1991

Difference schemes for the convective-diffusive equation

William Grant Rolston
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Rolston, William Grant, "Difference schemes for the convective-diffusive equation" (1991). *Master's Theses*. 160.

DOI: <https://doi.org/10.31979/etd.anb6-wgwh>
https://scholarworks.sjsu.edu/etd_theses/160

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1344306

Difference schemes for the Convective-Diffusive Equation

Rolston, William Grant, M.S.

San Jose State University, 1991

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**

DIFFERENCE SCHEMES FOR THE CONVECTIVE-DIFFUSIVE EQUATION

A thesis

Presented to

The Faculty of the Department of Physics

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

William Grant Rolston

May, 1991

APPROVED FOR THE DEPARTMENT OF PHYSICS

A handwritten signature in cursive script, reading "Patrick Hamill", written over a horizontal line.

Dr. Patrick Hamill

A handwritten signature in cursive script, reading "Yue-Kwen Kwok", written over a horizontal line.

Dr. Yue - Kwen Kwok

A handwritten signature in cursive script, reading "Alejandro Garcia", written over a horizontal line.

Dr. Alejandro Garcia

APPROVED FOR THE UNIVERSITY

A handwritten signature in cursive script, reading "Serena H. Stanford", written over a horizontal line.

ABSTRACT

DIFFERENCE SCHEMES FOR THE CONVECTIVE-DIFFUSIVE EQUATION

by William Rolston

This thesis develops new schemes to approximate the Convective-Diffusive Equation, using the Modified Equation Approach. The stabilities of the new schemes are examined by the von Neumann and Matrix method. The diffusion and accuracy of a selection of new schemes is also presented, along with the stability and accuracy of several well-known previous schemes for the Convective-Diffusive Equation.

The use of the Modified Equation Approach allows the generation of any number of new schemes. All schemes considered are of the two-level, six-point form. The stability requirements for the general two-level, six-point case are considered, allowing the development of an infinite number of second-order accurate schemes which are unconditionally von Neumann stable. A family of third-order, von Neumann stable schemes is also presented, along with a conditionally stable fourth-order scheme.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
INTRODUCTION	1
GENERATION OF SCHEMES	3
STABILITY	13
DIFFUSION	22
NUMERICAL TESTS	24
CONCLUSION	35
BIBLIOGRAPHY	37
TABLE 1	38
FIGURES	44
APPENDIX	60

LIST OF TABLES

Table	Page
1. Modified Equivalent Partial Differential Equation for the Convective -Diffusive Equation	38

LIST OF FIGURES

Figure	Page
1. von Neumann stability of the general second-order implicit scheme. The shaded regions indicate the regions of stability. . . .	44
2. Stability of the FTCS scheme. The solid line indicates numerically determined stability. The dashed line indicates the von Neumann stability. The dotted line indicates the matrix method stability.	45
3. Stability of the Lax-Wendroff scheme. The solid line indicates numerically determined stability. The dashed line indicates the von Neumann stability. The dotted line indicates the matrix method stability.	46
4. Stability of the FTCS family, $p = .2$ scheme. The solid line indicates numerically determined stability. The dashed line indicates von Neumann stability. The dotted line indicates the matrix method stability.	47
5. Dissipation curve for the Crank-Nicholson scheme. Γ is set to 1. .	48
6. Dispersion curve for the Crank-Nicholson scheme. Γ is set to 1. .	49
7. Dissipation curve for the 2A scheme. Γ is set to 1.	50
8. Dispersion curve for the 2A scheme. Γ is set to 1.	51
9. Dissipation curve for the 2B scheme. Γ is set to 1.	52
10. Dispersion curve for the 2B scheme. Γ is set to 1.	53
11. Dissipation curve for the 2C scheme. Γ is set to 1.	54
12. Dispersion curve for the 2C scheme. Γ is set to 1.	55
13. Dissipation curve for the 3D scheme. Γ is set to 1.	56
14. Dispersion curve for the 3D scheme. Γ is set to 1.	57
15. Dissipation curve for the 4E scheme. Γ is set to 1.	58

16. Dispersion curve for the 4E scheme. Γ is set to 1.	59
---	----

INTRODUCTION

The primary goal of this project has been to develop new difference schemes to approximate the convection-diffusion equation:

$$\frac{\partial T}{\partial t} + \omega \frac{\partial T}{\partial x} = \gamma \frac{\partial^2 T}{\partial x^2} . \quad (1)$$

The difference schemes determined will depend on the nature of ω and γ , whether they are complex or real, and whether they are positive or negative. For the rest of this paper, ω and γ will be assumed to be real and positive.

The convection-diffusion equation is the simplest description of fluid flow involving both convection and diffusion. An example of the type of system that might be modeled by equation (1) would be an ink blot diffusing in laminar flow. The first spatial derivative term will express the flow of the fluid downstream, and the second derivative will express the diffusion of the ink from the concentrated center of the blot. This example and equation bears about the same relation to real physical systems as an introductory physics problem does to real pulleys, artillery shells, car crashes, etc. It is not detailed enough to explain a natural system, but is a description of the general processes which are at work. Extensions from this most simple case are applied in many fluid flow problems. Examples of the type of problems that utilize this form include the following: heat transport, flow through porous media, the spread of contaminants through rivers and estuaries, leaching of salts in soil, mixing in coastal waters, dispersion in open channel flow, and absorption of chemicals into beds¹.

Following this introduction is a section describing the derivation of schemes to approximate (1). The schemes are developed using the modified equivalent partial

differential equation approach developed by Warming and Hyett². In the next section, the stability of the schemes is examined. The von Neumann and matrix stability methods are described, and then applied to the schemes generated in the first section. Other methods of determining stability will be briefly mentioned, and their relevance to the current problem will be outlined. It is shown that there are an infinite number of unconditionally stable schemes of second-order accuracy for equation (1), and an entire family of stable third-order schemes. A fourth-order scheme is presented. Following the determination of stability, the way in which each Fourier mode of the solution is affected by the approximation schemes will be detailed. Numerical tests of the actual stability of the different schemes are presented, as a check on the analytical results presented above.

GENERATION OF SCHEMES

The system to be solved consists of the differential equation, with the appropriate boundary and initial conditions. For real ω , Γ , equation (1) will be referred to as the CDE. As the CDE is one-dimensional, the system is defined along an interval. The initial conditions are given as the values of the solution, assumed exact, at a series of evenly-spaced points along the interval. These will be taken as the solution and time $t = 0$. Call the distance between points h . The total number of points will be called N . The approximate solution to the differential equation over time is modeled by an array of values. Each row is made up of the values of the solution of the differential equation at one time. The amount of time between each row, or time-step, will be called k . Each column of the array is the spatial location at which the value is being measured or approximated. Therefore, the exact value of the solution of the differential equation at a point $x = j \cdot h$ from the left end of the interval, at a time $t = n \cdot k$ from the initial time will be approximated by the array element (n, j) . Calling the approximate solution T , array element (n, j) will be referred to as T_j^n . The initial values are read into the first row of the array, and then from the first row, the values of the next row, the solution one time step in the future, are determined. The value of the boundary points, row elements 0 and N , will be determined from the boundary conditions. This process is then repeated until the row corresponding to the desired future time is completed. In order to approximate a differential equation, each array element at new and the old time levels will be multiplied by a weighting coefficient, so that the solution progresses from one time level to the next in the same fashion as the actual solution of the differential equation would progress. Each point at the new time level will be affected by the value of

the solution at each point along the old time level, and by the value of the solution at each point along the new time level. Solving for a point T_j^{n+1} will then mean solving an equation which is a function of all the values at the old time level, as well as implicitly a function of all the values at the new time level.

An analogy might be the example given above, of an ink blot in a smoothly flowing stream. The stream is assumed to be one-dimensional, with diffusion and convection each occurring along the same axis. The initial conditions would be the concentration of ink at N points, each a distance h apart, spaced along the streamline. The boundary conditions would describe what is happening to the ink at the top and bottom of the interval of the stream under observation. The weighting coefficients at the current time level might be thought of describing how a concentration of ink at one point will affect the concentration of ink at future time points. The weighting coefficients at the new time level could be thought of as describing how the concentration at a point has been affected by the concentrations at points in the past. This is an analogy, not a mathematical description. The weighting coefficient for the array element T_j^n will be called b_0 ; the coefficient for T_{j-1}^n will be b_{-1} ; the coefficient for T_{j+1}^n will be b_1 , and so on. Similarly, the coefficients for the new time level will be called a_{-1} , a_0 , a_1 , etc.

Determining the value of a point at a future time level T_j^{n+1} , will mean solving the following equation:

$$a_{-j+1}T_1^{n+1} + \dots + a_{-1}T_{j-1}^{n+1} + a_0T_j^{n+1} + a_1T_{j+1}^{n+1} + \dots + a_{N-j+1}T_{N-1}^{n+1} = b_{-j}T_0^n + \dots + b_{-1}T_{j-1}^n + b_0T_j^n + b_1T_{j+1}^n + \dots + b_{N-1}T_N^n \quad (2)$$

$$AT_{:n+1} = BT_{:n} + b \quad (4)$$

$$\begin{array}{ccccccc}
a_0 & a_1 & 0 & 0 & \dots & T_1^{n+1} & b_0 & b_1 & 0 & 0 & \dots & T_1^n & (b_{-1}-a_{-1})T_0^n \\
a_1 & a_0 & a_1 & 0 & . & T_2^{n+1} & b_{-1} & b_0 & b_1 & 0 & . & T_2^n & 0 \\
0 & a_{-1} & a_0 & a_1 & . & . & 0 & b_{-1} & b_0 & b_1 & . & . & . \\
. & . & . & . & . & = & . & . & . & . & . & + & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & 0 \\
. & . & a_{-1} & a_0 & . & T_{N-1}^{n+1} & . & . & b_{-1} & b_0 & . & T_{N-1}^n & (b_{-1}-a_{-1})T_N^n
\end{array} \quad (5)$$

and equation (2) to be

$$a_{-1}T_{j-1}^{n+1} + a_0T_j^{n+1} + a_1T_{j+1}^{n+1} = b_{-1}T_{j-1}^n + b_0T_j^n + b_1T_{j+1}^n. \quad (6)$$

Equation (6) could be called the general two-level, six- point scheme. In order to approximate a differential equation, the a's and b's of equation (6) must be chosen such that the approximate solution is moved forward in time in a similar fashion to the solution of the differential equation. Each set of coefficients that corresponds to a differential equation is an approximating scheme for that differential equation.

Schemes which have $a_0 = 1$, and all other $a_n = 0$ are called explicit schemes, because the value at the next time level is given explicitly in terms of known quantities. There is no need to solve for eigenvalues. Schemes which have a_{-1} and/or a_1 as well as a_0 non-zero are called implicit schemes, because unknowns appear implicitly in equation (6). Explicit schemes can determine new time levels quickly, but stability considerations generally limit the allowable size of the time step much more stringently than implicit schemes' time step. Because a constant function would satisfy the CDE, the sum of coefficients at each level must be the same. Without loss of generality, this sum can be set to one.

$$a_{-1} + a_0 + a_1 = b_{-1} + b_0 + b_1 = 1 \quad (7)$$

This gives two equations connecting the a's and b's, leaving only four free parameters. It is convenient to rewrite the parameters as the following:

$$\alpha_+ = a_1 + a_{-1}, \quad \alpha_- = a_1 - a_{-1}, \quad \beta_+ = b_1 + b_{-1}, \quad \beta_- = b_1 - b_{-1} . \quad (8)$$

A powerful method of determining sets of coefficients was developed by Warming and Hyett: the modified partial differential equation approach². Warming and Hyett assume that there is a function which exactly equals the approximate solution at every point (n,j) , and that that function is continuously differentiable in time and space. They then take a Taylor series about the point (n,j) . Expanding in time and space, this gives a relation between neighboring points, and the derivatives of the function at (n,j) . Each of the six points in equation (6) can be replaced by an infinite sum of derivatives, taken at (n,j) . Substituting into equation (6), and pulling all terms to one side, gives an infinite series of time, mixed, and spatial derivatives at the point (n,j) . This summation will be in terms of the α 's and β 's, h , and k . In order to develop a usable equation, unwanted derivatives must be eliminated. These include all mixed derivatives, and time derivatives beyond the first. This will result in an equation with one time derivative, and spatial derivatives of every order. Given a set of the α 's, β 's, h , and k , the resulting equation, called the Modified Equivalent Partial Differential Equation, or MEPDE, is the actual differential equation that is exactly solved by the approximation scheme. In order to eliminate all unwanted derivatives, write the original complete summation as the first row in a table. Table 1 shows the table generated by the following process. As the summation is equal to zero, a multiple and/or derivative of the summation can be added without harm. Take the derivative with respect to time, and multiply by the negative of

the ratio of the second time derivative and the first time derivative. Adding this multiple of the derivative to the original equation will give a new equation without the second time derivative. Write the equation found by taking the derivative underneath the original equation, with each similar derivative term aligned as a column. By taking the corresponding derivatives and multiples of the original equation, succeeding terms of the equation given by summing all the rows of the table can be eliminated. When the final equation will have the first time derivative, as many spatial derivatives as desired, and no other terms of order equal to or less than the highest order spatial derivative, the rows can be summed, giving the MEPDE. The MEPDE so generated, up to fourth order, is the following:

$$\begin{aligned}
 \partial T / \partial t + ((\alpha_- - \beta_-)h/k) \partial T / \partial x = & (1/2)(h^2/k)[(\beta_+ - \alpha_+) - (\beta_-^2 - \alpha_-^2)] \partial^2 T / \partial x^2 \\
 & + (1/6)(h^3/k)[(\beta_- - \alpha_-) + 3(\alpha_- \alpha_+ - \beta_- \beta_+) + 2(\beta_-^3 - \alpha_-^3)] \partial^3 T / \partial x^3 \\
 & + (h^4/24k)(2(\alpha_-^2 - \beta_-^2) - (\alpha_+ - \beta_+)[1 - 3(\alpha_+ + \beta_+) + 4(\alpha_-^2 + \alpha_- \beta_- + \beta_-^2)]) \partial^4 T / \partial x^4 \\
 & + \text{order}(h^5) .
 \end{aligned} \tag{9}$$

In order to approximate a differential equation, all that is required is to choose the α 's and β 's, as functions of h and k , such that the coefficients of the MEPDE equal the coefficients of the actual differential equation up to the order of accuracy desired. As there are four free parameters, up to fourth-order accuracy can be achieved. For explicit schemes, α_+ and α_- will be zero, leaving two free parameters to vary. Each order of accuracy desired gives one restriction on the allowable values of the α 's and β 's. The four equations given by matching the CDE to the MEPDE are the following:

$$\alpha_- - \beta_- = c \quad (10a)$$

$$(\beta_+ - \alpha_+) - (\beta_-^2 - \alpha_-^2) = 2\Gamma \quad (10b)$$

$$(h^3/6k) [(\beta_- - \alpha_-) + 3(\alpha_- \alpha_+ - \beta_- \beta_+) + 2(\beta_-^3 - \alpha_-^3)] = 0 \quad (10c)$$

$$(h^4/24k) \{ 2(\alpha_-^2 - \beta_-^2) - (\alpha_+ - \beta_+) [1 - 3(\alpha_+ + \beta_+) + 4(\alpha_-^2 + \alpha_- \beta_- + \beta_-^2)] \} = 0. \quad (10d)$$

h and k have been rewritten as the variables c and Γ . $c = \omega k / h$, and $\Gamma = \gamma k / h^2$. c is often referred to as the Courant number. The equations are listed in order. For first order accuracy, (10a) must be satisfied; for second order, (10a) and (10b), and so on. Satisfying all four will generate a unique fourth order accurate scheme. For schemes of lower accuracy, one or more variables will be unrestricted. The free variables can then be chosen so as to maximize the resulting scheme's stability. Schemes of first through third order accuracy will be presented in the following section on stability. The fourth order scheme is given below:

$$\begin{aligned} \beta_- &= (-c^4 + c^2 - 2\Gamma - 12\Gamma^2 + 4\Gamma c^2) / (2c^3 - 2c + 24\Gamma^2/c) \\ \alpha_- &= (c^4 - c^2 - 2\Gamma + 12\Gamma^2 + 4\Gamma c^2) / (2c^3 - 2c + 24\Gamma^2/c) \\ \beta_+ &= -(1/3 + 2\Gamma - c^2/3) + [(c^6 + c^4(-1 - 6\Gamma) + c^2(20\Gamma^2 + 4\Gamma) \\ &\quad - 4\Gamma^2 - 24\Gamma^3) / (2c^4 - 2c^2 + 24\Gamma^2)] \\ \alpha_+ &= (-1/3 - 4\Gamma + 4c^2/3) + [(c^6 + c^4(1 + 2\Gamma) + c^2(-4\Gamma^2) \\ &\quad - 4\Gamma^2 - 24\Gamma^3) / (2c^4 - 2c^2 + 24\Gamma^2)] \end{aligned} \quad (11)$$

Although this scheme is very accurate, it is cumbersome, and evaluation of stability is very difficult.

Explicit schemes have only two free parameters, and so can only achieve second-order accuracy. The second-order accurate explicit scheme is well known:

$$\alpha_+ = \alpha_- = 0, \quad \beta_- = -c, \quad \beta_+ = c^2 + 2\Gamma \quad . \quad (12)$$

This scheme is commonly called the Lax-Wendroff scheme.

A much-studied method of scheme generation leads to Pade schemes. This type of scheme will be discussed in more detail, as the stability of such schemes is easier to determine than the general case. The generation of Pade schemes begins with the method of line or semi-discretization³. This method approximates the CDE by a system of ordinary differential equations of the form

$$dT/dt = M T(t) + c(t) \quad . \quad (13)$$

where $T(t) = (T(h,t), T(2h,t), \dots, T((N-1)h,t))$ and M is a tri-diagonal $(N-1) \times (N-1)$ matrix. $c(t)$ is a $(n-1)$ -component vector which arises from the boundary values. It will be neglected for the moment. The solution to equation (13) then becomes

$$T(t+k) = \exp(kM) T(t) \quad . \quad (14)$$

A Pade scheme is a scheme which is generated by approximating $\exp(kM)$. A family of schemes is labeled by how many terms are retained in the approximation. A six-point scheme will be generated by the (1-1) approximation:

$$e^{kM} = (I + \theta_1 kM) / (I + \theta_2 kM) \quad . \quad (15)$$

The matrices $(I + \theta_1 kM)$ and $(I + \theta_2 kM)$ share the same set of eigenvectors. They correspond to the matrices A and B , respectively, from equation (4). θ_1 and θ_2 are

scalars, bound by the restriction $\theta_1 - \theta_2 = 1$. This gives a family of schemes. The family includes strictly explicit schemes, strictly implicit schemes, and mixed schemes. The famous Crank-Nicholson scheme is of this family:

$$\alpha_+ = -\Gamma, \quad \beta_+ = \Gamma, \quad \alpha_- = c/2, \quad \beta_- = -c/2. \quad (16)$$

The Crank-Nicholson is the only previously published second-order, six-point scheme for the CDE that is unconditionally stable. It can be shown the coefficients of all Pade schemes will obey the following relation:

$$\alpha_+ / \alpha_- = \beta_+ / \beta_- . \quad (17)$$

There are other methods of generating schemes, and many schemes have been generated in the past for this equation. Listed below are several which are the best known, or are connected to this paper.

1) The FTCS scheme. FTCS stands for Forward Time, Centered Space. This scheme is generated by substituting in the forward difference for the time derivative, and the centered differences for the spatial derivatives. The resulting scheme is the following:

$$(T_j^{n+1} - T_j^n)/k + \omega(T_{j+1}^n - T_{j-1}^n)/2h = \gamma(T_{j+1}^n - 2T_j^n + T_{j-1}^n)/h^2. \quad (18)$$

In terms of β_+ and β_- , this is written:

$$\alpha_+ = 0, \quad \alpha_- = 0, \quad \beta_- = -c, \quad \beta_+ = 2\Gamma. \quad (19)$$

This scheme is also called the Explicit scheme. This is the most straightforward scheme to generate.

2) The Lax-Wendroff scheme, mentioned above. This scheme is special because it is the only second-order explicit scheme.

3) The Crank-Nicholson scheme, also mentioned above. This scheme is the only previously-published, unconditionally stable, second-order, six point difference scheme for the CDE. It is also the prime representative of the Pade type scheme.

4) Noye and Tans' third-order scheme¹. This equation was introduced in a paper in December, 1988. It is the only previously-published third-order six point scheme for the CDE. Noye and Tan use the MEPDE method on five data points to develop a third-order scheme. This paper was the original inspiration for this thesis, which uses the same method on six points, instead of assuming $a_1 = 0$.

STABILITY

A scheme will be stable if the initial error associated with each term decreases with time, and if the values at each time level do not increase to infinity. If the actual solution should increase to infinity with time, then stability implies the approximate solutions are not increasing faster than the actual solution. A scheme may be stable, but not accurate. To usefully approximate a differential equation, a scheme must be stable and accurate. Accuracy will be discussed in a following section of this paper. There are several methods of checking a scheme's stability. These include the von Neumann or Fourier method, the Matrix method, the energy method, and L_0 stability. Each method makes certain assumptions about the scheme and the domain under approximation, so each may give different results when applied to the same scheme. Often a scheme's stability will depend on the values chosen for h and k . A scheme that is unstable for all choices of h and k is called unconditionally unstable; a scheme whose stability depends on the values of h and k chosen is called conditionally stable; a scheme which is stable for all h and k is called unconditionally stable.

A major objective of this thesis is to present unconditionally stable schemes which approximate the CDE.

VON NEUMANN STABILITY

The simplest method of checking for stability is the von Neumann, or Fourier method. The von Neumann method assumes an infinite domain, given by periodic boundary conditions. Although it can be used as a rough guide for systems with finite domains, it is strictly true only for systems with an infinite domain. The von Neumann

stability is calculated as follows. The effect of the scheme upon a Fourier mode is determined. If the amplitude of a Fourier mode will be increased as it is propagated from one time level to the next, the scheme will eventually blow up. If the amplitude of every Fourier component is decreasing or staying the same with time, the scheme is said to be stable. The ratio of the magnitude of a Fourier mode of wavenumber k one time level ahead, $A_{n+1}(k)$, versus the magnitude of that mode at the current level, $A_n(k)$, is called the Amplification Factor, written as $G(k)$. Note that " k " here refers to a wavenumber, not the time step; this unfortunate notation is chosen to agree with the literature. The quantity $k\Delta x$ is referred to as β . The strict von Neumann condition is given below:

$$|G(k)| \leq 1, \text{ for all } k \text{ such that } 0 \leq \beta \leq \pi. \quad (20)$$

The strict von Neumann condition is applicable to schemes which are not approximating an increasing solution. This will include schemes for the CDE. If the solution is expected to exponentially increase, a more general form is used:

$$|G(k)| \leq 1 + O(k), \text{ for all } k \text{ such that } 0 \leq \beta \leq \pi. \quad (21)$$

Equation (6) will have a solution of the form of:

$$\tau_j^n = A_n(k) e^{ij\beta}. \quad (22)$$

Substituting equation (22) into equation (6) gives the following expression:

$$A_{n+1}(k) (a_1 e^{i\beta} + a_0 + a_{-1} e^{-i\beta}) = A_n(k) (b_1 e^{i\beta} + b_0 + b_{-1} e^{-i\beta}). \quad (23)$$

This can be rearranged to become the following:

$$G(k) = [(1-\beta_+)+\beta_+\cos\beta + i \beta_+\sin\beta]/[(1-\alpha_+) + \alpha_+\cos\beta + i \alpha_+\sin\beta] . \quad (24)$$

Equation (24) is the general statement of von Neumann stability. It is possible to rewrite this condition, for real α 's and β 's, as the following two equations⁴:

$$\alpha_-^2 - \beta_-^2 - (\alpha_+ - \beta_+) \geq 0 \quad (25a)$$

$$(\alpha_+ - \beta_+) (\alpha_+ + \beta_+ - 1) \geq 0 . \quad (25b)$$

This is the condition for von Neumann stability for the general two-level, six point scheme, equation (6). The stability of a particular scheme can be determined directly by substituting in the appropriate values for the α 's and β 's.

It is useful to express the general stability requirements graphically. If a graph is imagined which is drawn on the α_+ , α_- , β_+ , and β_- axes, a point on the graph will represent a specific set of values for all four variables, and will thus define a discretization scheme. The entire space spanned by the four variables would include every possible two-level, six-point discretization scheme. The equations for stability, equations (25), will then define a region in that space. A scheme with the α 's and β 's expressed in terms of h and k will show up as a path in the four-space. If the path does not cross the region of stability, the scheme will be unconditionally unstable. If it does cross into or out of the region of stability, it will be conditionally stable, and the conditions for its stability will be given by the conditions on h and k which cause it to leave the region of stability. If a scheme's path is entirely within the region of stability, it will be unconditionally stable. Fortunately, the scheme must also correspond to the differential equation under consideration. The equations that match a differential equation to the MEPDE will cut surfaces through the region of stability, and only those

surfaces need to be graphed, reducing the dimension of the region down to a manageable number. For explicit schemes, α_+ and α_- will be zero, reducing the graph to two dimensions. As there is no use for a first-order implicit scheme, second-order will be considered. This again reduces the dimension of the region under consideration down to two.

For explicit schemes, the equations (25) reduce to:

$$\beta_-^2 \leq \beta_+ \leq 1. \quad (26)$$

For implicit schemes of second order, the first of the implicit stability equations, (25a), will be satisfied already, because equations (10a) and (10b) must be satisfied. The two accuracy equations also give β_- and α_- in terms of β_+ and α_+ , so a second order scheme will be entirely determined by β_+ and α_+ . The remaining stability equation, (25b), is only in terms of β_+ and α_+ , so only the values of β_+ and α_+ are significant in determining stability. A graph of (25b) is given as figure 1 *Achieving second-order accuracy only restricts two variables, so β_+ and α_+ can be freely chosen. Any choice of β_+ and α_+ which will lie within the stable regions for all h and k will represent an unconditionally stable scheme.* One unconditionally stable six-point scheme for the CDE has been previously determined, the Crank-Nicholson scheme. It lies along the line $\beta_+ = -\alpha_+$, in the second quadrant. It is clearly within the stable region for all h and k . Similarly, any number of unconditionally stable schemes can be generated by carefully choosing β_+ and α_+ . The unconditionally stable schemes can be grouped into families, such as the following:

$$\begin{aligned}\alpha_+ - \beta_+ &= -2\Gamma \\ \alpha_+ + \beta_+ &= p \quad ; \quad p \text{ real, } \leq 1 \quad .\end{aligned}\quad (27)$$

This is the family of straight lines parallel to the line $\beta_+ + \alpha_+ = 1$, and inside the left stable region. The Crank-Nicholson scheme corresponds to the case $p = 0$. Another family is shown below:

$$\begin{aligned}\alpha_+ - \beta_+ &= c^2 \\ \alpha_+ + \beta_+ &= q \quad ; \quad q \text{ real, } \geq 1 \quad .\end{aligned}\quad (28)$$

This is the family of straight lines parallel to the line $\beta_+ + \alpha_+ = 1$, and inside the right stable region. The following family contains both the previous families:

$$(\beta_+ - 1/2) = r (\alpha_+ - 1/2) \quad ; \quad r \text{ real} \quad . \quad (29)$$

This family contains all points in both stable regions.

It is also possible to generate third-order accurate unconditionally stable schemes. Such schemes must satisfy equations (10a), (10b), (10c), and (25b) simultaneously. Equations 10 a-c can be used to solve α_+ , β_+ , and α_- in terms of β_+

$$\alpha_- = \beta_- + c \quad (30a)$$

$$\alpha_+ = (1/3 + 2c^2/3) + \beta_-(2\Gamma/c + c) \quad (30b)$$

$$\beta_+ = (1/3 + 2\Gamma - c^2/3) + \beta_-(2\Gamma/c - c) \quad . \quad (30c)$$

(25b) can be rewritten in terms of β_- :

$$[(c^2 - 2\Gamma) + 2c\beta_-] [(c^2/3 + 2\Gamma - 1/3) + 4\Gamma\beta_-/c] \geq 0. \quad (31)$$

A choice of β_- that fills this requirement will give rise to an unconditionally stable third-order scheme. The third-order schemes are long and complicated, and would be difficult to apply. An example of a third-order scheme is the following:

$$\begin{aligned} \beta_- &= \Gamma/c + c/(12\Gamma) & \alpha_- &= (c^2 + \Gamma)/c + c/(12\Gamma) \\ \alpha_+ &= 1/2 + \Gamma + 2c^2/3 + 2\Gamma^2/c^2 + c^2/(12\Gamma). \end{aligned} \quad (32)$$

Graphs will be shown below of the von Neumann, matrix, and actual stability regions for the FTCS and Lax-Wendroff schemes.

MATRIX METHOD STABILITY

The second major method of determining stability is the Matrix Method. The matrix method is more difficult to apply than the von Neumann method, but it is applicable to systems a finite domain. The effects of the boundary conditions are directly included in the determination of stability. The matrix method starts from the matrix representation of the approximation scheme, equation (4). This can be rewritten as the following:

$$\mathbf{T}_j^{n+1} = \mathbf{A}^{-1}\mathbf{B} \mathbf{T}_j^n + \mathbf{A}^{-1}\mathbf{b}. \quad (33)$$

A scheme will be matrix stable if the magnitude of every eigenvalue of $\mathbf{A}^{-1}\mathbf{B}$ is equal to or less than one, and if no eigenvalue with a magnitude equal to one is repeated. This directly includes the finite nature of the domain and the boundary conditions of the system, because the size of the matrix will depend on the number of points taken, the values of each element will depend on the step size chosen, and the first and last rows of

the matrix will be affected by the boundary conditions. The necessity of finding the eigenvalues of a large matrix makes this stability method difficult to apply, especially in general. There is no easy way to determine what the eigenvalues of the product of A^{-1} and B will be in general. This is a currently an area of considerable research in Linear Algebra. For special cases, the eigenvalues may be easy to determine. The two prime examples are explicit schemes and Pade schemes, with homogeneous boundary conditions. Each will be described in detail. Otherwise, determining the stability of a scheme would mean substituting in a particular set of h and k into the scheme's expressions for the α 's and β 's, finding the values of each array element, performing the inversion and multiplication, then finding the eigenvalues. Although all of this is straightforward, the amount of time required is prohibitive, especially as each scheme would require testing at a number of values for h and k .

It is straightforward to determine the eigenvalues of a tri-diagonal matrix, all of whose rows are identical. An $N \times N$ matrix of this form, such as matrix B in equation (33), will have eigenvalues of the following form:

$$\lambda_j = 1 - \beta_+ + (\beta_+^2 - \beta_-^2)^{1/2} \cos(j\pi/(N+1)) , \quad j = 1, 2, \dots, N . \quad (34)$$

The easiest type of schemes to apply the matrix method to are explicit schemes. Here, A is the identity matrix, so the desired eigenvalues will be given by equation (34) directly. This is assuming that the boundary conditions are homogeneous, so that the first and last rows of the matrix do not need to be adjusted to match more difficult boundary conditions. So for explicit schemes, the matrix method stability is given by a straightforward expression:

$$1 \geq |\lambda_j| = 1 - \beta_+ + (\beta_+^2 - \beta_-^2)^{1/2} \cos(j\pi/(N+1)) , j = 1, 2, \dots, N . \quad (35)$$

Taking the limit as N goes to infinity, meaning an infinite domain, this can be reworked into the following condition for matrix stability of the general explicit scheme:

$$2\beta_+ \leq \beta_-^2 \leq 4(\beta_+ - 1) . \quad (36)$$

Implicit schemes can be easily tested for matrix stability if the eigenvectors of A and B are the same. If the eigenvectors are the same, then the eigenvalues of $A^{-1}B$ will be equal to the eigenvalues of B divided by the eigenvalues of A . Pade schemes will have matrices A and B which have the same eigenvectors. The eigenvalues of $A^{-1}B$ then become:

$$\lambda_j = \frac{[1 - \beta_+ (1 - (1 - (\beta_+/\beta_-)^2)^{1/2} \cos(j\pi/(N+1)))]}{[1 - \alpha_+ (1 - (1 - (\alpha_+/\alpha_-)^2)^{1/2} \cos(j\pi/(N+1)))]} . \quad (37)$$

This allows for easy testing of the matrix stability of Pade schemes. Direct substitution of equation (16) into the above equation shows that the Crank-Nicholson scheme is unconditionally stable by the Matrix method.

ENERGY METHOD STABILITY

A more precise, but much more difficult method of determining stability is the energy method. This will be outlined here, but will not be applied to any scheme. The energy method involves determining the sum of the square of the magnitude of the each of the values at a time level. If this sum is increasing, the scheme is blowing up. There is a method of determining whether the sum will be increasing and decreasing without

actually running the scheme, and in a more general form, but it is extremely complex. The energy method was not successfully applied to the schemes described or generated in this thesis.

L_0 STABILITY

L_0 stability tests for a different type of instability. A solution may be von Neumann and Matrix stable and still have oscillations when run. When this occurs, the value the scheme determines at each data point does not converge to a single value, but oscillates wildly with each time step. The oscillations slowly decrease, or they may increase, and head to infinity. If a scheme is matrix stable and free of oscillations, it is called L_0 stable. L_0 stability is determined by first finding the eigenvalues, as in matrix stability, and then taking the limit of the eigenvalue as the time step goes to infinity. If the eigenvalue goes to zero, the scheme may be L_0 stable, and free of oscillations. If not, the scheme will oscillate. The magnitude of the oscillations will often depend on the size of the time step, and so using a sufficiently small time step will scale down the oscillations to an acceptable size. The Crank-Nicholson is severely afflicted with oscillations⁵, which reduces its practical usefulness. It is very difficult to determine when an implicit scheme is L_0 stable, because the matrix method must be determined first. From the expression for the eigenvalues of the Pade schemes, equation (37), and equation (17), it can be seen that for large values of α_+ and β_+ , the eigenvalues approach α_+/β_+ . For the Crank-Nicholson, this is clearly unstable, as $\alpha_+/\beta_+ = -1$. It is suspected that all schemes in the second and fourth quadrants of figure 1 will have oscillations for the same reasons. No scheme was discovered which could be demonstrated to be von Neumann and matrix unconditionally stable, and free of oscillations for all sizes of the time step.

DIFFUSION

It was stated in the last section that a scheme must be stable and accurate to be useful. The last section described how to determine a scheme's stability; this section will describe how to analytically estimate a scheme's accuracy. This is done by examining how the scheme will propagate a single Fourier mode of the solution. By comparing how the differential equation actually propagates the Fourier mode, and how the scheme will, it can be estimated how accurate the scheme will be. While advancing through time, the amplitude of different Fourier modes will be adjusted differently by the difference scheme. How the scheme affects the amplitudes of the Fourier modes is called the scheme's *dissipation*. Fourier modes will also be propagated at different speeds. How fast the scheme propagates different modes is called the scheme's *dispersion*, because the modes will tend to separate out, or to disperse. The combination of dispersion and dissipation for a scheme is called the scheme's diffusion. It is the diffusion which must be checked to determine whether a scheme will continue to give accurate results.

The dissipation of a scheme can be determined directly from G , the amplification factor. Equation (24) describes G for the general six-point scheme. The magnitude of G can be determined to be⁴:

$$|G(k)| = 1 - (1/2)(1 - \cos\beta) \{ [2((\alpha_+^2 - \alpha_-^2) - (\beta_+^2 - \beta_-^2))(1 - \cos\beta) + 8\Gamma] / [(\beta_+^2 - \beta_-^2)(1 - \cos\beta)^2 + 2(\alpha_-^2 - \alpha_+)(1 - \cos\beta) + 1] \} . \quad (38)$$

This can be compared to the exact dissipation of the CDE⁵:

$$|G_{\text{exact}}(k)| = e^{-(\gamma k \Delta t)} . \quad (39)$$

k in this equation refers to the wavenumber of the mode under observation. γ is the diffusive parameter from equation (1). A scheme's stability is often shown by a polar plot of the magnitude of G against β . In the following section, dissipation curves will be shown for each of the schemes presented.

The dispersion of a scheme can also be found directly from G . The angle through which a Fourier mode is advanced by a difference scheme will be given by⁶:

$$\phi(k) = \tan^{-1} (\text{Im} (G(k)) / \text{Re} (G(k))) \quad . \quad (40)$$

Putting in the general form of G , Equation (24), this becomes:

$$\phi(k) = \tan^{-1} \left(\frac{(\alpha_- \sin\beta)[1 + \beta_+(\cos\beta-1)] + (\beta_- \sin\beta)[1 + \alpha_+(\cos\beta-1)]}{\{ [1 + \beta_+ (\cos\beta - 1)] [1 + \alpha_- (\cos\beta - 1)] + \alpha_- \beta_- \sin^2\beta \}} \right) \quad . \quad (41)$$

The actual dispersion of the CDE is (Paolucci and Chenoweth, 1982):

$$\phi_{\text{exact}}(k) = c\beta \quad . \quad (42)$$

The dispersion is also often portrayed as a polar graph of the diffusion divided by the exact diffusion, versus β . Dispersion curves will be shown in the following section.

For a scheme to be accurate, both its dissipation and dispersion must be close to that of the actual equation. The diffusion of specific schemes will be presented along with the stability tests for individual schemes, in the following section for numerical verification.

NUMERICAL TESTS

Although the stability of a scheme may be tested in a number of ways, the only certain method to determine stability is to actually run the scheme. It can then be seen directly whether the scheme is blowing up or not. This section will discuss the testing of the schemes' numerical stability, and compare the results with the analytic stability outlined above. The code used to implement the schemes is given as the Appendix. The system used for the testing consisted of the CDE, over a region one unit long, boundaries held at zero, and tent-shaped initial conditions, rising to one at $x = 1/2$. If a scheme is afflicted with oscillations, the effect is most pronounced for the high frequency Fourier modes. The scheme may be essentially stable for smooth initial conditions, but unstable for initial conditions with sharp edges. The initial conditions for the test case were taken to be a straight line, a point, and a straight line back to zero in order to emphasize the high frequency modes. The easiest method of testing stability, although not accuracy, is to run the scheme for very long time periods. As the actual solution will diffuse, and flow out of the domain into the zero boundary conditions, the actual solution will go to zero rather quickly. Testing stability then requires looking after a long time, and seeing if there is any solution left. If there are still non-zero values at any point in the scheme, a mode is not being damped to zero with all the rest, and the scheme is unstable.

Although the stability test described above will determine if a Fourier mode is not being damped to zero, it will not show if the mode is being damped too fast. In order to determine the accuracy of the scheme, the diffusion of the scheme must be checked directly. This is done by substituting into the general equations, equations (38) and

(41). The dispersion and dissipation curves for a scheme will depend on the spatial and time step.

The exact solution to the CDE can be determined for the test case, given the boundary and initial conditions. The accuracy of a scheme can be directly tested by comparing the approximate values generated by the scheme against the actual solution. The error can be thought of as the values of an "error vector," an $N-1$ dimensional vector, whose j^{th} component is the error at the j^{th} spatial step. A convenient expression for the error at one time level is the norm of the error vector at that time level. This is calculated by summing the square of each element, and taking the square root of the sum. The proportional error can be determined by dividing by N , and the percent error is found by dividing the error norm by the norm of the actual solution at that time level.

Moving from the general to the specific, the following is a listing of some of the previous schemes for the CDE, and some of the new schemes generated. Each scheme is given a page showing the following information: The scheme; the order of accuracy; the von Neumann stability; the matrix stability, for explicit and Pade schemes; a table of the error norm of the scheme for the test case; and the dispersion and dissipation results for implicit schemes. Explicit schemes are represented by the FTCS scheme, the Lax-Wendroff scheme, and a new scheme similar to the FTCS scheme. Figures 2, 3, and 4 show the stability of the FTCS, Lax-Wendroff, and FTCS, $\rho = .2$ schemes. Figures 5 - 16 show the dissipation and dispersion curves for each implicit scheme. Implicit schemes of second, third and fourth order are given. The Crank-Nicolson scheme is listed, along with a selection of three new second-order schemes. The third-order scheme listed previously is included, and lastly, the fourth-order scheme is described.

FTCS SCHEMEScheme:

$$\alpha_- = \alpha_+ = 0, \quad \beta_- = -c, \quad \beta_+ = 2 \Gamma$$

First-order accurateStability:Von Neumann stability:

$$k \leq h^2/(2 \Gamma)$$

Matrix Stability: (Infinite approximation)

$$k \leq (2/\omega) [2k/\omega - (4k^2/\omega^2 - h^2)^{1/2}], \text{ for } h \leq 2 \kappa / \omega .$$

$$k \leq 4 \kappa / (\omega^2) , \text{ for } h \geq 2 \kappa / \omega .$$

Numerical Stability:

<u>dx</u>	<u>von Neumann Stability</u>	<u>Matrix Stability</u>	<u>Actual Stability</u>
.1	.10000	.20000	.1428
.083	.06944	.08944	.0833
.071	.05102	.06003	.0588
.0625	.03906	.04388	.0323
.05	.02500	.02679	.0263
.038	.01479	.01538	.0154
.029	.00865	.00885	.0111
.02	.00400	.00404	.0029
.01	.00100	.00100	.0012

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
.0466	2.09	.100	.025
.0463	2.08	.100	.020
.0895	4.03	.100	.017
.0377	1.69	.083	.025
.0374	1.68	.083	.020
.0665	2.99	.083	.017
.0323	1.45	.071	.025
.0321	1.44	.071	.020
.0557	2.50	.071	.017
.0320	1.44	.063	.025
.0318	1.43	.063	.020
.1135	5.10	.063	.017
.0237	1.07	.050	.0125
.0191	0.86	.038	.0125
.0138	0.62	.029	.0100
.0363	1.63	.020	.0029

LAX-WENDROFF SCHEME

Scheme:

$$\alpha_+ = \alpha_- = 0, \quad \beta_- = -c, \quad \beta_+ = c^2 + 2\Gamma$$

Second-order accurate

Stability:

Von Neumann stability:

$$k \leq (1/\omega) [(h^2 + k^2/\omega^2)^{1/2} - k/\omega^2]$$

Matrix Stability:

$$k \leq ((h^2 / \omega^2) + 4 \kappa / (3 \omega^2))^{1/2} - (4 \kappa / (3 \omega^2))^2$$

Numerical Stability:

<u>dx</u>	<u>von Neumann Stability</u>	<u>Matrix Stability</u>	<u>Actual Stability</u>
.1	.06180	.06666	.0666 - .0714
.083	.04718	.05040	.05 - .053
.071	.03719	.03939	.038 - .040
.063	.03004	.03158	.0312 - .0333
.05	.02071	.02153	.0213 - .0222
.038	.01308	.01344	.0133 - .0137
.029	.00801	.00815	.0081 - .0082
.02	.00385	.00389	.0021 - .0022
.01	.00099	.00099	.0009 - .001

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
.0474	2.13	.100	.050
.0467	2.10	.100	.033
.0910	4.09	.100	.025
.0381	1.71	.083	.050
.0375	1.69	.083	.033
.0661	2.97	.083	.025
.0332	1.49	.071	.050
.0324	1.46	.071	.033
.0604	2.72	.071	.025
.0337	1.51	.063	.050
.0320	1.44	.063	.033
.1135	5.10	.063	.025
.0908	4.08	.050	.0125
.0534	2.40	.029	.0077
.0363	1.63	.020	.0021
.0182	0.82	.010	.0009

FTCS FAMILY, P = .2Scheme:

$$\alpha_- = \alpha_+ = 0, \quad \beta_- = -c, \quad \beta_+ = .2 \Gamma$$

First order accurateStability:Von Neumann stability:

$$k \leq h^2 / (.2 \Gamma)$$

Matrix method stability:

$$k \leq (2 / \omega) [.2 \kappa / \omega - ((.04) \kappa^2 / \omega^2 - h^2)^{1/2}], h \leq .2 \kappa / \omega .$$

$$k \leq .4 \kappa / \omega^2, \quad h \geq .2 \kappa / \omega .$$

Numerical stability:

<u>dx</u>	<u>von Neumann Stability</u>	<u>Matrix Stability</u>	<u>Actual Stability</u>
.1	.10000	.02000	.0217 - .0227
.083	.06944	.02000	.0208 - .0217
.071	.05102	.02000	.0200 - .0208
.0625	.03906	.02000	.0200 - .0208
.05	.02500	.02000	.0200 - .0208
.038	.01479	.02000	.0172 - .0179
.029	.00865	.02000	.0100 - .0125
.02	.00400	.02000	.0017 - .0018
.01	.00100	.02000	NOT AVAILABLE

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
.0497	2.23	.100	.020
.0496	2.23	.100	.017
.0958	4.31	.100	.014
.0418	1.88	.083	.020
.0416	1.87	.083	.017
.0717	3.22	.083	.014
.0351	1.58	.071	.020
.0351	1.58	.071	.017
.0590	2.65	.071	.014
.0322	1.45	.063	.020
.0321	1.44	.063	.017
.1135	5.10	.063	.014
.0247	1.11	.050	.0125
.0202	0.91	.038	.0125
.0534	2.40	.029	.0100
.0363	1.63	.020	.0016

CRANK-NICHOLSON SCHEME

Scheme:

$$\alpha_- = c/2, \quad \beta_- = -c/2, \quad \alpha_+ = -\Gamma, \quad \beta_+ = \Gamma.$$

Second order accurate:

Stability:

Von Neumann stability:

Unconditionally Stable.

Matrix Stability:

Unconditionally Stable.

L₀ Stability:

Unstable, oscillations affect solution.

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.181656	8.435	.100	.100
0.044610	2.071	.100	.050
0.044884	2.084	.100	.033
0.044937	2.087	.100	.025
0.044955	2.087	.100	.020
0.076476	3.551	.050	.100
0.023348	1.084	.050	.050
0.023180	1.076	.050	.033
0.023225	1.078	.050	.025
0.023234	1.079	.050	.020
0.031279	1.452	.033	.100
0.015777	0.733	.033	.050
0.015484	0.719	.033	.033
0.015600	0.724	.033	.025
0.015587	0.724	.033	.020

The dissipation and dispersion curves for the Crank-Nicholson scheme are shown in Figures 5 and 6.

2A SCHEME

Scheme:

$$\alpha_+ = -87 \Gamma, \quad \beta_+ = -86 \Gamma, \quad \beta_- = (\Gamma - c^2) / 2c, \quad \beta_+ = (\Gamma + c^2) / 2c .$$

Second order accurate

Stability:

Von Neumann stability:

Unconditionally Stable.

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.181656	8.435	.100	.100
0.004646	0.216	.100	.050
0.006601	0.307	.100	.033
0.009725	0.452	.100	.025
0.012601	0.585	.100	.020
0.078457	3.643	.050	.100
0.002321	0.108	.050	.050
0.003317	0.154	.050	.033
0.004903	0.228	.050	.025
0.006363	0.295	.050	.020
0.035709	1.658	.033	.100
0.001547	0.072	.033	.100
0.002213	0.103	.033	.033
0.003274	0.152	.033	.025
0.004250	0.197	.033	.020

The dissipation and dispersion curves for the 2A scheme are shown in Figures 7 and 8.

2B SCHEME

Scheme:

$$a_+ = -\Gamma, \quad \beta_+ = \Gamma/2, \quad \alpha_- = (\Gamma/2 + c^2)/(2c), \quad \beta_- = (\Gamma/2 - c^2)/(2c) \quad .$$

Second order accurate

Stability:

Von Neumann stability:

Unconditionally Stable.

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.181656	8.435	.100	.100
0.043242	2.008	.100	.050
0.044128	2.008	.100	.033
0.044380	2.049	.100	.025
0.044498	2.061	.100	.020
0.076510	2.066	.050	.100
0.022292	1.035	.050	.050
0.022650	1.052	.050	.033
0.022760	1.057	.050	.025
0.022811	1.059	.050	.020
0.031262	1.452	.033	.100
0.015014	0.697	.033	.050
0.015190	0.705	.033	.033
0.015271	0.709	.033	.025
0.015305	0.711	.033	.020

The dissipation and dispersion curves for the 2B scheme are shown in Figures 9 and 10.

2C SCHEME

Scheme:

$$\alpha_+ = -2\Gamma, \quad \beta_+ = 0, \quad \alpha_- = c/2, \quad \beta_- = -c/2.$$

Second order accurate

Stability:

Von Neumann stability:

Unconditionally Stable.

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.181656	8.435	.100	.100
0.040402	1.876	.100	.050
0.042662	1.981	.100	.033
0.043427	2.016	.100	.025
0.043812	2.034	.100	.020
0.076498	3.552	.050	.100
0.020846	0.968	.050	.050
0.022072	1.025	.050	.033
0.022472	1.043	.050	.025
0.022669	1.053	.050	.020
0.031402	1.458	.033	.100
0.014000	0.650	.033	.050
0.014827	0.688	.033	.033
0.015090	0.701	.033	.025
0.015219	0.707	.033	.020

The dissipation and dispersion curves for the 2C scheme are shown in Figures 11 and 12.

3D SCHEME

Scheme:

$$\beta_- = \Gamma/c + c/(12 \Gamma) , \quad \beta_+ = 1/2 + \Gamma - c^2/3 + 2 \Gamma^2/c^2 - c^2/(12 \Gamma)$$

$$\alpha_- = (c^2 + \Gamma)/c + c/(12 \Gamma) , \quad \alpha_+ = 1/2 + \Gamma + 2c^2/3 + 2 \Gamma^2/c^2 + c^2/(12 \Gamma) .$$

Third order accurate

Stability:

Von Neumann stability:

Unconditionally stable.

Numerical stability is terrible, immediate instability.

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.18165	8.435	.100	.100
1707.172000	79268.760	.100	.033
29606.940000	1374733.000	.100	.020
473.908500	22004.900	.050	.100
1573.422000	73058.400	.050	.033
30256.320000	1404885.000	.050	.020
2217.097000	102946.000	.033	.100
1249.511000	58018.270	.033	.033
26320.390000	1222129.000	.033	.020

The dissipation and dispersion curves for the 3D scheme are shown in Figures 13 and 14.

4E SCHEME

Scheme:

$$\begin{aligned}\beta_- &= (-c^4 + c^2 - 2\Gamma - 12\Gamma^2 + 4\Gamma c^2) / (2c^3 - 2c + 24\Gamma^2/c) \\ a_- &= (c^4 - c^2 - 2\Gamma + 12\Gamma^2 + 4\Gamma c^2) / (2c^3 - 2c + 24\Gamma^2/c) \\ \beta_+ &= -(1/3 + 2\Gamma - c^2/3) + [(c^6 + c^4(-1 - 6\Gamma) + c^2(20\Gamma^2 + 4\Gamma) \\ &\quad - 4\Gamma^2 - 24\Gamma^3) / (2c^4 - 2c^2 + 24\Gamma^2)] \\ \alpha_+ &= (-1/3 - 4\Gamma + 4c^2/3) + [(c^6 + c^4(1 + 2\Gamma) + c^2(-4\Gamma^2) \\ &\quad - 4\Gamma^2 - 24\Gamma^3) / (2c^4 - 2c^2 + 24\Gamma^2)] .\end{aligned}$$

Fourth order accurate

Stability:

Von Neumann stability:

$$\Gamma c^{10} - 2\Gamma c^8 + c^6 (24\Gamma^3 + \Gamma + 1) + c^2 (-432\Gamma^5 - 12\Gamma^3) + 288\Gamma^5 \geq 0 .$$

Error norm:

<u>Norm</u>	<u>Percent error</u>	<u>dx</u>	<u>dt</u>
0.181656	8.435	.100	.100
0.036433	1.692	.100	.033
0.037924	1.761	.100	.020
0.076683	3.561	.050	.100
0.019678	0.914	.050	.033
0.020786	0.965	.050	.020
0.031737	1.474	.033	.100
0.013327	0.619	.033	.033
0.014140	0.657	.033	.020

The dissipation and dispersion curves for the 4E scheme are shown in Figures 15 and 16.

CONCLUSION

The primary objective of this thesis is to present new schemes for the Convective Diffusive Equation, equation (1). Schemes were taken to be of the two-level, six-point form; the general two-level, six-point scheme is given as equation (6). New schemes are generated through the modified equation approach of Warming and Hyett. This method generated relations which the weighting parameters must satisfy. One relation is determined for each order of accuracy desired, up to fourth. Fourth-order accuracy is the highest accuracy possible for a six-point scheme; there is one unique fourth-order scheme. The fourth-order scheme is determined, and is presented here as equation (11). There is an infinite number of possible choices for schemes of first through third order accuracy.

After generation, the stability of the potential schemes are examined. There are several different methods of determining the stability of a given numerical scheme. Some method of determining stability are difficult to apply to a numerical scheme, and all make different assumptions about the system under observation, causing inconsistent results between methods. The two methods utilized in this thesis are the von Neumann method and the Matrix method. The von Neumann method is the most straightforward method of determining stability; however, it does not take into account boundary conditions. A general expression for the von Neumann stability of the two-level, six-point scheme is presented (equation (25)). Using this expression, any number of unconditionally stable second-order or third-order schemes can be determined. Previous to this work, only one unconditionally stable scheme had been developed, the Crank-Nicholson scheme. The Matrix method is difficult to apply to the general implicit

scheme, and so was only applied to Pade and explicit schemes. The Matrix method is more difficult to apply, but has the advantage of including the effects of the boundary conditions directly. The Matrix method stability of the general four-point explicit scheme is presented, as well as the Matrix method stability of the Crank-Nicholson scheme. The Matrix method stability of the general implicit scheme could not be determined.

Following stability, the dispersion, the effect of a scheme on the Fourier modes of a solution, was determined for the general case. The general expressions for the numerical dissipation and numerical diffusion are presented; these are equations (38) and (41), respectively. The dispersion of a scheme is critical to its actual accuracy.

After the analytic examination of the schemes, numerical tests were run to check the actual quality of the schemes. The schemes were run against an analytically solvable test case. The accuracy of both explicit schemes and implicit schemes was estimated. Accuracy was estimated by comparing the predicted values against the actual solution to the test case. The stability of explicit schemes were checked against the predicted stability. Stability was tested by running the scheme for a long time, so that the actual solution should be zero, and then checking to see if there was a non-zero solution found by the scheme. If so, the scheme was considered to be unstable. Implicit schemes were not tested for stability, but for dispersion. The diffusion and dissipation for each scheme was measured at a selection of step sizes, and wavenumbers. The stability of the explicit schemes tested is shown to correspond closely to the matrix method stability determined analytically. The numerical results for a selection of the new schemes developed, along with the best-known of previous schemes, are presented.

BIBLIOGRAPHY

- 1) B. J. Noye and H. H. Tan, A Third-Order Semi-Implicit Finite Difference Method for Solving the One-Dimensional Convection-Diffusion Equation, *International Journal for Numerical Methods in Engineering*, **26**, 1615-1629 (1988).
- 2) R. F. Warming and B. J. Hyett, "The Modified Equation Approach to the Stability and Accuracy Analysis of Finite-Difference Methods", *Journal of Computational Physics*, **14**, 159-179 (1974).
- 3) E. Huntley, Accurate Finite Difference Approximations for the Solution of Parabolic Partial Differential Equations by Semi-Discretization, *Int. Jour. for Num. Methods in Eng.* **23**, 2325-2336 (1986).
- 4) R. Peyret and T. Taylor, *Computational Methods for Fluid Flow*, Springer-Verlag, 1986.
- 5) S. Paolucci and D. R. Chenoweth, Stability of the Explicit Finite Differenced Transport Equation, *Journal of Computational Physics* **47**, 489-496 (1982).
- 6) D. A. Anderson, John C. Tannehill, and Richard H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Washington: Hemisphere Pub. Corp., New York: McGraw-Hill, 1984.
- 7) G. D. Smith, *Numerical Solutions of Partial Differential Equations: Finite Difference Methods*, third edition, Oxford University Press (1985).

Table 1

This table shows the generation of the MEPDE. Each column holds one term of each equation. The top row lists the derivative associated with the terms in that column. The row between the lines are the terms of the original equation found by Taylor expansion. The column under "operation" shows what multiplication and derivation of the original equation is being taken in order to generate the next row of the table. The MEPDE is then found by summing each column of the table. The last four multiplicative factors are listed below, as the variables η_1 through η_4 :

$$\eta_1 = h^2 (\alpha_-^2 + \alpha_- \beta_- - \beta_-^2)/3 - h^2 (\alpha_+ + \beta_+)/4$$

$$\eta_2 = -k^2 h \beta_- /12$$

$$\eta_3 = kh^2 (\alpha_+ + 2\beta_+ - \alpha_-^2 - 2\alpha_- \beta_- + \beta_-^2)/12$$

$$\eta_4 = h^3 (4\alpha_+ \alpha_- + 2\alpha_+ \beta_- + 2\beta_+ \alpha_- + 4\beta_+ \beta_- - 3\alpha_-^3 - 3\alpha_-^2 \beta_- + \alpha_- \beta_-^2 - \alpha_- - \beta_- + \beta_-^3) .$$

Operation	$\partial T / \partial t$	$\partial T / \partial x$	$\partial^2 T / \partial t^2$
	k	$h(\alpha_- - \beta_-)$	$k^2/2$
$-k/2 \partial / \partial t$	0	0	$-k^2/2$
$-h(\alpha_- + \beta_-)/2 \partial / \partial x$	0	0	0
$k^2/12 \partial^2 / \partial t^2$	0	0	0
$hk(\alpha_-/2 + \beta_-)/3 \partial^2 / \partial x \partial t$	0	0	0
$h^2[3\beta_+/2 + \beta_-^2 - 2\alpha_- (\alpha_- + \beta_-)] \partial^2 / \partial x^2$	0	0	0
$\eta_2 \partial^3 / \partial t^2 \partial x$	0	0	0
$\eta_3 \partial^3 / \partial t \partial x^2$	0	0	0
$\eta_4 \partial^3 / \partial x^3$	0	0	0

$\partial^2 T / \partial x \partial t$	$\partial^2 T / \partial x^2$	$\partial^3 T / \partial t^3$
$\alpha_- k \hbar$	$(\hbar^2 / 2) (\alpha_+ - \beta_+)$	$k^3 / 3!$
$-k \hbar (\alpha_- + \beta_-) / 2$	$-\hbar^2 (\alpha_-^2 - \beta_-^2) / 2$	0
0	0	$k^3 / 12$
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

$\partial^3 T / \partial t^2 \partial x$	$\partial^3 T / \partial t \partial x^2$	$\partial^3 T / \partial x^3$
$\alpha_- h k^2 / 2$	$\alpha_+ k h^2 / 2$	$h^3 (\alpha_- - \beta_-) / 3!$
$-\alpha_- h k^2 / 2$	$-k h^2 (\alpha_+ - \beta_+) / 4$	0
$-h k^2 (\alpha_- + \beta_-) / 4$	$-k h^2 (\alpha_-^2 + \alpha_- \beta_-) / 2$	$-h^3 (\alpha_- + \beta_-) (\alpha_+ - \beta_+) / 4$
$h k^2 (\alpha_- - \beta_-) / 12$	0	0
$h k^2 (\alpha_- / 2 + \beta_-) / 3$	$h^2 k (\alpha_- / 2 + \beta_-) (\alpha_- - \beta_-) / 3$	0
0	$k \eta_1$	$h (\alpha_- - \beta_-) \eta_1$
0	0	0
0	0	0
0	0	0

$\partial^4 T / \partial t^4$	$\partial^4 T / \partial t^3 \partial x$	$\partial^4 T / \partial t^2 \partial x^2$
$k^4/4!$	$\alpha_- k^3 h/3!$	$\alpha_+ k^2 h^2/4$
$-k^4/12$	$-hk^3 \alpha_-/4$	$-k^2 h^2 \alpha_+/4$
0	$-hk^3(\alpha_- + \beta_-)/12$	$-h^2 k^2(\alpha_-^2 + \alpha_- \beta_-)/4$
$k^4/24$	$k^3 h \alpha_-/12$	$k^2 h^2(\alpha_+ - \beta_+)/24$
0	$hk^3(\alpha_-/2 + \beta_-)/6$	$k^2 h^2 \alpha_-(\alpha_-/2 + \beta_-)/3$
0	0	$k^2 \eta_1/2$
0	$k \eta_2$	$h(\alpha_- - \beta_-) \eta_2$
0	0	$k \eta_3$
0	0	0

$\partial^4 T / \partial t \partial^3 x$	$\partial^4 T / \partial x^4$
$\alpha_- k h^3 / 3!$	$h^4 (\alpha_+ - \beta_+) / 4!$
$-k h^3 (\alpha_- - \beta_-) / 12$	0
$-k h^3 (\alpha_+ \alpha_- + \alpha_+ \beta_-) / 4$	$-h^4 (\alpha_-^2 - \beta_-^2) / 12$
0	0
$h^3 k (\alpha_- / 2 + \beta_-) (\alpha_+ - \beta_+) / 6$	0
$\alpha_- h k \eta_1$	$h_2 (\alpha_+ - \beta_+) \eta_1$
0	0
$h (\alpha_- - \beta_-) \eta_3$	0
$k \eta_4$	$h (\alpha_- - \beta_-) \eta_4$

Figure 1

Fig. 1. von Neumann stability of the general second-order implicit scheme. The shaded regions indicate the regions of stability.

Figure 2

Fig. 2. Stability of the FTCS scheme. The solid line indicates numerically determined stability. The dashed line indicates the von Neumann stability. The dotted line indicates the matrix method stability.

Figure 3

Fig. 3. Stability of the Lax-Wendroff scheme. The solid line indicates numerically determined stability. The dashed line indicates the von Neumann stability. The dotted line indicates the matrix method stability.

Figure 4

Fig. 4. Stability of the FTCS family, $\rho = .2$ scheme. The solid line indicates numerically determined stability. The dashed line indicates von Neumann stability. The dotted line indicates the matrix method stability.

Figure 5

Fig. 5. Dissipation curve for the Crank-Nicholson scheme. Γ is set to 1.

Figure 6

Fig. 6. Dispersion curve for the Crank-Nicholson scheme. Γ is set to 1.

Figure 7

Fig. 7. Dissipation curve for the 2A scheme. Γ is set to 1.

Figure 8

Fig. 8. Dispersion curve for the 2A scheme. Γ is set to 1.

Figure 9

Fig. 9. Dissipation curve for the 2B scheme. Γ is set to 1.

Figure 10

Fig. 10. Dispersion curve for the 2B scheme. Γ is set to 1.

Figure 11

Fig. 11. Dissipation curve for the 2C scheme. Γ is set to 1.

Figure 12

Fig. 12. Dispersion curve for the 2C scheme. Γ is set to 1.

Figure 13

Fig. 13. Dissipation curve for the 3D scheme. Γ is set to 1.

Figure 14

Fig. 14. Dispersion curve for the 3D scheme. Γ is set to 1.

Figure 15

Fig. 15. Dissipation curve for the 4E scheme. Γ is set to 1.

Figure 16

Fig. 16. Dispersion curve for the 4E scheme. Γ is set to 1.

APPENDIX

```

200     DIM U(501,11)
210     DIM AROW(501)
212         DIM BROW(501)
214         DIM CROW(501)
216         DIM DROW(501)
220     DIM ACTUAL(11)
225 '
230     FOR RHO! = 0 TO 2 STEP .1
250         FOR L% = 10 TO 1 STEP -1
260             FOR M% = 20 TO 1 STEP -1
265 '
300                 NX% = 10 * L% ' NX IS THE STEPWIDTH, h.
310                 NT% = 10 * M% ' NT IS THE TIME STEP, k.
320                 KAPPA! = .05 ' KAPPA IS THE DIFFUSIVE PARAMETER.
321 '                 THE VELOCITY PARAMETER HAS BEEN NORMALIZED TO ONE.
325                 TMAX% = 10
330                 C! = (1!/NT%) / (1!/NX%) ' SEE LINE 321.
340                 G! = KAPPA! * (1!/NT%) / ((1!/NX%)^2)
350 '
400 '                 THIS SECTION WILL SET THE INITIAL AND BOUNDARY VALUES.
410                     FOR T% = 0 TO TMAX%
420                         U(0,T%) = 0
430                         U(NX%,T%) = 0
440                     NEXT T%
450 '
460                     FOR X% = 0 TO (NX%/2)
470                         U(X%,0) = 2! * X% / NX%
480                         U(X% + (NX%/2),0) = 2! * (1! - ((NX%/2) + X%) /
NX%)
490                     NEXT X%
500 '
590 '                 THIS SECTION DEFINES THE SCHEME IN USE.
600                     AP! = - RHO! * G!
610                     BP! = RHO! * G!
620                     AM! = (2*G! + C!^2 + AP! - BP!) / (2*C!)
630                     BM! = (2*G! - C!^2 + AP! - BP!) / (2*C!)
650 '
700 '                 THIS SECTION FORMS THE ELEMENTS OF THE TRIDAIAGONAL
705 '                 MATRIX USED TO ADVANCE THE SOLUTION.
710                     CWO! = (BP! - BM!) / 2 ' B-1
720                     CCO! = 1! - BP! ' B0
730                     CEO! = (BP! + BM!) / 2 ' B1
740                     CWN! = (AP! - AM!) / 2 ' a-1
750                     CCN! = 1! - AP! ' a0

```



```

760          CEN! = (AP! + AM!) / 2          ' a1
770 '
800 '      THIS SECTION LISTS THE ACTUAL VALUES FOR T = .1 .
811          ACTUAL(1) = .06724
812          ACTUAL(2) = .21249
813          ACTUAL(3) = .40097
814          ACTUAL(4) = .59665
815          ACTUAL(5) = .76668
816          ACTUAL(6) = .84042
817          ACTUAL(7) = .76666
818          ACTUAL(8) = .59638
819          ACTUAL(9) = .39072
820 '
1000 '      THIS LOOP ITERATES THE SCHEME.
1050          FOR Q% = 1 TO (NX% / 10)
1060              T% = 1
1070 '
1100 '      THESE LOOPS FILL THE MATRIX TO BE SOLVED.
1120          FOR X% = 1 TO (NX% - 2)
1130              AROW(X%) = CEN!
1140          NEXT X%
1150 '
1160          FOR X% = 2 TO (NX% - 1)
1170              BROW(X%) = CWN!
1180          NEXT X%
1190 '
1200          FOR X% = 1 TO (NX% - 1)
1210              DROW(X%) = CCN!
1220          NEXT X%
1230 '
1240          FOR X% = 1 TO (NX% - 1)
1250              CROW(X%) = CWO!*U(X% - 1, T% - 1) + CCO!*U(X%, T% -
1260              + CEO!*U(X% + 1, T% - 1)
1270          NEXT X%
1270 '
1400 '      THIS SECTION SOLVES THE MATRIX.
1440          NEQN% = NX% - 1
1450          FOR J% = 2 TO NEQN%
1460              R! = BROW(J%) / DROW(J% - 1)
1470              DROW(J%) = DROW(J%) - R! * AROW(J% - 1)
1480              CROW(J%) = CROW(J%) - R! * CROW(J% - 1)
1490          NEXT J%
1500 '
1510          CROW(NEQN%) = CROW(NEQN%) / DROW(NEQN%)
1520          FOR J% = (NEQN% - 1) TO 1 STEP -1
1530              CROW(J%) = (CROW(J%) -
AROW(J%)*CROW(J%+1)) / DROW(J%)

```

```

1540             NEXT J%
1550 '
1690 '         THIS SECTION READS THE NEW VALUES INTO THE ARRAY.
1700             FOR X% = 1 TO NX%
1710                 U(X%, 0) = CROW(X%)
1720             NEXT X%
1800         NEXT Q%
1820 '     THIS SECTION CALCULATES THE ERROR.
1850         ACTSUM! = 0
1860         SUMDIFF! = 0
1870         FOR X% = 1 TO 9
1880             ZDETA% = X% * NX% / 10
1890             DIFF! = ACTUAL(X%) - U(ZDETA%, 1)
1900             ACTSUM! = ACTSUM! + ACTUAL(X%)
1910             SDIFF! = DIFF!^2
1920             SUMDIFF! = SUMDIFF! + SDIFF!
1930         NEXT X%
1940         NORM! = SQR(SUMDIFF!) / NX%
1950         PERC! = NORM! / SQR(ACTSUM!)
1960 '
1980 '     THIS SECTION UPDATES THE ARRAY.
2000         FOR X% = 1 TO (NX% - 1)
2010             U(X%,1) = U(X%,0)
2020         NEXT X%
2050 '
4095         PRINT USING "NORM = #####, PERCENT ERROR = #####,
DX = .###, DT = .###"; NORM!, PERC!, 1!/NX%, 1!/NT%
10000     NEXT M%
11000     NEXT L%
12000     NEXT RHO!
15000     END

```