, and e are constants, can be transformed into one and only one of the following standard

forms :

$$W_{pp} + W_{nn} + rW = Q(p,n) \tag{7}$$

$$W_{pp} + W_{nn} + rW = Q(p,n) \tag{8}$$

$$W_{pp} - W_{n} = Q(p,n) \tag{9}$$

$$W_{pp} + rW = Q(p,n) \tag{10}$$

where r is a constant with one of the values -1, 0, or 1."

From the classifications mentioned earlier (7) is elliptic, (8) is hyperbolic, (9) is parabolic, and (10) is degenerate.

Just as the boundary conditions can be given in a variety of forms, the heat equation itself can assume many different forms. Some of the more common varieties are given below (ref. 7 and 9) :

(i) $U_t = kU_{xx} + q(x,t,U)$

(ii) $U_t = kU_{xx} + r(x,t)U + q(x,t)$

(iii) $\partial U/\partial t = \partial(g(x)U_x)/\partial x$

Having introduced the basic concepts for partial differential equations and, in particular, for parabolic PDE's, what constitutes a "solvable" problem ? The three criteria that are given in references 1, 3, and 8 are :

(i) The solution must exist.

(ii) The solution must be unique.

(iii) The solution must depend continuously on the auxiliary conditions.

For the basic heat problem stated earlier, a solution was found in the form of an infinite series, hence a solution exists. There is, however, an important point to keep in mind. From a practical viewpoint we expect a solution for the physical problem to exist but the mathematical model for the physical problem may be in error and could have no solution. This fact applies equally to the uniqueness requirement. Nevertheless, let us assume that the basic heat problem accurately reflects the physical situation. The following theorem in Ames' text (ref. 3) guarantees uniqueness (D and B_t are the domains of x and t respectively) :

> "Given the initial boundary value problem
>
> $L(U) = g(x,t)U_{xx} - U_t = f(x,t,U,U_x)$ in D : $a < x < b$ and $B_t : 0 < t < T$ with $U(x,0) = h(x)$, if
>
> (i) $g(x,t)$ is bounded in $D + B_t$ and
>
> (ii) $f(x,t,U,U_x)$ is monotonic decreasing in U
>
> then there exists at most one solution. "

The last requirement (iii) for a "solvable" problem says that small changes in the input data should produce small changes in the solution, thus the system should be stable. That last condition is also the basis for the "repeatability" of an experiment; we expect experimental results to be nearly the same when very similar conditions are present for each experiment. For the numerical analyst, condition (iii) says that if small roundoff errors and truncation errors are introduced, the numerical solution will still be "close" to the analytic solution. Any problem that meets requirements (i), (ii), and (iii) is said to be <u>well-posed</u>.

Before examining some of the numerical methods for solving the heat equation it would be instructive to examine four sample problems and their solutions.

(i) Find $U(x,t)$ such that $U_t = U_{xx}$ subject to the initial condition $U(x,0) = \sin(\pi x)$, $0 < x < 1$, and the boundary conditions $U(0,t) = U(1,t) = 0$ for all $t \geq 0$. The analytic solution is $U(x,t) = \exp(-\pi^2 t)\sin(\pi x)$. (ref. 1)

(ii) Find $U(x,t)$ such that $U_t = U_{xx}$ subject to the initial condition $U(x,0) = \exp(x)$, $0 < x < 1$, and the boundary conditions $U(0,t) = \exp(t)$, $U(1,t) = \exp(t+1)$ for all $t \geq 0$.

The analytic solution is $U(x,t) = \exp(x)\exp(t)$.

The ease with which the solutions to examples (i) and (ii) can be evaluated makes them ideal test cases for investigating the accuracy and stability of a numerical method for solving the heat equation.

(iii) Find $U(x,t)$ such that $U_t = U_{xx}$ subject to the initial condition $U(x,0) = 1$ for $0 < x < 1$, and the boundary conditions $U(0,t) = U(1,t) = 0$ for all $t \geq 0$. The analytic solution is

$$U(x,t) = (4/\pi) \sum_0^\infty \exp(-(2n+1)^2\pi^2 t)\sin((2n+1)\pi x) /(2n+1) \quad .$$

(ref. 1)

(iv) Find $U(x,t)$ such that $U_t = U_{xx}$ subject to the initial condition

$$U(x,0) = \begin{cases} 2x & \text{for } 0 \leq x < .5 \\ 2(1-x) & \text{for } .5 \leq x \leq 1 \end{cases}$$

and the boundary conditions $U(0,t) = U(1,t) = 0$ for $t \geq 0$. The analytic solution is

$$U(x,t) = (8/\pi^2) \sum_1^\infty \{(\sin(n\pi/2)\sin(n\pi x)\exp(-n^2\pi^2 t))/n^2\} \ .$$

(ref. 1)

The last two examples illustrate the possible difficulty one can have in evaluating the analytic solution at a given point (x,t). Even though an analytic solution may be available (often that's not the case) one must still turn to numerical methods to evaluate the solution and even in the first two examples numerical methods are still needed to evaluate the exponential and trigonometric functions.

CHAPTER II

THE MODIFIED CRANK-NICOLSON METHOD

Numerical approximations

The second-order PDE's of the parabolic type are often described as marching problems due to the fact that a numerical solution is usually obtained by "fixing" the time position, i.e., a time row, and computing a solution at each space point, then advancing to the next row to compute a solution. For example, in figure 1 one would be given the initial conditions for time row t=0 and the boundary conditions for x=0 and x=1 for all $t \geq 0$ (i.e., values are known at those grid points that are boxed). The values for $k\Delta x$, $k = 1, 2, \ldots, n-1$ in time row $t = \Delta t$ could be calculated from the preceding values in time row t=0. Similarly the values for time row $t=2\Delta t$ could be calculated from some combination of the preceding time rows.

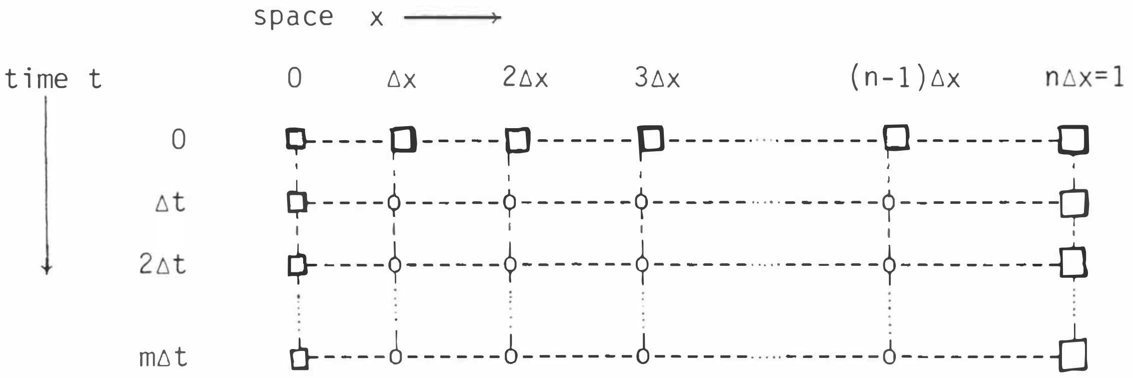


figure 1

Continuing in this fashion a temperature can be computed for any grid point. The methods for calculating the solution to a heat problem through a finite difference approach can be grouped into two types : implicit and explicit methods. As suggested by their name, *explicit* methods produce a solution at a point (x,t) by enabling one to solve for the unknown U(x,t) yielding a "formula" type of solution that gives the value for U(x,t) in terms of the known preceding values. On the other hand, *implicit* methods usually have many unknowns which can be solved via a linear system.

Due to their nature, explicit methods are generally easier to implement than implicit methods and give good results if Δx and Δt are appropriately chosen. Unfortunately the range of values for Δx and Δt necessary for stability and accuracy necessitates small steps in time. This implies a lot of computation for reasonable values of U, especially for large t. In turn, this large volume of computation introduces the potential for disastrous results through the introduction of round-off errors in the solution via the finite arithmetic used in all computers. If Δx and Δt are not chosen appropriately the round-off error can grow until it overwhelms the solution ! The amount of computation depends upon the method so an example is in order.

To approximate $U_{xx} = U_t$ a Taylor expansion can be used. Assuming t is held constant and U is sufficiently differentiable with respect to x, a useful expansion is

$$U(x+h) = U(x) + hU'(x) + h^2U''(x)/2 + h^3U'''(x)/6 + O(h^4) \quad . \quad (11)$$

Substituting -h for h gives :

$$U(x-h) = U(x) - hU'(x) + h^2U''(x)/2 - h^3U'''(x)/6 + O(h^4) \quad . \quad (12)$$

Assuming x is held constant and U is sufficiently differentiable with respect to t, a forward difference approximation to U'(t) can be obtained from (11) by substituting t for x and k for h :

$$U'(t) = (\ U(t+k) - U(t)\ )/k + O(k) \quad . \qquad (13)$$

Similarly, using (12) a backward difference approximation is

$$U'(t) = (\ U(t) - U(t-k)\ )/k + O(k) \quad . \qquad (14)$$

Adding (11) and (12) gives a central difference approximation for $U''(x)$:

$$U''(x) = (U(x+h) - 2U(x) + U(x-h))/h^2 + O(h^2) \quad . \qquad (15)$$

Now set $x_{i+1} - x_i = h$ and $t_{j+1} - t_j = k$ and $U_{ij} = U(x_i,t_j)$. Using equations (13) and (15) the basic heat equation (ignoring higher order terms) becomes :

$$U_{xx} - U_t = 0$$

$$(U_{i+1,j} - 2U_{ij} + U_{i-1,j})/h^2 - (U_{i,j+1} - U_{ij})/k = 0 \qquad (16)$$

Solving for $U_{i,j+1}$ gives the explicit scheme

$$U_{i,j+1} = (k/h^2)(U_{i+1,j} - 2U_{ij} + U_{i-1,j}) + U_{ij} \quad . \qquad (17)$$

Set $r = k/h^2$ to yield

$$U_{i,j+1} = rU_{i+1,j} - (1 - 2r)U_{ij} + rU_{i-1,j} \quad \text{or} \qquad (18a)$$

$$U_{i,j+1} - rU_{i+1,j} + (1 - 2r)U_{ij} - rU_{i-1,j} = 0 \quad . \qquad (18b)$$

Perhaps this result can be better illustrated by the computational molecule (see appendix C) in figure 2.

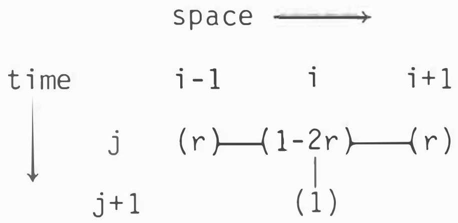


figure 2

Suppose the initial condition is $U(x,0) = \exp(x)$ for $0 < x < 1$ and the boundary conditions are $U(0,t) = \exp(t)$ and $U(1,t) = \exp(t+1)$ for $t \geq 0$. If we set $h = \Delta x = 0.2$ and $k = \Delta t = 0.01$ then $r = 0.25$ and the initial grid rounded to 4 decimal places is that shown in figure 3. Using the computational molecule in figure 4 the interior grid points are shown in figure 5. (The true errors are shown in parentheses.) If there are m internal space →

time ↓	0	.2	.4	.6	.8	1.0
0	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183
.01	1.0101					2.7456
.02	1.0202					2.7732
.03	1.0305					2.8011

figure 3

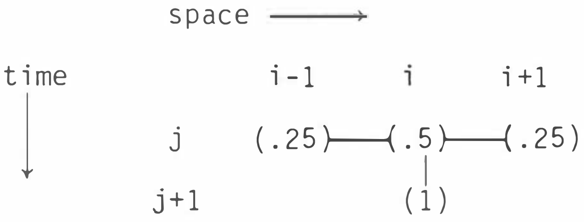


figure 4

space →

time ↓	0	.2	.4	.6	.8	1.0
0	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183
.01	1.0101	1.2337	1.5068	1.8404	2.2479	2.7456
.02	1.0202	1.2461	1.5219 (-.0001)	1.8589	2.2705	2.7732
.03	1.0305	1.2586	1.5372 (-.0001)	1.8776	2.2933	2.8011

figure 5

columns and n time rows ($t>0$) then the amount of work to produce a complete grid of values is 3mn multiplications and 2mn additions. For the sake of accuracy suppose we need $\Delta t = .001$ and $\Delta x = .1$ on the region $0 \leq x \leq 1$ and $t \geq 0$. Then there are nine internal columns and 1000 rows to reach $t = 1.0$ so the solution at $t = 1.0$ requires 27,000 multiplications and 18,000 additions. Obviously as the grid is refined to reduce the truncation error the number of operations increases dramatically. With a large number of operations, the round-off error due to machine limitations on storing decimal digits may become a serious factor. Also, because approximate equations are being used, their truncation error in conjunction with the machine round-off error may actually overwhelm the solution.

Both implicit and explicit methods result in a finite difference approximation to (2) and both are subject to several types of errors. Let U be the exact solution of (2) and let u be the exact solution of the finite difference equations. Then the discretization error is $U - u$. Next define the difference operator $D_{i,j+1}$ such that the finite difference equation (18b) becomes $D_{i,j+1}(u) = 0$. According to Smith (ref. 1), the local truncation error is $D_{i,j+1}(U)$. The finite difference equations are said to be consistent with (2) if

$$\lim_{\Delta x, \Delta t \to 0} D_{i,j+1}(U) = 0 \quad .$$

In reality the finite difference equations are not solved exactly due to the finite arithmetic performed on every computer. The difference between u and the computed solution c is called the round-off error :

$$u - c = \text{round-off error} \quad .$$

Suppose at some point, say (x_i, t_j), an error is introduced so that instead of u_{ij} the value is $u_{ij} + e_{ij}$. Assuming that all subsequent arithmetic is exact, a finite difference scheme is said to be stable if the error e_{ij} at all subsequent points decreases steadily.

As shown in (ref. 1) and (ref. 3), the explicit method just described in (18a) is stable for $0 < r \leq 0.5$ or $k \leq h^2/2$. To minimize the truncation error necessitates the use of small value for h and a correspondingly smaller value for k. Implicit methods permit a coarser mesh to be used while still yielding comparable accuracy. They are also less susceptible to fluctuations due to aberrations or discontinuities in the initial or boundary conditions.

The Crank-Nicolson Method

One of the main implicit methods is called the Crank-Nicolson method. This approach depends upon an average of two time rows in computing U_{xx} . More precisely, suppose we start with equation (16) but modify it to

$$\Big((U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1})/h^2 + (U_{i+1,j} - 2U_{ij} + U_{i-1,j})/h^2 \Big)/2 = (U_{i,j+1} - U_{ij})/k \ . \qquad (19)$$

Once again set $r = k/h^2$ and separate the "j+1" terms from the "j" terms :

$$-rU_{i-1,j+1} + (2+2r)U_{i,j+1} - rU_{i+1,j+1} = rU_{i-1,j} - (2-2r)U_{ij} + rU_{i+1,j} \ . \qquad (20)$$

To illustrate this finite difference equation the associated computational molecule is :

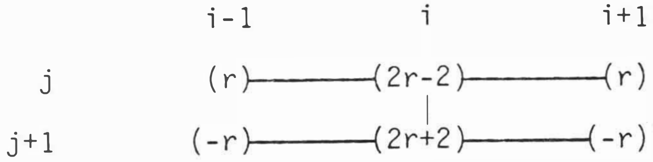


figure 6

A linear system is formed as i varies i = 1, 2,..., m (m internal columns) :

$$\begin{bmatrix} 2+2r & -r & & \\ -r & \ddots & \ddots & \\ & \ddots & \ddots & -r \\ & & -r & 2+2r \end{bmatrix} \cdot \begin{bmatrix} U_{1,j+1} \\ \vdots \\ \vdots \\ U_{m,j+1} \end{bmatrix} = \begin{bmatrix} 2r-2 & r & & \\ r & \ddots & \ddots & \\ & \ddots & \ddots & r \\ & & r & 2r-2 \end{bmatrix} \cdot \begin{bmatrix} U_{1,j} \\ \vdots \\ \vdots \\ U_{mj} \end{bmatrix} \qquad (21)$$

On the right-hand side the first and last equations need to be adjusted to correctly reflect the known boundary values at $U_{0,j+1}$, $U_{m+1,j+1}$, and $U_{0,j}$, $U_{m+1,j}$. The correct right-hand side should be :

$$\begin{bmatrix} 2r-2 & r & & \\ r & \ddots & \ddots & \\ & \ddots & \ddots & r \\ & & r & 2r-2 \end{bmatrix} \cdot \begin{bmatrix} U_{1,j} \\ \vdots \\ \vdots \\ U_{mj} \end{bmatrix} + r \begin{bmatrix} U_{0,j} \\ 0 \\ \vdots \\ 0 \\ U_{m+1,j} \end{bmatrix} + r \begin{bmatrix} U_{0,j+1} \\ 0 \\ \vdots \\ 0 \\ U_{m+1,j+1} \end{bmatrix}$$

(Number the corrected equation (22) and note that the last two vectors contain m-2 zeros.) The linear system created is a symmetric tridiagonal linear system which can be solved by a special method described in (ref. 7) which requires 3m-3 additions and multiplications and 2m-1 divisions for an m by m system. Suppose there are n time rows; there are n linear systems to be solved requiring a total of 3n(m-1) additions/multiplications and 2mn-n divisions. For a problem having 100 internal space points and 1000 time rows this amounts to 297,000 additions/multiplications and 199,000 divisions.

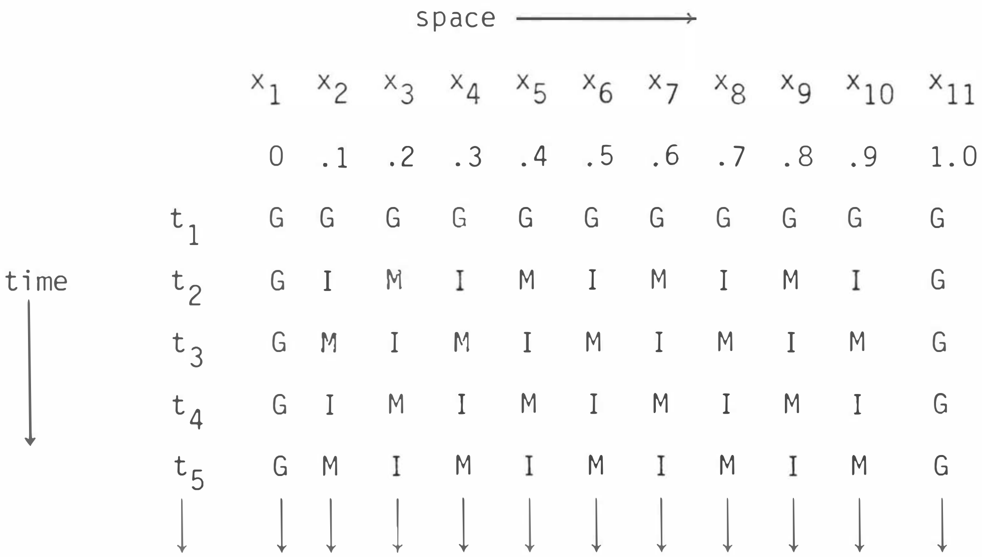
The Crank-Nicolson method is stable for all values of r>0, however r should be judiciously chosen to minimize the truncation error. This stability is demonstrated via the matrix approach to stability analysis in (ref. 1) and (ref. 3). The importance of stability is that it guarantees convergence. For parabolic partial differential equations this fact is stated in the Lax equivalence theorem, also found in (ref. 1) and (ref. 3) :

> "Given a properly posed initial boundary value problem and a finite difference approximation to it that satisfies the consistency condition, then stability is the necessary and sufficient condition for convergence. " (ref. 3)

The Modified Crank-Nicolson Method

The next implicit method is the main subject of research in this paper. The method is a modification of the Crank-Nicolson method and employs a mixture of explicit and implicit computations. The aim of this modified Crank-Nicolson method (ModCNM) is to reduce the number of computations while still maintaining the stability and accuracy of the standard Crank-Nicolson method (SCNM).

Before giving the derivation of the equations used in the ModCNM, a brief description would be helpful. To facilitate the description, refer to figure 7. The values for "G" grid points are obtained from the given initial conditions and the boundary value conditions. Beginning with row t_2 the ModCNM first computes the values for the "M" grid points via the SCNM (20). Since the SCNM is being applied to every other grid point, equation (20) must be modified by replacing h with 2h. To reflect this modification the abbreviation SCNM2 will be used. The values for the "I" grid points are then computed using some explicit interpolant. For the



	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
t_1	G	G	G	G	G	G	G	G	G	G	G
t_2	G	I	M	I	M	I	M	I	M	I	G
t_3	G	M	I	M	I	M	I	M	I	M	G
t_4	G	I	M	I	M	I	M	I	M	I	G
t_5	G	M	I	M	I	M	I	M	I	M	G

figure 7

next row, row t_3, the same computations are applied except that the values for the points in alternate columns are computed by the SCNM2 followed again by some interpolant for the remaining points in row t_3. The values for the remaining rows are similarly computed. A detailed derivation of the ModCNM equations will now be given.

To begin, the same notation will be used as was introduced earlier : $U_{ij} = U(x_i,t_j)$, $h = \Delta x$, $k = \Delta t$, and $r = k/h^2$. Also let the number of internal columns be 2n-1 and the number of rows be 2m+1 and start all subscript numbering with 1. Although the grid in figure 7 shows $\Delta x = .1$, for the ModCNM the value for h will actually be $2\Delta x$. Keep in mind that the ModCNM is applied to every other column so that equation (19) becomes :

$$((U_{i+2,j+1} - 2U_{i,j+1} + U_{i-2,j+1}) + (U_{i+2,j} - 2U_{ij} + U_{i-2,j}))/(8h^2) = (U_{i,j+1} - U_{ij})/k \qquad (23)$$

Further simplifying gives equations (24) and (25) :

$$r(U_{i+2,j+1} - 2U_{i,j+1} + U_{i-2,j+1} + U_{i+2,j} - 2U_{ij} + U_{i-2,j}) = 8U_{i,j+1} - 8U_{ij} \qquad (24)$$

$$-rU_{i+2,j+1} + (8+2r)U_{i,j+1} - rU_{i-2,j+1} = rU_{i+2,j} + (8-2r)U_{ij} + rU_{i-2,j} \qquad (25)$$

This last equation is applicable to the "M" points in time rows t_2, t_4, $t_6, \ldots, t_{2m}$ and as i varies, i = 2k-1 for k = 2, 3,..., n the linear system in (26) arises :

$$\begin{bmatrix} 8+2r & -r & & \\ -r & \ddots & \ddots & \\ & \ddots & \ddots & -r \\ & & -r & 8+2r \end{bmatrix} \cdot \begin{bmatrix} U_{3,j+1} \\ U_{5,j+1} \\ \vdots \\ U_{2n-1,j+1} \end{bmatrix} = \begin{bmatrix} 8-2r & r & & \\ r & \ddots & \ddots & \\ & \ddots & \ddots & r \\ & & r & 8-2r \end{bmatrix} \cdot \begin{bmatrix} U_{3,j} \\ U_{5,j} \\ \vdots \\ U_{2n-1,j} \end{bmatrix}$$

(Call this system (26).) Once again the right-hand side should be modified to reflect the known boundary values $U_{1,j}$, $U_{2n+1,j}$ and $U_{1,j+1}$, $U_{2n+1,j+1}$:

$$\begin{bmatrix} 8-2r & r & & \\ r & \ddots & \ddots & \\ & \ddots & \ddots & r \\ & & r & 8-2r \end{bmatrix} \cdot \begin{bmatrix} U_{3,j} \\ U_{5,j} \\ \vdots \\ U_{2n-1,j+1} \end{bmatrix} + r \begin{bmatrix} U_{1,j} \\ 0 \\ \vdots \\ 0 \\ U_{2n+1,j} \end{bmatrix} + r \begin{bmatrix} U_{1,j+1} \\ 0 \\ \vdots \\ 0 \\ U_{2n+1,j+1} \end{bmatrix} \quad (27)$$

(Note that the last two vectors contain n-3 zeroes.) The corrected system, system (27) requires the solution of an n-1 x n-1 linear system. The same system applies to the "M" grid points in time rows t_3, t_5, $t_7, \ldots,$ t_{2m+1} except that the approximation to U_{xx} at the first and last points must be modified due to the asymmetric space step at these points. To approximate U_{xx} one can rely upon the definition of f"(x) as the $\lim\limits_{h\to 0} (f'(x+h)-f(x))/h$.

For the point (x_2,t_3) in figure 7, a central difference approximation can be used for f'(x+h) in conjunction with a backward difference approximation for f'(x). The approximation for U_{xx} at (x_i,t_j) becomes

$$U_{xx} = (\ (U_{i+2,j} - U_{ij})/2h - (U_{ij} - U_{i-1,j})/h\ )/h \qquad (28)$$

or

$$U_{xx} = (U_{i+2,j} - 3U_{ij} + 2U_{i-1,j})/(2h^2) \quad . \qquad (29)$$

Substituting (29) into (23) as the approximation to U_{xx} in the ModCNM gives the modified linear system applicable to time rows t_3, t_5, $t_7, \ldots, t_{2m+1}$ (for details see appendix B) :

$$\begin{bmatrix} 4+3r & -r & & \\ -r & 8+2r & \ddots & \\ & \ddots & 8+2r & -r \\ & & -r & 4+3r \end{bmatrix} \cdot \begin{bmatrix} U_{2,j+2} \\ U_{4,j+2} \\ \vdots \\ U_{2n,j+2} \end{bmatrix} = \begin{bmatrix} 4-3r & r & & \\ r & 8-2r & \ddots & \\ & \ddots & 8-2r & r \\ & & r & 4-3r \end{bmatrix} \cdot \begin{bmatrix} U_{2,j+1} \\ U_{4,j+1} \\ \vdots \\ U_{2n,j+1} \end{bmatrix} + 2r \begin{bmatrix} U_{1,j+1} \\ 0 \\ \vdots \\ 0 \\ U_{2n+1,j+1} \end{bmatrix} + 2r \begin{bmatrix} U_{1,j+2} \\ 0 \\ \vdots \\ 0 \\ U_{2n+1,j+2} \end{bmatrix} \quad (30)$$

(Note that the last two vectors contain n-2 zeros.)

What can be used for the interpolant for each time row t_2, t_3, t_4, $t_5, \ldots, t_{2m+1}$? An explicit method similar to one already described will work nicely. Using equations (14) and (15) gives the explicit method in equation (31) :

$$U_{i,j+1} = (rU_{i-1,j+1} + U_{ij} + rU_{i+1,j+1})/(1 + 2r) \; . \qquad (31)$$

The computational molecule associated with (31) is shown in figure 8.

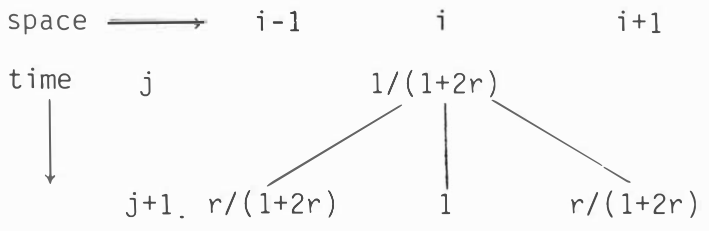


figure 8

By itself equation (31) is only stable for $0 < r \leq 0.5$ but used in conjunction with the more widely stable SCNM2 the two methods may also be stable. This stability will be addressed in a later chapter.

The ModCNM is described by the following algorithm : (32)

I. Conventions : Let the columns and rows be numbered as shown in figure 7 and let the number of columns be 2n+1 where n is some positive integer greater than 1. Similarly let the number of rows be 2m+1 where m is some positive integer greater than 1.

II. Steps : a. Initialize row 1 according to the initial conditions.

b. Initialize columns 1 and 2n+1 according to the boundary value conditions.

c. Apply system (27) to columns 3, 5, 7,..., 2n-1 in row two.

d. Apply equation (31) to columns 2, 4, 6,..., 2n in row two.

e. Apply system (30) to columns 2, 4, 6,..., 2n in row three.

f. Apply equation (31) to columns 3, 5, 7,..., 2n-1 in row three.

g. Repeat steps b - f to the succeeding rows until row 2m+1 is finished.

To illustrate this algorithm let's solve the sample problem (i) described at the end of chapter one. Set $\Delta x = \Delta t = 0.1$ so $r = 10$. Using the initial conditions and the boundary value conditions the time-space grid is that shown in figure 9. For this problem systems (27) and (30) are 4x4 and 5x5 systems respectively and are shown in equations (33) and (34) respectively. The interpolant, equation (31), becomes equation (35) and its computational molecule is that shown in figure 10. Applying the ModCNM algorithm described in (32) gives the results shown in figure 11. The reader is encouraged to verify these results.

space →

		0	.1	.2	.3	.4	.5
time ↓	0	0	.3090	.5878	.8090	.9511	1.0000
	.1	0					
	.2	0					
	⋮	⋮					
	1.0	0					

(The right half is symmetric to the left half.)

figure 9

$$\begin{bmatrix} 28 & -10 & 0 & 0 \\ -10 & 28 & -10 & 0 \\ 0 & -10 & 28 & -10 \\ 0 & 0 & -10 & 28 \end{bmatrix} \cdot \begin{bmatrix} U_{3,j+1} \\ U_{5,j+1} \\ U_{7,j+1} \\ U_{9,j+1} \end{bmatrix} = \begin{bmatrix} -12 & 10 & 0 & 0 \\ 10 & -12 & 10 & 0 \\ 0 & 10 & -12 & 10 \\ 0 & 0 & 10 & -12 \end{bmatrix} \cdot \begin{bmatrix} U_{3,j} \\ U_{5,j} \\ U_{7,j} \\ U_{9,j} \end{bmatrix}$$

equation 33

$$\begin{bmatrix} 34 & -10 & 0 & 0 & 0 \\ -10 & 28 & -10 & 0 & 0 \\ 0 & -10 & 28 & -10 & 0 \\ 0 & 0 & -10 & 28 & -10 \\ 0 & 0 & 0 & -10 & 34 \end{bmatrix} \cdot \begin{bmatrix} U_{2,j+2} \\ U_{4,j+2} \\ U_{6,j+2} \\ U_{8,j+2} \\ U_{10,j+2} \end{bmatrix} = \begin{bmatrix} -26 & 10 & 0 & 0 & 0 \\ 10 & -12 & 10 & 0 & 0 \\ 0 & 10 & -12 & 10 & 0 \\ 0 & 0 & 10 & -12 & 0 \\ 0 & 0 & 0 & 10 & -26 \end{bmatrix} \cdot \begin{bmatrix} U_{2,j+1} \\ U_{4,j+1} \\ U_{6,j+1} \\ U_{8,j+1} \\ U_{10,j+1} \end{bmatrix}$$

equation 34

$$U_{i,j+1} = (10U_{i-1,j+1} + U_{ij} + 10U_{i+1,j+1})/21$$

equation 35

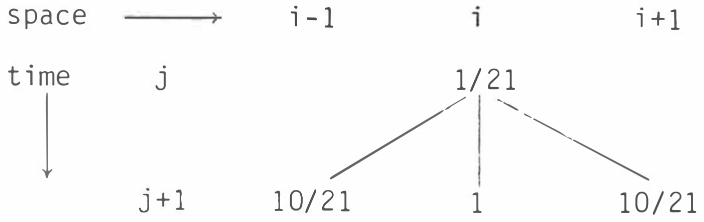


figure 10

space →

		0	.1	.2	.3	.4	.5
time ↓	0	0	.3090	.5878	.8090	.9511	1.0000
	.1	0	.0819	.1970	.2730	.3393	.3443
	.2	0	.0409	.0778	.1048	.1150	.1259
	.3	0	.0096	.0246	.0343	.0435	.0454
	.4	0	.0053	.0101	.0134	.0145	.0160
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	1.0	0	.1088	.2097	.2742	.2960	.3267

(Multiply the results in the last row by .0001)

figure 11

CHAPTER III

STABILITY ANALYSIS

The stability of the ModCNM will be examined using the matrix approach demonstrated in (ref. 1) and (ref. 3). The linear system in equation (27) can be written in matrix notation as

$$A\,U_{j+1} = B\,U_j + rV_j + rV_{j+1} \quad . \qquad (36)$$

Similarly equation (30) can be written as

$$\bar{A}\,\bar{U}_{j+2} = \bar{B}\,\bar{U}_{j+1} + 2r\bar{V}_{j+1} + 2r\bar{V}_{j+2} \quad . \qquad (37)$$

The interpolant for columns 2, 4,..., 2n in rows 2, 4,..., 2m given by equation (31) can be written as the system

$$\begin{bmatrix} U_{2,j+1} \\ U_{4,j+1} \\ \vdots \\ \\ U_{2n,j+1} \end{bmatrix} = \frac{r}{1+2r}\begin{bmatrix} U_{1,j+1} \\ 0 \\ \vdots \\ 0 \\ U_{2n+1,j+1} \end{bmatrix} + \frac{r}{1+2r}\begin{bmatrix} 1 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 1 \end{bmatrix} \cdot \begin{bmatrix} U_{3,j+1} \\ U_{5,j+1} \\ \vdots \\ \\ U_{2n-1,j+1} \end{bmatrix} + \frac{1}{1+2r}\begin{bmatrix} U_{2,j} \\ U_{4,j} \\ \vdots \\ \\ U_{2n,j} \end{bmatrix} \qquad (38)$$

and similarly the interpolant for columns 3, 5, 7,..., 2n-1 in rows 3, 5, 7,..., 2m+1 can be written as

$$\begin{bmatrix} U_{3,j+2} \\ U_{5,j+2} \\ \vdots \\ \\ U_{2n-1,j+2} \end{bmatrix} = \frac{r}{1+2r}\begin{bmatrix} 1 & 1 & & & 0 \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} U_{2,j+2} \\ U_{4,j+2} \\ \vdots \\ \\ U_{2n,j+2} \end{bmatrix} + \frac{1}{1+2r}\begin{bmatrix} U_{3,j+1} \\ U_{5,j+1} \\ \vdots \\ \\ U_{2n-1,j+1} \end{bmatrix} \quad . \qquad (39)$$

Using the matrix notation introduced in equations (36) and (37) gives the matrix equations for (38) and (39) as :

$$\overline{U}_{j+1} = a\overline{V}_{j+1} + aD\ U_{j+1} + b\overline{U}_j \qquad (40)$$

$$U_{j+2} = aD^T\overline{U}_{j+2} + bU_{j+1} \qquad (41)$$

where $a = r/(1+2r)$ and $b = 1/(1+2r)$. Thus the algorithm given in chapter two in (32) can be reduced to computing U_{j+1}, $\overline{U}_{j+1}$, U_{j+2}, $\overline{U}_{j+2}$ given by equations (36), (37), (40), and (41) respectively.

In order to determine the stability of the ModCNM, an iteration equation of the form $U_{j+1} = G\ U_j$ is desired. Extending the right-hand side gives $U_{j+1} = G\ U_j = G(G\ U_{j-1}) = G^2\ U_{j-1} = \ldots = G^{j+1}\ U_0$. (42)

Let U be the exact solution and u be the computed solution. Then the error e_{j+1} depends upon the matrix G and the initial error e_o as shown :

$$e_{j+1} = U_{j+1} - u_{j+1} = G^{j+1}U_0 - G^{j+1}u_0 = G^{j+1}(U_0 - u_0) = G^{j+1}e_0\ . \qquad (43)$$

If the eigenvalues p_i of G are distinct the initial error e_o can be expressed as a linear combination of the eigenvectors g_i of G :

$$e_0 = c_1g_1 + c_2g_2 + \ldots + c_ng_n \qquad \text{and}$$

$$e_{j+1} = G^{j+1}e_0 = c_1G^{j+1}g_1 + \ldots + c_nG^{j+1}g_n\ . \qquad (44)$$

By the definition of an eigenvalue and eigenvector $Gg_i = p_ig_i$ and equation (44) becomes

$$e_{j+1} = c_1{p_1}^{j+1}g_1 + c_2{p_2}^{j+1}g_2 + \ldots + c_n{p_n}^{j+1}g_n\ . \qquad (45)$$

If the modulus of the eigenvalues p_i are less than or equal to one then e_{j+1} will decay with each iteration making the iteration scheme stable.

Can the ModCNM be expressed in the desired form ? From equation (36) solve for U_{j+1} :

$$U_{j+1} = A^{-1}BU_j + rA^{-1}V_j + rA^{-1}V_{j+1} \qquad (46)$$

and substitute for U_{j+1} in (40) to give

$$\bar{U}_{j+1} = a\bar{V}_{j+1} + aDA^{-1}BU_j + arDA^{-1}V_j + arDA^{-1}V_{j+1} + b\bar{U}_j \quad . \qquad (47)$$

The linear systems for (46) and (47) can be written as the larger system :

$$\begin{bmatrix} U_{j+1} \\ \bar{U}_{j+1} \end{bmatrix} = \begin{bmatrix} A^{-1}B & 0 \\ aDA^{-1}B & bI \end{bmatrix} \cdot \begin{bmatrix} U_j \\ \bar{U}_j \end{bmatrix} + \begin{bmatrix} rA^{-1} & 0 \\ arDA^{-1} & 0 \end{bmatrix} \cdot \begin{bmatrix} V_j \\ \bar{V}_j \end{bmatrix} + \begin{bmatrix} rA^{-1} & 0 \\ arDA^{-1} & aI \end{bmatrix} \cdot \begin{bmatrix} V_{j+1} \\ \bar{V}_{j+1} \end{bmatrix} \qquad (48)$$

The solution along the j+1 row, system (48), can be written in matrix notation as

$$X_{j+1} = C_1X_j + C_2W_j + C_3W_{j+1} \quad . \qquad (49)$$

Similarly equation (37) can be solved for $\bar{U}_{j+2}$ and substituted into (41). The equations for U_{j+2} and $\bar{U}_{j+2}$ can be written as the system given in (50) and (51) :

$$\begin{bmatrix} U_{j+2} \\ \bar{U}_{j+2} \end{bmatrix} = \begin{bmatrix} bI & aD^T\bar{A}^{-1}\bar{B} \\ 0 & \bar{A}^{-1}\bar{B} \end{bmatrix} \cdot \begin{bmatrix} U_{j+1} \\ \bar{U}_{j+1} \end{bmatrix} + \begin{bmatrix} 0 & 2arD^T\bar{A}^{-1} \\ 0 & 2r\bar{A}^{-1} \end{bmatrix} \cdot \begin{bmatrix} V_{j+1} \\ \bar{V}_{j+1} \end{bmatrix} + \begin{bmatrix} 0 & 2arD^T\bar{A}^{-1} \\ 0 & 2r\bar{A}^{-1} \end{bmatrix} \cdot \begin{bmatrix} V_{j+2} \\ \bar{V}_{j+2} \end{bmatrix} \qquad (50)$$

$$X_{j+2} = \bar{C}_1X_{j+1} + \bar{C}_2W_{j+1} + \bar{C}_3W_{j+2} \qquad (51)$$

Together systems (49) and (51) represent the solutions along rows j+1 and j+2, that is, the solutions that result after a "complete" application of the ModCNM; the application is complete in the sense that every column has received an interpolated value and a standard Crank-Nicolson value. By combining systems (49) and (51), a complete application of the ModCNM can be embodied in the following linear system :

$$\begin{bmatrix} X_{j+1} \\ X_{j+2} \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & \bar{C}_1 \end{bmatrix} \cdot \begin{bmatrix} X_j \\ X_{j+1} \end{bmatrix} + \begin{bmatrix} C_2 & 0 \\ 0 & \bar{C}_2 \end{bmatrix} \cdot \begin{bmatrix} W_j \\ W_{j+1} \end{bmatrix} + \begin{bmatrix} C_3 & 0 \\ 0 & \bar{C}_3 \end{bmatrix} \cdot \begin{bmatrix} W_{j+1} \\ W_{j+2} \end{bmatrix} \qquad (52)$$

or
$$Z_{j+1} = M_1 Z_j + M_2 Y_j + M_3 Y_{j+1} \quad . \tag{53}$$

The ModCNM has been expressed as a matrix difference equation in the desired form (42) mentioned earlier in the chapter and the dependency of the j+1 application upon the initial application is

$$Z_{j+1} = M_1^{\,j+1} Z_0 + \sum_{k=0}^{j} M_1^{\,k} \left(M_2 Y_{j-k} + M_3 Y_{j-k+1} \right) \quad . \tag{54}$$

Thus the modified Crank-Nicolson method will be stable if the modulus of the eigenvalues of M_1 is less than or equal to unity.

What are the eigenvalues of M_1 ? From equations (48) and (49) comes the substitution represented by C_1 ,

$$C_1 = \begin{bmatrix} A^{-1}B & 0 \\ aDA^{-1}B & bI \end{bmatrix} \tag{55}$$

and from (50) and (51) the substitution represented by $\overline{C}_1$,

$$\overline{C}_1 = \begin{bmatrix} bI & aD^T\overline{A}^{-1}\overline{B} \\ 0 & \overline{A}^{-1}\overline{B} \end{bmatrix} \quad . \tag{56}$$

Using fact (i) from appendix A, the eigenvalues of M_1 are the eigenvalues of C_1 and $\overline{C}_1$, hence it will be necessary to find the eigenvalues of $A^{-1}B$ and $\overline{A}^{-1}\overline{B}$.

The product $A^{-1}B$ can be re-written using (27) in the form $(8I - rS)^{-1}(8I + rS)$ where

$$S = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \quad .$$

Matrix S is a symmetric tridiagonal matrix. From appendix A, fact (iii) gives the $k^{\underline{th}}$ eigenvalue p_k of $A^{-1}B$ as

$p_k = (8 - 2r(1 - \cos(k\pi/n))) \;/\; (8 + 2r(1 - \cos(k\pi/n)))$ and using the trigonometric identity $\sin^2(a/2) = (1 - \cos(a))/2$ yields the form

$$p_k = (8 - 4r\sin^2(k\pi/2n)) \;/\; (8 + 4r\sin^2(k\pi/2n)) \qquad (57)$$

for $k = 1, 2, \ldots, n-1$. The 2n-1 eigenvalues q_k of C_1 can be found using fact (i) in appendix A as

$$q_k = \begin{cases} p_k , & k = 1, \ldots, n-1 \\ b , & k = n, \ldots, 2n-1 \quad . \end{cases} \qquad (58)$$

Because $r > 0$ and $0 \leq \sin^2 a \leq 1$, the modulus of the eigenvalues p_k are less than one for all r and clearly $b \leq 1$ for all $r > 0$ since $b = 1/(1+2r)$, hence the modulus of all the eigenvalues of C_1 are less than unity.

To find the eigenvalues of $\bar{A}^{-1}\bar{B}$, the matrices $\bar{A}$ and $\bar{B}$ (equation 30) can expressed in a related form :

$$\bar{A} = D + rG \qquad \text{and} \qquad \bar{B} = D - rG \qquad (59)$$

where $D = \text{diag}(4,8,8,\ldots,8,8,4)$ and

$$G = \begin{bmatrix} 3 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 3 \end{bmatrix} .$$

This permits $\bar{A}$ and $\bar{B}$ to be factored as

$$\bar{A} = D(I + rD^{-1}G) \quad \text{and} \quad \bar{B} = D(I - rD^{-1}G) \quad \text{so that}$$

$$\bar{A}^{-1}\bar{B} = (I + rD^{-1}G)^{-1}D^{-1}D(I - rD^{-1}G) \quad \text{or}$$

$$\bar{A}^{-1}\bar{B} = (I + rD^{-1}G)(I - rD^{-1}G) \ .$$

The product $D^{-1}G$ can be computed directly as

$$D^{-1}G = \text{diag}(1/4,1/8,\ldots,1/8,1/4)\cdot G = \begin{bmatrix} 3/4 & -1/4 & & \\ -1/8 & 1/4 & -1/8 & \\ & \ddots\;-1/8 & \ddots\;1/4 & \ddots\;-1/8 \\ & & -1/4 & 3/4 \end{bmatrix}$$

By Smith in (ref. 1) (see fact (iii) in appendix A), the eigenvalues of the product $\bar{A}^{-1}\bar{B}$ are $(1-r\lambda)/(1+r\lambda)$ where λ is an eigenvalue of $D^{-1}G$. Given that $r > 0$, if $\lambda \geq 0$ then the modulus of the eigenvalues of $\bar{A}^{-1}\bar{B}$ will be less than or equal to one. Gerschgorin's circle theorem states that each eigenvalue of an n by n matrix A lies on or in the union of the circles $| \lambda - a_{ii}| = \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$. For the matrix $D^{-1}G$ the circles are $|\lambda - 3/4| \leq 1/4$ and $|\lambda - 1/4| \leq 1/4$ or $1/2 \leq \lambda \leq 1$ and $0 \leq \lambda \leq 1/2$ so each eigenvalue of $D^{-1}G$ is non-negative. Hence the modulus of the eigenvalues of $\bar{A}^{-1}\bar{B}$, $\bar{C}_1$, and M_1 are less than or equal to one, and so the modified Crank-Nicolson method is stable.

CHAPTER IV

EXPERIMENTAL RESULTS

A perturbational analysis was used to experimentally examine the stability of the ModCNM. This technique may be used to demonstrate the manner in which a rounding error is propagated with each successive application of the ModCNM. For the perturbational analysis, zero boundary conditions were used as well as zero initial conditions with the exception U(.5,0) was set to one to represent a rounding error.

To examine the effect of mesh length variation upon the decrease in the initial perturbation, the mesh lengths in the space and time dimensions were varied independently. The results were plotted for four cases :

(i) At the points (.5,t), t = .1, .2,..., 1.0, fix h and vary k (figure 12).

(ii) At the points (.5,t), t = .1, .2,..., 1.0, fix k and vary h (figure 13).

(iii) At the point (.5,1.0) fix h and vary k (figure 14).

(iv) At the point (.5,1.0) fix k and vary h (figure 15).

The truncation error in the approximation (13) may be minimized by decreasing k but, as illustrated in figure 12, there is a corresponding decrease in stability. From figure 13 the greatest stability occurs when the truncation error to U_{xx} is minimized, however this rule cannot be rigidly applied. The line h = 0.025 has a slightly larger truncation error than h = 0.02 but the corresponding gain in stability may be worth the larger truncation error.

Figures 14 and 15 indicate that certain mesh sizes are to be avoided if stability and reasonable accuracy are desired. Of the mesh lengths that were tested, the optimum space-mesh length appears to be $h = 0.025$. For this value of h the decrease in the initial perturbation is second only to $h = 0.01$ which requires approximately twice as much work; this can be seen in figures 13, 14, 15, and table 3.

To experimentally examine the accuracy of the ModCNM, the four problems at the end of chapter one were tested. The relative error was computed as the (true value - computed value)/(true value). Some of the results are tabulated in tables 1, 2, and 3.

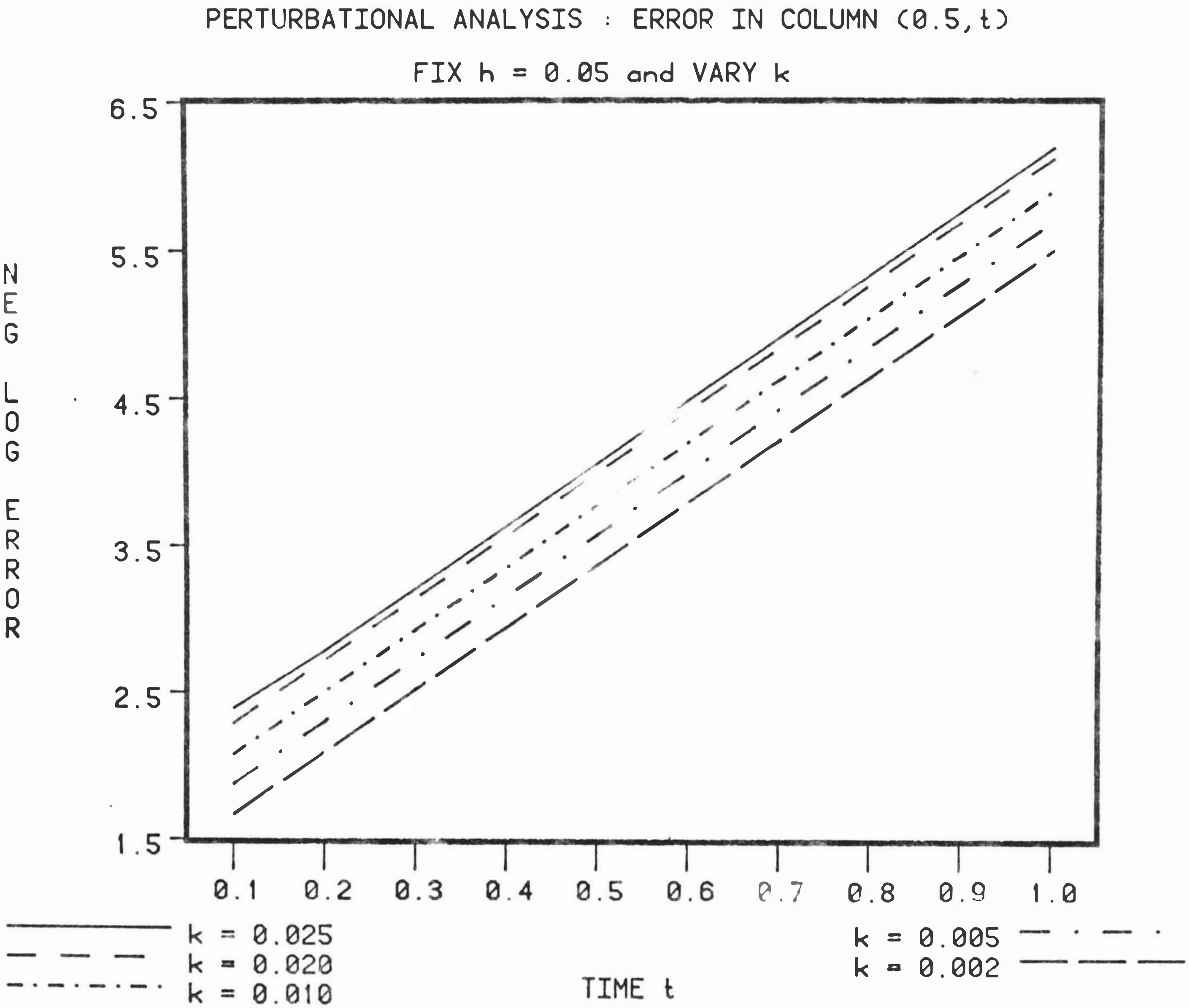

PERTURBATIONAL ANALYSIS : ERROR IN COLUMN (0.5,t)
FIX h = 0.05 and VARY k
NEG LOG ERROR
6.5
5.5
4.5
3.5
2.5
1.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
TIME t
k = 0.025
k = 0.020
k = 0.010
k = 0.005
k = 0.002

figure 12

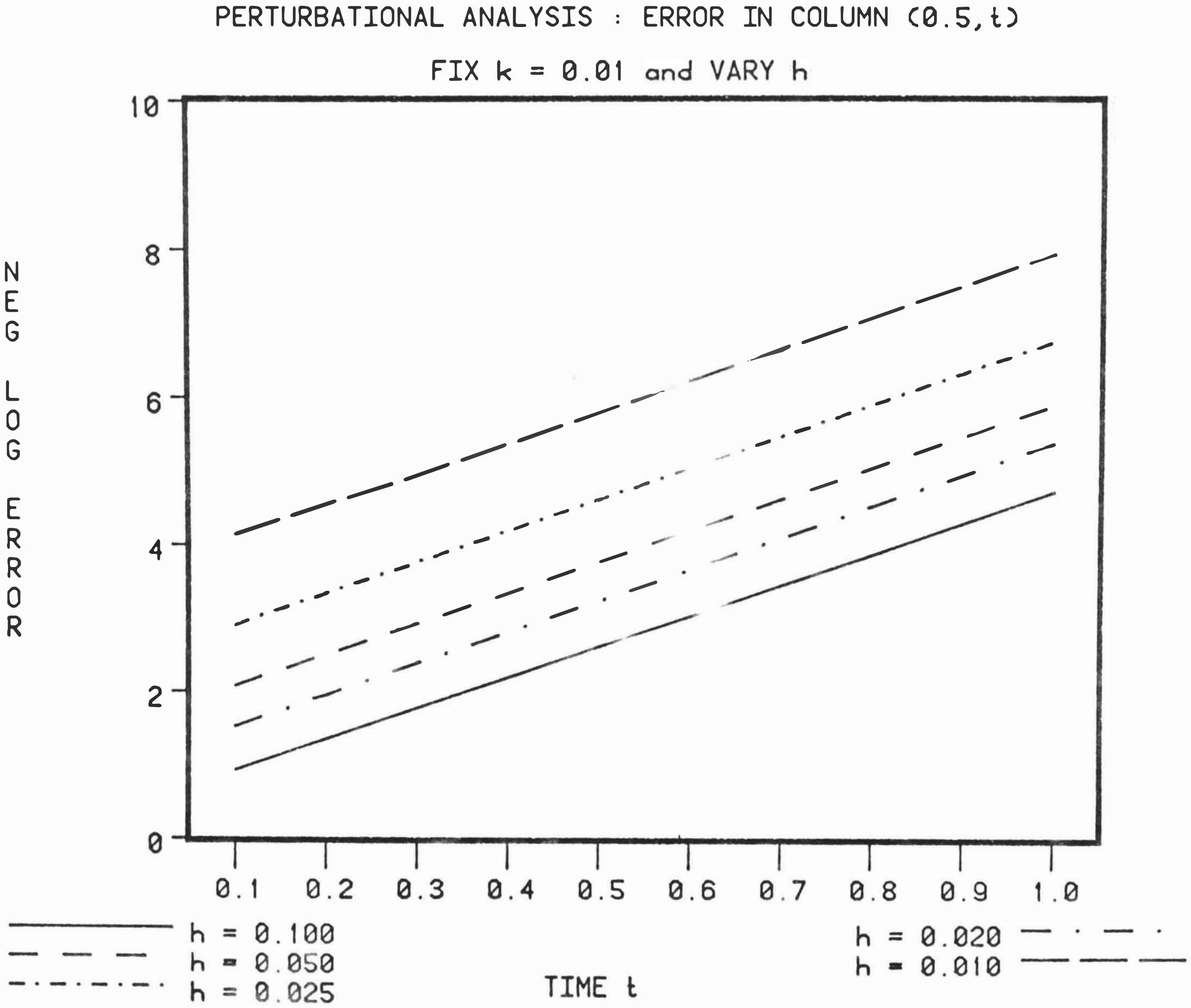

PERTURBATIONAL ANALYSIS : ERROR IN COLUMN (0.5,t)
FIX k = 0.01 and VARY h
NEG LOG ERROR
10
8
6
4
2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
TIME t
h = 0.100
h = 0.050
h = 0.025
h = 0.020
h = 0.010

figure 13

PERTURBATIONAL ANALYSIS : ERROR AT POINT (0.5,1.0)

FIX h = .1, .05, .025, .02, .01 and VARY k

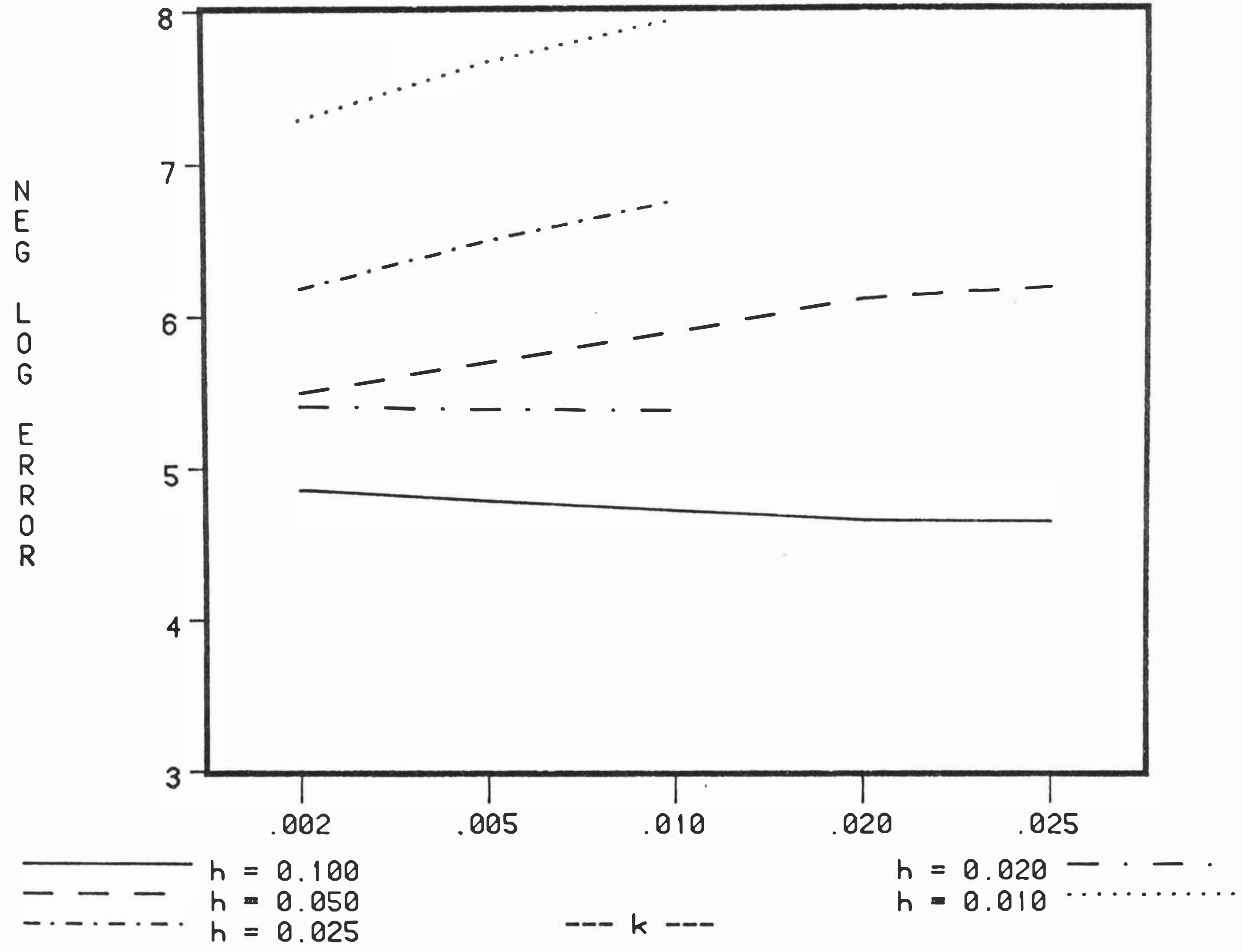


figure 14

PERTURBATIONAL ANALYSIS : ERROR AT POINT (0.5,1.0)

FIX k = 0.010, 0.002 and VARY h

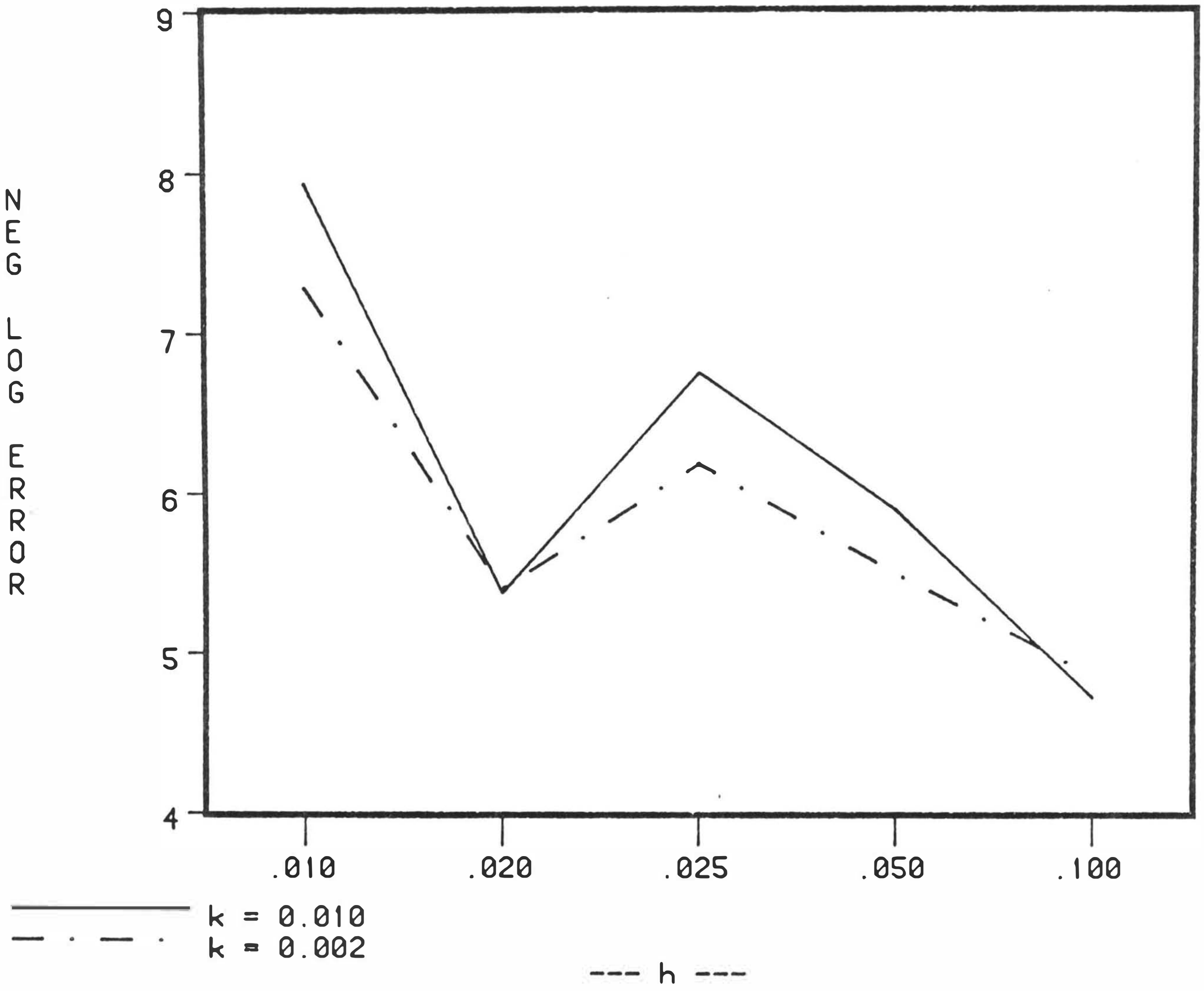


figure 15

Relative Error x 100 at (.5,t)
h = 0.025 and k = 0.02

Problem Number

t	1	2	3	4
.1	-.0161	-.0165	.929	.4998
.2	.0072	-.0149	1.334	.0444
.3	-.0367	-.0225	1.303	.0548
.4	-.0156	-.0203	1.323	.0703
.5	-.0556	-.0184	1.281	.0292
.6	-.0336	-.0200	1.304	.0506
.7	-.0751	-.0211	1.258	.0111
.8	-.0510	-.0191	1.285	.0331
.9	-.0937	-.0197	1.245	-.0089
1.0	-.0735	-.0201	1.265	.0119

table 1

Relative Error x 100 at (.5,1.0)
h = 0.05

Problem Number

k	1	2	3	4
.025	7.691	-.0803	-3.054	-6.216
.020	-8.068	-.0803	-4.993	-7.809
.010	-9.534	-.0803	-7.249	-9.314
.005	-8.818	-.0781	-7.087	-8.680
.002	-6.978	-.0714	-5.717	-6.974

table 2

Relative Error x 100 at (.5,1.0)
k = 0.02

Problem Number

h	1	2	3	4
.100	-25.66	-.3035	-18.88	-29.64
.050	-8.068	-.0803	-4.993	-7.809
.025	-.0735	-.0201	1.265	.0119
.020	1.027	-.0134	2.082	.8921
.010	2.618	-.0022	3.298	2.068

table 3

CHAPTER V

FUTURE CONSIDERATIONS

To compare the amount of work required by the ModCNM and the SCNM, these assumptions will be used : (i) There are 2n-1 internal mesh points in each row; (ii) There are 2m rows beyond the initial row; (iii) Gaussian elimination requires approximately $k^3/3$ operations to solve a large k by k system; (iv) The tridiagonal system solver requires 3k additions/multiplications and 2k divisions which will be counted jointly as 5k operations; (v) The interpolant given by equation (31) requires two additions/multiplications and one division which combines for a count of three operations per interpolated value.

Using Gaussian elimination the number of operations required by the ModCNM is $m(n^3/3 + 3(n-1) + (n-1)^3/3 + 3n)$ which is approximately $2mn^3/3$ operations for large n. The SCNM requires $2m(2n-1)^3/3$ operations, or approximately $16mn^3/3$ operations for large n. The ModCNM requires 87% fewer operations.

Using the tridiagonal solver, the number of operations required by the ModCNM and the SCNM are respectively $m(5n + 3(n-1) + 5(n-1) + 3n)$ which is approximately 16mn and $2m(5(2n-1))$ which is approximately 20mn, hence the ModCNM requires 20% fewer operations.

The modified Crank-Nicolson method can be used effectively to solve the heat equation with reasonable accuracy. The savings in computation makes it an attractive alternative to the SCNM. Although both methods result in a tridiagonal matrix structure which may be stored in three vectors, the lengths of the vectors in the ModCNM are half the lengths required by the SCNM so there is a 50% savings in storage requirements.

To be used effectively careful consideration must be given to the choice of mesh lengths. Further research is needed to determine the optimum mesh lengths that minimize the truncation error and the amount of work while maximizing stability.

In this thesis the ModCNM has been developed only for the one-dimensional problem. The following questions spotlight areas for future study. How should the ModCNM be extended to handle the two- and three-dimensional heat equation ? What should be the pattern of alternating implicit/explicit computations in these higher dimensions ?

Suppose the region over which the heat equation is to be solved is irregular. How should the ModCNM be altered ? How can the ModCNM be adapted to non-rectangular coordinates ? Can the idea behind the ModCNM (using alternating implicit/explicit computations) be adapted for use by other PDE methods ?

APPENDIX A

FACTS AND THEOREMS FROM LINEAR ALGEBRA

1. Given the square matrix S that has been partitioned as shown where A and D are also square matrices, then det(S) = det(A)det(D) and the eigenvalues of S are the eigenvalues of A and D.

$$S = \begin{bmatrix} A & B \\ 0 & D \end{bmatrix}$$

2. The eigenvalues of an n by n tridiagonal matrix of the form below are $p_k = a + 2\sqrt{bc}\cos(k\pi/(n+1))$, $k = 1,\ldots, n$. See Smith (ref. 1).

$$\begin{bmatrix} a & b & & \\ c & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & c & a \end{bmatrix}$$

3. If $(A + hI)$ and $(A + kI)$ are two matrices and if p is an eigenvalue of A, then $(p + h)/(p + k)$ is an eigenvalue of $(A + kI)^{-1}(A + hI)$. See Smith (ref. 1).

APPENDIX B

DETAILS FOR THE DERIVATION OF SYSTEM (30)

Equation (29) represents the approximation to U_{xx} at the points (x_2,t_{2j+1}) , $j = 1, 2, \ldots,$ thus (29) could be written as

$$U_{xx} = (U_{4,2j+1} - 3U_{2,2j+1} + 2U_{1,2j+1}) / (2h^2) \quad .$$

The approximation to U_t is $U_t = (U_{2,2j+1} - U_{2,2j}) / k$. The ModCNM approximation to the heat equation at (x_2,t_{2j+1}) is $(z_1 + z_2) / 2 = z_3$ where

$$z_1 = (U_{4,2j+1} - 3U_{2,2j+1} + 2U_{1,2j+1}) / (2h^2) \ , \quad \text{and}$$

$$z_2 = (U_{4,2j} - 3U_{2,2j} + 2U_{1,2j}) / (2h^2) \ , \text{ and}$$

$z_3 = (U_{2,2j+1} - U_{2,2j}) / k$. Separating the "2j+1" terms from the "2j" terms and simplifying gives $z_4 = z_5$ where

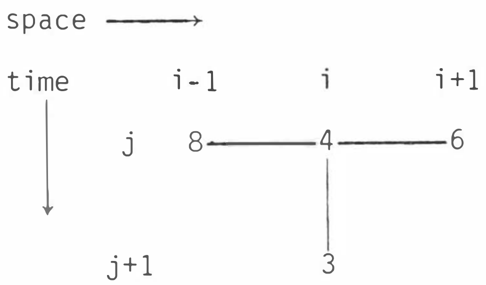
$$z_4 = -2rU_{1,2j+1} + (4 + 3r)U_{2,2j+1} - rU_{4,2j+1} \quad , \quad \text{and}$$

$z_5 = 2rU_{1,2j} + (4 - 3r)U_{2,2j} + rU_{4,2j}$ which is the equation in the first row of system (30). Because the boundary values are known, $U_{1,2j+1}$ and $U_{2n+1,2j+1}$ should be moved to the right-hand side as shown in (30). Due to symmetry, a similar derivation applies to the approximation at (x_{2n},t_{2j+1}) .

APPENDIX C

COMPUTATIONAL MOLECULES

A computational molecule is also referred to as a star, a stencil, or a lozenge. Its purpose is to illustrate the course of a computation. For example, suppose the star below was applied to U(x,t). Using the notation



$U(x_i,t_j) = U_{ij}$, the star represents the finite difference equation

$$3U_{i,j+1} = 8U_{i-1,j} + 4U_{ij} + 6U_{i+1,j} \quad .$$

LIST OF REFERENCES

LIST OF REFERENCES

1. Smith, Gordon D., Numerical Solution of Partial Differential Equations, 2nd ed., Oxford University Press, Oxford, England, 1978

2. Greenspan, Donald, Lectures on the Numerical Solution of Linear, Singular, and Nonlinear Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1968

3. Ames, William F., Numerical Methods for Partial Differential Equations, Barnes & Noble, Inc., New York, N.Y., 1969

4. Spiegel, Murray R., Applied Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967

5. Friedman, A., Partial Differential Equations of the Parabolic Type, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1964

6. O'Neil, Peter V., Advanced Calculus, Macmillan Publishing Co., Inc., New York, N.Y., 1975

7. Dahlquist, Germund and Bjorck, Ake, Numerical Methods, Prentice-Hall, Inc., Englewood, N.J., 1974

8. Myint-U, Tyn, Partial Differential Equations of Mathematical Physics, Elsevier North-Holland, Inc., New York, N.Y., 1980

9. Berg, Paul W., and McGregor, James L., Elementary Partial Differential Equations, Holden-Day, Inc., San Francisco, Cal., 1966

10. Carslaw, H. S., An Introduction to the Theory of Fourier's Series and Integrals, Dover Publications, Inc., New York, N.Y., 1950

VITA

[redacted]