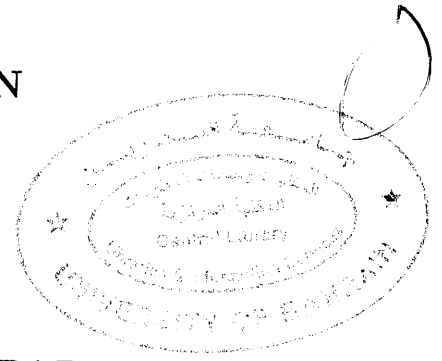


UNIVERSITY OF BAHRAIN



TIME STEPPING ALGORITHMS FOR PARABOLIC
PARTIAL DIFFERENTIAL EQUATIONS

BY

THURAYA JUMA ABDULLA

A THESIS SUBMITTED TO THE DEPARTMENT
OF MATHEMATICS AS A PARTIAL
FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
OF
MASTER OF SCIENCE

UNDER THE SUPERVISION OF

PROF. A.Q.M. KHALIQ

JUNE 1995

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ -TEX



No: 27 June 1995

الرقم:

Date:

التاريخ:

The Thesis defence committee considers the Thesis titled :

*"Time Stepping Algorithms for Parabolic
Partial Differential Equations"*

Submitted by Thuraya Juma Abdulla to the Department of Mathematics
satisfactory and acceptable for the Master of Science degree in
Mathematics

Thesis Defence Committee :

Prof. A.Q.M. Khaliq
Supervisor

Department of Mathematics
University of Bahrain

Prof. Abdulla Y. Al-Hawaj
Member

Department of Mathematics
University of Bahrain

Prof. Mustafa Amer
Member

Department of Mathamtics
Emirates University

Dr. Mohammed Abdulla Ali
Chairman

Department of Mathematics
University of Bahrain

ACKNOWLEDGMENTS

First, I would like to express my gratefulness to the God for enabling me to achieve my thesis.

Next, I express my thanks to Prof. A.Q.M. Khaliq ; my supervisor; for helping me, for his continuous support and encouragement to overcome the difficulties faced me during the preparation of the thesis.

Finally, I would like to thank my friends for encouraging me. In particular, I am in debted to Ms. Effat Assiri and Ms. Ashwaq Hashm.

ABSTRACT

A number of time stepping algorithms for solving parabolic partial differential equations (PDEs) in one and two space dimensions are studied. The parabolic partial differential equation is transformed into a system of ordinary differential equations (ODEs) *via* method of lines approach (MOL). The solution of the system of ODEs satisfies a recurrence relation which involves matrix exponential functions. Padé approximants are employed to approximate such exponential terms.

It is shown [9] that methods based on (1,1) diagonal Padé approximant suffer with time steps restriction for problems with high frequency components in the solution when a time discretization is imposed with steps that are “too large” relative to the spatial discretization. A monotonicity property is established for all diagonal Padé approximants from which it is shown that corresponding higher-order methods suffer a similar time step restriction as the (1,1) Padé. Next, various high-order methods based on subdiagonal Padé approximations are presented which, through a partial fraction expansion, are no more complicated to implement than the first order implicit Euler method based on the (1,0) Padé approximation; moreover, the resulting algorithms are free of a time step restriction intrinsic to those based on diagonal Padé approximations [7].

Splitting methods based on the (1,0) and (1,1) Padé approximants are considered in two space dimension with homogeneous and time dependent boundary conditions [9], [18].

All computations are performed on VAX machine, University of Bahrain. Packages, such as, *Mathematica* and *Spyglass* are used for symbolic computation and graphics.

TABLE OF CONTENTS

Chapter 1. Introduction	1
1.1 Partial Differential Equations	1
1.2 The Method of Lines	3
1.3 A Family Of Numerical Methods	5
1.3.1 An Explicit Method	6
1.3.2 A Fully Implicit Method	6
1.3.3 The Crank-Nicolson Method	7
1.4 Padé Approximants	7
1.5 Stability Analysis	8
1.5.1 Conventional Stability Analyses	8
1.5.2 A_0 – Stability and L_0 – Stability	11
1.6 Local Truncation Errors	12
1.7 Outline Of Thesis	14
Chapter 2. Time Stepping Algorithms For Parabolic PDEs Based on a Padé Approximations	15
2.1 Introduction	15
2.2 Time Step Restriction for (1,1) Padé Approximation	16
2.3 Factorized Padé Approximations	21
2.4 Partial Fraction Decomposition of Padé Approximations	23
2.5 The Implementation	25
2.5.1 A Second Order Method	26
2.5.2 Fourth Order Methods	29
2.6 Numerical Results	35
Chapter 3. The Extrapolation Of First Order Methods For Parabolic PDE's Based On (1,0) Padé Approximations	38
3.1 Extrapolatin Of The Fully Implicit Scheme	38
3.2 Numerical Results I	40
3.3 The Extrapolated Method In Two Space Variables	44
3.4 Numerical Results II	49
Chapter 4. Time Dependent Boundary And Interior Forcing In Locally One Dimensional methods (LOD)	57
4.1 Introduction	57
4.2 An Example Of Boundary Forcing	59
4.3 One Dimentional Approximation	62
4.4 Interior Forcing	71
4.5 Numerical Result	72
Appendix	74
References	75

CHAPTER I

INTRODUCTION

1.1 Partial Differential Equations

Second order parabolic partial differential equations (PDE's) have important applications in science, biomedical systems and engineering. For instance, they occur in the mathematical modelling of problems in heat flow in the human body, neurophysiology, genetics, reaction-diffusion, percutaneous drug absorption, etc. Consider the quasi-linear partial differential equation in two independent variables x and t of the form

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial t} + c \frac{\partial^2 u}{\partial t^2} = e \quad (1.1.1)$$

where $u = u(x, t)$ and a, b, c, e , may be functions of $x, t, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}$ but not of the second derivatives. Denoting $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial t}, \frac{\partial^2 u}{\partial t^2}$ by p, q, r, s, w respectively, (1.1.1) may be written as

$$ar + bs + cw = e. \quad (1.1.2)$$

Suppose that the values of u, p, q at all points (x, t) on some smooth curve in the (x, t) plane are known, then these values must satisfy the total differential formula

$$du = \frac{\partial u}{\partial x} dx + \frac{\partial u}{\partial t} dt = p dx + q dt, \quad (1.1.3)$$

similarly, p and q satisfy the formula

$$\begin{aligned} dp &= \frac{\partial p}{\partial x} dx + \frac{\partial p}{\partial t} dt = r dx + s dt, \\ dq &= \frac{\partial q}{\partial x} dx + \frac{\partial q}{\partial t} dt = s dx + w dt. \end{aligned} \quad (1.1.4)$$

Equations (1.1.2) and (1.1.4) are combined in a matrix-vector form as follows

$$\begin{bmatrix} a & b & c \\ dx & dt & 0 \\ 0 & dx & dt \end{bmatrix} \begin{bmatrix} r \\ s \\ w \end{bmatrix} = \begin{bmatrix} e \\ dp \\ dq \end{bmatrix} \quad (1.1.5)$$

which will not have a unique solution at any point at which the determinant of the coefficients of r, s, w vanish. That is, whenever,

$$\begin{vmatrix} a & b & c \\ dx & dt & 0 \\ 0 & dx & dt \end{vmatrix} = 0 \quad (1.1.6)$$

holds. The determinant of (1.1.6) is

$$a\left(\frac{dt}{dx}\right)^2 - b\left(\frac{dt}{dx}\right) + c = 0 \quad (1.1.7)$$

which has two real distinct, one real repeated, or conjugate complex roots in $\frac{dt}{dx}$ according to the discriminant $b^2 - 4ac$ is greater than, equal to, or less than zero. In a domain in the xt -plane in which u, p, q are defined, and throughout which equation (1.1.7) has two, one, or no real roots, the partial differential equation (1.1.1) is said to be hyperbolic, parabolic or elliptic, respectively. For example, the partial differential equation

$$\frac{\partial^2 u}{\partial x^2} + u \frac{\partial^2 u}{\partial x \partial y} + (1 + u^2) \frac{\partial^2 u}{\partial y^2} = 0 \quad (1.1.8)$$

is elliptic as $a = 1, b = u, c = 1 + u^2, e = 0$ and $b^2 - 4ac = -(4 + 3u^2) < 0$. Similarly, the partial differential equation

$$\frac{\partial^2 u}{\partial x^2} + 2 \frac{\partial^2 u}{\partial x \partial t} + (1 - x^2 - t^2) \frac{\partial^2 u}{\partial t^2} = x^2 + t^2 \quad (1.1.9)$$

is hyperbolic since $a = 1, b = 2, c = 1 - x^2 - t^2, e = x^2 + t^2$ and $b^2 - 4ac = 4(x^2 + t^2) > 0$ except at the point $(0, 0)$ where it is parabolic. The most widely used parabolic equation is obtained by writing $a = 1, b = c = 0$ and $e = \frac{\partial u}{\partial t}$ in (1.1.1) to give

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad (1.1.10)$$

in some region Ω . Equation (1.1.10) is called the one dimensional heat equation . Writing $a = \frac{1}{Re}, b = c = 0$ and $e = \frac{\partial u}{\partial t} + \psi \frac{\partial u}{\partial x}$ gives the one dimensional diffusion-convection equation

$$\frac{\partial u}{\partial t} + \psi \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} \quad (1.1.11)$$

in which Re is the Reynolds number and $\psi > 0$ is the convection coefficient. Equation (1.1.1) is said to be linear if a, b, c, e do not depend on u , otherwise it is nonlinear. When a, b, c, e depend on x and/or t only, (1.1.1) is said to be of the linear, variable coefficient type and when a, b, c, e are constants, (1.1.1) is termed a constant coefficient partial differential equation. The heat equation

$$\beta(x, t) \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[\alpha(x, t) \frac{\partial u}{\partial x} \right] - \gamma(x, t) u \quad (1.1.12)$$

is an example of self-adjoint parabolic partial differential equation whenever α is a function of x only and β is a constant.

1.2 The Method of Lines

The method of lines, MOL, is an approach to the numerical solution of quite general partial differential equations, PDEs, that involve a time variable t and one or more space variables $x, y, \dots$. The partial derivatives with respect to the space variables are discretized to result in an approximating system of ODEs in the variable t . Such a semi-discretization is often an intermediate step in the derivation of a fully discrete scheme, but in the MOL approach, the ODEs are integrated directly with a standard code for the task. The convenience and power of readily available ODE solvers permits complicated PDEs to be solved with comparative ease in this manner, especially when there is only one space variable. It might well be more efficient to solve a given PDE by a method particularly suited to the equation, but the easy solution of rather problems with acceptable efficiency makes the approach attractive. (see, Shampine [16])

Consider the one dimensional heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad ; \quad X_0 < x < X_1, \quad t > 0 \quad (1.2.1)$$

together with the boundary conditions

$$u(X_0, t) = u(X_1, t) = 0, \quad t > 0, \quad (1.2.2)$$

and initial conditions

$$u(x, 0) = g(x) \quad ; \quad X_0 \leq x \leq X_1, \quad (1.2.3)$$

where $g(x)$ is a given continuous function of x and it is not specified that $g(X_0) = 0$ or $g(X_1) = 0$ so that discontinuities between initial conditions and boundary conditions may occur. The interval $X_0 \leq x \leq X_1$ is divided into $N+1$ subintervals each of width h so that $(N+1)h = X_1 - X_0$ and the time variable t is incremented in steps of length τ . The open region Ω and its boundary $\partial\Omega$ consisting of the lines $x = X_0$, $x = X_1$ and the axis $t = 0$ have thus been covered by a rectangular mesh, the mesh points having coordinates $x_m = X_0 + mh$ and $t_n = n\tau$ where $m = 0, 1, \dots, N+1$ and $n = 0, 1, 2, \dots$.

The space derivative in (1.2.1) is replaced by its second-order central difference approximation

$$\frac{\partial^2 u}{\partial x^2} = h^{-2} [u(x-h, t) - 2u(x, t) + u(x+h, t)] + O(h^2), \quad (1.2.4)$$

and (1.2.1) together with the boundary condition (1.2.2) are applied to all N interior mesh points at time level $t = n\tau$ ($n = 0, 1, 2, \dots$) to give

$$\begin{aligned} \frac{dU_1}{dt} &= h^{-2} [U_0(t) - 2U_1(t) + U_2(t)] \\ \frac{dU_2}{dt} &= h^{-2} [U_1(t) - 2U_2(t) + U_3(t)] \\ \frac{dU_3}{dt} &= h^{-2} [U_2(t) - 2U_3(t) + U_4(t)] \\ &\vdots \\ \frac{dU_m}{dt} &= h^{-2} [U_{m-1}(t) - 2U_m(t) + U_{m+1}(t)] \\ &\vdots \\ \frac{dU_{N-1}}{dt} &= h^{-2} [U_{N-2}(t) - 2U_{N-1}(t) + U_N(t)] \\ \frac{dU_N}{dt} &= h^{-2} [U_{N-1}(t) - 2U_N(t) + U_{N+1}(t)] \end{aligned} \quad (1.2.5)$$

where $U_m(t)$ is the (semi-discrete) solution of $\{(1.2.1), (1.2.2), (1.2.3)\}$ when $x = x_m$ for any $t > 0$. Equations (1.2.5) may be written in the form

$$\frac{dU(t)}{dt} = AU(t) \quad ; \quad U(0) = g, \quad (1.2.6)$$

where $U(t) = [U_1(t), U_2(t), \dots, U_N(t)]^T$, T denoting transpose, is the vector of approximations to the solution of the PDE (1.2.1) at $t = t_n = n\tau$ and g is the vector of initial conditions. The matrix A is tridiagonal of order N and is given by

$$A = h^{-2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{bmatrix}; \quad (1.2.7)$$

it has real and negative eigenvalues given by

$$\lambda_i = \frac{-4}{h^2} \sin^2 \left[\frac{i\pi}{2(N+1)} \right], \quad i = 1, 2, \dots, N. \quad (1.2.7')$$

This method of transforming the initial/boundary-value problem $\{(1.2.1), (1.2.2), (1.2.3)\}$ into the initial value problem (1.2.6) is known as the *Method of Lines*.

The solution of the system of first-order, linear ordinary differential equations (1.2.6) is

$$U(t) = \exp(tA)g \quad (1.2.8)$$

which satisfies the recurrence relation

$$U(t + \tau) = \exp(\tau A)U(t) \quad ; \quad t = 0, \tau, 2\tau, 3\tau, \dots \quad (1.2.9)$$

1.3 A Family of Numerical Methods

The development of this family of numerical methods will be based on making rational approximations to the matrix exponential term in (1.2.9). A commonly used family of approximants is

$$\exp(\tau A) \approx [I - \theta \tau A]^{-1} [I + (1 - \theta) \tau A] \quad (1.3.1)$$

in which $0 \leq \theta \leq 1$ is a parameter. Putting (1.3.1) in (1.2.9) leads to the family of methods

$$(I - \theta \tau A)U(t + \tau) = [I + (1 - \theta) \tau A]U(t). \quad (1.3.2)$$

The three well known methods of solving (1.2.1) with (1.2.2) and (1.2.3) are obtained by setting $\theta = 0, 1, \frac{1}{2}$.

1.3.1 An explicit method

Writing $\theta = 0$ in (1.3.2) gives

$$U(t + \tau) = (I + \tau A)U(t) \quad (1.3.3)$$

which, when applied to the general mesh point (x_m, t_n) gives the four-point, two-time level explicit scheme

$$U_m^{n+1} = pU_{m-1}^n + (1 - 2p)U_m^n + pU_{m+1}^n \quad (1.3.4)$$

where $p = \frac{\tau}{h^2}$, $m = 1, 2, \dots, N$ and $n = 0, 1, \dots$, and U_m^n is the fully discrete solution of $\{(1.2.1), (1.2.2), (1.2.3)\}$ at the general mesh point. Clearly, U_m^n is an approximation to $u_m^n = u(x_m, t_n)$, the analytical solution of $\{(1.2.1), (1.2.2), (1.2.3)\}$.

1.3.2 A fully implicit method

Writing, next, $\theta = 1$ in (1.3.2) gives

$$(I - \tau A)U(t + \tau) = U(t) \quad (1.3.5)$$

which, when applied to the general mesh point (x_m, t_n) , gives the four-point, two-time level, fully implicit scheme

$$-pU_{m-1}^{n+1} + (1 + 2p)U_m^{n+1} - pU_{m+1}^{n+1} = U_m^n. \quad (1.3.6)$$

This method requires the application of a tridiagonal solver at each time step.

1.3.3 The Crank-Nicolson method

Putting $\theta = \frac{1}{2}$ in (1.3.2) gives the well-known method

$$(I - \frac{1}{2}\tau A)U(t + \tau) = (I + \frac{1}{2}\tau A)U(t) \quad (1.3.7)$$

which, on application to the general mesh point (x_m, t_n) leads to the six-point, two-time level, implicit scheme

$$-\frac{p}{2}U_{m-1}^{n+1} + (1+p)U_m^{n+1} - \frac{p}{2}U_{m+1}^{n+1} = \frac{p}{2}U_{m-1}^n + (1-p)U_m^n + \frac{p}{2}U_{m+1}^n. \quad (1.3.8)$$

This formula is known as the *Crank-Nicolson method* (Crank and Nicolson, [1]). It is unconditionally stable but the restriction (see chapter 2)

$$r = \frac{\tau}{h} < \frac{X_1 - X_0}{\pi} \quad (1.3.9)$$

must be imposed for problems with discontinuities between initial and boundary conditions.

1.4 Padé Approximants

The methods used in this thesis are mainly based on the use of Padé approximants of the matrix-exponential in the recurrence relation. The Padé approximants (Padé [12]) are now defined.

A Padé approximant $R_{m,k}(z)$ to an analytic function $f(z)$ in a region of the complex plane containing the origin $z = 0$ is given by

$$R_{m,k}(z) = \frac{P_k(z)}{Q_m(z)} \quad (1.4.1)$$

(Graves-Morris, [3]) where $P_k(z)$ and $Q_m(z)$ are polynomials in z of degrees k and m respectively having leading coefficient unity, and k, m are nonnegative integers. For each pair of m and k , $P_k(z)$ and $Q_m(z)$ are those polynomials for which the Taylor series expansion of $R_{m,k}(z)$ about the origin agrees with the Taylor series of $f(z)$ for as many terms as possible. The ratio (1.4.1) contains essentially $m + k + 1$ unknown coefficients, therefore the requirement

$$Q_m(z)f(z) - P_k(z) = O(|z|^{m+k+1}), \quad |z| \rightarrow 0 \quad (1.4.2)$$

results in $m + k + 1$ linear equations for the coefficients. For $f(z) = \exp(z)$, the entries in the Padé Table are given explicitly by Varga [21] as

$$P_k(z) = \sum_{j=0}^m \frac{(m+k-j)!m!}{(m+k)!j!(m-j)!} z^j \quad (1.4.3)$$

and

$$Q_m(z) = \sum_{j=0}^k \frac{(m+k-j)!k!}{(m+k)!j!(k-j)!} (-z)^j \quad (1.4.4)$$

and if

$$\exp(z) = \frac{P_k(z)}{Q_m(z)} + R_{m,k}^*(z) \quad (1.4.5)$$

then the remainder $R_{m,k}^*(z)$ is given by

$$R_{m,k}^*(z) = \frac{(-1)^{k+1} z^{(m+k+1)}}{(m+k)!Q_m(z)} \int_0^1 \exp(z(1-u)) u^k (1-u)^m du. \quad (1.4.6)$$

The first twenty-four entries of the Padé Table for $f(z) = \exp(z)$ are given in the appendix.

Some Properties of the Padé approximants are given by Lambert [8] as follows :

“let $R_{m,k}(z)$ be the (m,k) Padé approximant to $\exp(z)$, then $R_{m,k}(z)$ is A – acceptable if $m = k$,

A_0 – acceptable if $m \geq k$,

L – acceptable if $m = k + 1$ or $m = k + 2$ ”

The region of acceptability of $R_{m,k}(z)$ is that area of the complex plane within which the approximant $R_{m,k}(z)$ satisfies

$$|R_{m,k}(z)| < 1. \quad (1.4.7)$$

1.5 Stability Analyses

1.5.1 Conventional stability analyses

The concept of stability is concerned with the boundedness of the solution of the finite difference equations and is examined by

finding conditions under which the difference between the analytical and numerical solution of the difference equation given at the mesh point (x_m, t_n) by

$$Z_m^n = U_m^n - U_m^{*n} \quad (1.5.1)$$

remains bounded as n increases, for fixed h and τ . In (1.5.1), U_m^{*n} denotes the actual numerical solution obtained at the general mesh point (x_m, t_n) . It is an approximation to U_m^n which incorporates computer round-off errors. A stability analysis considers the growth of a perturbation in initial data or the growth of errors introduced at the mesh points at a given time level. There are three commonly used methods to investigate stability, namely the energy method, the matrix method, and the von-Neumann or Fourier method.

The energy method is mainly applicable to most initial/boundary-value problems including variable coefficient problems and nonlinear problems. Its successful application depends largely on the ingenuity of the user because of its complicated nature. But despite proving the stability of numerical methods, the energy methods indicate the correct choice of a method. The method, however, provides only sufficient conditions for stability which may be far removed from what is necessary in certain initial/boundary-value problem. The method entails calculating the squares of the errors Z_m^n at a time level $t = n\tau$ called the energy (hence the name of the method). The method determines criteria under which the energy (not the physical energy of the system) remains bounded as the computation proceeds.

The matrix method is applicable to initial-value problems. A necessary and sufficient condition for stability is

$$\max |\lambda_s| \leq 1 \quad , \quad s = 1, 2, \dots, N \quad (1.5.2)$$

where λ_s are the distinct eigenvalues of the matrix A in (1.2.6). For more details on this method see for example Smith [17] or Mitchell and Griffiths [10].

The von-Neumann method, developed by J. von Neumann in the early 1940's and first discussed by O'Brien et al. [11], provides a simple necessary condition for numerical stability which essentially depends on the uniform boundedness of the Fourier coefficients of the solution

of the difference equation. It is assumed that there exist harmonic decompositions of the grid functions u_k^0 of the initial time level

$$u_k^0 = \sum_j A_j \exp(i\beta_j x_k) \quad , \quad (1.5.3)$$

where $i = \sqrt{-1}$, the frequencies β_j are, in general use, arbitrary and a uniform grid is used. It is only necessary to consider the single term $\exp(i\beta x)$ where β may be investigated by finding a solution of the difference equation which reduce to $\exp(i\beta x)$ when $t = 0$. Such a solution is

$$\exp(\alpha t) \exp(i\beta x) \quad (1.5.4)$$

where $\alpha = \alpha(\beta)$ is, in general, complex. The original grid function $\exp(i\beta x)$ will not grow with time if

$$|\exp(\alpha \tau)| \leq 1 \quad (1.5.5)$$

Equation (1.5.5) is called the *von-Neumann criterion for stability*, sometimes called the *von-Neumann necessary condition for stability*.

It is noted, however, that von-Neumann's method applies only to linear difference equations with constant coefficients, and to be more precise only to initial-value problems with periodic data. If the difference scheme has variable coefficients the method can still be applied at every mesh point, individually, of the mesh (this is known as examining the local stability) and it is reasonable to assume that the difference scheme is stable globally if it is stable at every mesh point. The criterion (1.5.5) is necessary and sufficient for two time level difference equations but is not always sufficient for three or more level equations although it is always necessary. This also applies to problems with more than one space variable.

Since Z_m^n satisfies the original difference equation, the von-Neumann analysis may be applied using Z_m^n in place of U_m^n . Thus the criterion (1.5.5) ensures that the rounding errors introduced will not grow as the computation proceeds in time.

For example, consider the explicit scheme (1.3.4), the computed solution satisfies

$$U_m^{n+1} = pU_{m-1}^n + (1 - 2p)U_m^n + pU_{m+1}^n \quad , \quad (1.5.6)$$

and the error

$$Z_m^n = U_m^n - U_m^{*n} = \exp(\alpha n \tau) \exp(i \beta m h) , \quad (1.5.7)$$

satisfies

$$Z_m^{n+1} = p Z_{m-1}^n + (1 - 2p) Z_m^n + p Z_{m+1}^n \quad (1.5.8)$$

which may be simplified to

$$\zeta = \exp(\alpha \tau) = 1 - 4p \sin^2 \frac{1}{2} \beta h. \quad (1.5.9)$$

the von-Neumann stability requirement $|\exp(\alpha \tau)| \leq 1$ leads to

$$-1 \leq 1 - 4p \sin^2 \frac{1}{2} \beta h \leq 1 \quad (1.5.10)$$

which is satisfied by

$$0 < p \leq \frac{1}{2}. \quad (1.5.11)$$

In (1.5.9) ζ is called the *amplification symbol* of the numerical method.

Similarly, the amplification symbols of the fully implicit method (1.3.6) and the Crank-Nicolson method (1.3.8) are

$$\zeta = \frac{1}{1 + 4p \sin^2 \frac{1}{2} \beta h} \quad (1.5.12)$$

and,

$$\zeta = \frac{1 - 2p \sin^2 \frac{1}{2} \beta h}{1 + 2p \sin^2 \frac{1}{2} \beta h}, \quad (1.5.13)$$

respectively.

1.5.2 A_0 – stability and L_0 – stability

Each of the numerical methods (1.3.3), (1.3.5) and (1.3.7) is of the form

$$U(t + \tau) = R(\tau A) U(t) \quad ; \quad t = 0, \tau, 2\tau, \dots \quad (1.5.14)$$

where $R(\tau A)$ is the *amplification matrix*. For example, the amplification matrices for the explicit, fully implicit and Crank-Nicolson methods are $(I + \tau A)$, $(I - \tau A)^{-1}$ and $(I - \frac{1}{2} \tau A)^{-1} (I + \frac{1}{2} \tau A)$, respectively.

Definition 1.1 If $U(t + \tau) = R(\tau A)U(t)$, where A is a symmetric matrix of order N with negative, real eigenvalues $(\lambda_s, s = 1, 2, \dots, N)$, then the resulting numerical method is said to be A_0 – stable if

$$\| R(\tau A) \| \leq 1 \quad \text{or} \quad |R(\tau \lambda_s)| \leq 1 \quad \text{or} \quad |R(-z_s)| \leq 1 \quad (1.5.15)$$

where $R(\tau \lambda_s)$ or $R(-z_s)$ is the *amplification symbol* of the numerical method and $z_s = -\tau \lambda_s > 0$.

It follows that A_0 – stability guarantees unconditional stability in the von-Neumann sense and absolute stability in the terminology of ordinary differential equations (Lambert, [8])

Definition 1.2 An A_0 – stable method for which, additionally,

$$\lim_{z \rightarrow \infty} R(-z) = 0 \quad (1.5.16)$$

is called an L_0 – stable method.

An L_0 – stable method has the advantage of damping all components of the numerical solution to zero as $t \rightarrow \infty$. It can therefore be seen that the Crank-Nicolson method (1.3.8) is A_0 – stable while the fully implicit scheme (1.3.6) is L_0 – stable. For an A_0 – stable method such as the Crank-Nicolson method, discontinuities between initial and boundary conditions are propagated as $t \rightarrow \infty$ but for an L_0 – stable method they are not.

1.6 Local Truncation Errors

The *local truncation error* of a finite difference scheme at a mesh point $(mh, n\tau)$ is the difference between the finite difference scheme and the differential equation it is replacing. The difference between the analytical solution of the differential and difference equations is called the local discretization error and is represented at the mesh point $(mh, n\tau)$ by $u_m^n - U_m^n$ where $u_m^n = u(mh, n\tau)$ is the analytical solution of the partial differential equation.

A difference equation is said to be consistent if the local truncation error tends to zero as the mesh is refined, that is $h, \tau \rightarrow 0$

simultaneously. For example, consider the general finite difference method arising from the family (1.3.2). It can be seen that the method takes the form

$$\begin{aligned} \frac{U_m^{n+1} - U_m^n}{\tau} = h^{-2} [& \theta(U_{m-1}^{n+1} - 2U_m^{n+1} + U_{m+1}^{n+1}) \\ & + (1 - \theta)(U_{m-1}^n - 2U_m^n + U_{m+1}^n)] \end{aligned} \quad (1.6.1)$$

from which it follows that the left hand side approximates $\frac{\partial u}{\partial t}$ while the right hand side gives a weighted approximation to $\frac{\partial^2 u}{\partial x^2}$. The local truncation error of (1.6.1) is given by

$$\begin{aligned} L(x, t) = & \frac{u(x, t + \tau) - u(x, t)}{\tau} \\ & - \frac{\theta [u(x - h, t + \tau) - 2u(x, t + \tau) + u(x + h, t + \tau)]}{h^2} \\ & - \frac{(1 - \theta) [u(x - h, t) - 2u(x, t) + u(x + h, t)]}{h^2} \\ & - \left(\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} \right). \end{aligned} \quad (1.6.2)$$

The Taylor series expansion of (1.6.2) about $u(x, t)$ leads to

$$L(x, t) = \left(-\frac{1}{12} h^2 \frac{\partial^4 u}{\partial x^4} + C_q \tau \frac{\partial^2 u}{\partial t^2} \right) + C_{q+1} \tau^2 \frac{\partial^3 u}{\partial t^3} + \dots \quad (1.6.3)$$

where $C_q = C_2 = \frac{1}{2} - \theta$ and $C_{q+1} = C_3 = \frac{1}{6} - \frac{\theta}{2}$. Note that $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \Rightarrow \frac{\partial^2 u}{\partial t^2} = \frac{\partial^3 u}{\partial x^2 \partial t}$ and $\frac{\partial^3 u}{\partial t^3} = \frac{\partial^4 u}{\partial x^2 \partial t^2}$.

The term in the parentheses in (1.6.3) is called the *principal part* of the local truncation error. The component $-\frac{1}{12} h^2 \frac{\partial^4 u}{\partial x^4}$ is due to the space discretization and use of (1.2.4) in (1.2.1). This term will always appear in the principal part of the local truncation error of every method arising from (1.3.2) in which $0 \leq \theta \leq 1$.

The term C_q in (1.6.3) is the *time error constant* of the method. For the explicit scheme (1.3.4), θ is set to zero giving the time error constant $C_2 = \frac{1}{2}$ while for the fully implicit scheme (1.3.6) $\theta = 1$ giving $C_2 = -\frac{1}{2}$; thus the two methods are both first-order accurate in time. As for the Crank-Nicolson method (1.3.8), $\theta = \frac{1}{2}$ causing

C_2 to vanish and the time error constant is consequently given by $C_3 = \frac{1}{12}$. Thus the Crank-Nicolson method is second-order accurate in time and therefore more accurate. From (1.6.3) it can easily be seen that $L(x, t) \rightarrow 0$ as $h, \tau \rightarrow 0$ simultaneously therefore every finite difference method arising from (1.3.2) with $0 \leq \theta \leq 1$ is consistent with the partial differential equation (1.2.1).

1.7 Outline of Thesis

In Chapter 2, factorization and partial fraction decomposition of higher order Padé approximants are studied. Time stepping algorithms in serial and parallel forms are obtained. In Chapter 3 the extrapolation of first order methods for parabolic PDEs based on (1,0) Padé approximations in one and two space is analysed. In Chapter 4 the numerical solution of parabolic PDEs with time dependent boundary and interior forcing by locally one dimensional schemes (LOD) is considered.

CHAPTER II

TIME STEPPING ALGORITHMS FOR PARABOLIC PDEs BASED ON PADÉ APPROXIMATIONS

2.1 Introduction

In recent years much attention has been given in literature to the development, analysis and implementation of L_0 -stable methods for the numerical solution of second-order parabolic partial differential equations for which A_0 -stable methods such as the Crank-Nicolson method are unsatisfactory when a time discretization is imposed with time steps that are too large relative to the spatial discretization. For example, Lawson and Morris [9] developed a second-order L_0 -stable method as an extrapolation of a first-order backward difference method in one and two space dimensions (see chapter 3). This idea was developed further for one space variable by Gourlay and Morris [4] achieving third and fourth order accuracy in time *via* a multistage process. Most recently, Khaliq *et al.* [7] developed some parallel algorithms for semidiscretized parabolic partial differential equations based on higher-order subdiagonal Padé approximations. By establishing a monotonicity property on all the diagonal Padé approximants they showed that higher-order diagonal Padé approximants suffer similar time step restrictions as the second-order diagonal (1,1) Padé approximant (the Crank-Nicolson method) given by (1.3.9).

In this chapter, higher-order methods (in time) based on Padé approximants are considered. As in chapter 1, the method of lines is used to transform the parabolic partial differential equation into a system of first-order, linear, ordinary differential equations (ODEs) the solution of which satisfies a certain recurrence relation (Lawson and Morris, [9]) involving a matrix exponential term. The development of numerical methods will be based on making

rational approximations using higher-order diagonal Padé approximants to the exponential term. The resulting finite difference schemes are implicit in nature and require the application of higher-diagonal solvers (five, seven, nine, ect) to determine the solution, but the parallel and sequential methods of the same schemes obtained *via* the partial fraction decomposition technique need only the application of tridiagonal solvers with complex arithmetic [6],[7].

In section 2.2 a necessary constraint on the time step for the method based on the (1,1) Padé approximant is given. In section 2.3 higher order Padé approximants are presented in factored form along with time step restriction on them. Partial fraction decomposition is given in section 2.4 and their parallel implementation is presented in section 2.5. Numerical results are presented in 2.6.

2.2 Time Step Restriction for (1,1) Padé Approximation

Lawson and Morris [9], have shown that some well known methods which are based on the second order diagonal (1,1) Padé approximation (Crank-Nicolson) behave poorly on such problems when the time discretization is assumed to be too large relative to the space discretization.

Consider a constant coefficient heat equation in one space variable

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \{x \mid 0 < x < X\} \times \{t \mid t > 0\}, \quad (2.2.1)$$

with boundary conditions

$$u(0, t) = u(X, t) = 0,$$

and initial conditions

$$u(x, 0) = g(x),$$

where $g(x)$ is a given continuous function of x . It is not specified that $g(0) = 0$ or $g(X) = 0$, so that discontinuities between initial conditions and boundary conditions may exist.

Subdivid the interval $0 \leq x \leq X$ into $N + 1$ equal subintervals by the grid lines $x_i = ih, i = 0, \dots, N$ where $(N + 1)h = X$. Then

the space derivative in (2.2.1) is replaced by the central difference operator (1.2.4) so that (2.2.1) becomes

$$\frac{du}{dt} = \frac{\delta_x^2}{h^2} u(x, t) + O(h^2), \quad (2.2.2)$$

where $\delta_x^2 u(x, t) = u(x - h, t) - 2u(x, t) + u(x + h, t)$.

Equation (2.2.2) is then applied to all interior mesh points to produce a system of ordinary differential equations

$$\frac{dU}{dt} = AU(t), \quad (2.2.3)$$

where $U = (U_1, U_2, \dots, U_N)^T$ is an N -vector of the (unknown) approximation to u and A is the tridiagonal matrix of order N given by (1.2.7) of which the eigenvalues are given by (1.2.7). Solving (2.2.3) with the initial vector $U(0) = g$ gives

$$U(t) = \exp(\tau A) U(0) \quad (2.2.4)$$

which satisfies a recurrence relation of the form

$$U(t + \tau) = \exp(\tau A) U(t), \quad t = 0, \tau, 2\tau, \dots \quad (2.2.5)$$

If the diagonal (1,1) Padé approximation

$$\left(I - \frac{\tau}{2}A\right)^{-1} \left(I + \frac{\tau}{2}A\right) = \exp(\tau A)$$

is used to approximate $\exp(\tau A)$, we get the Crank-Nicolson method

$$U(t + \tau) = \left(I - \frac{\tau}{2}A\right)^{-1} \left(I + \frac{\tau}{2}A\right) U(t) \quad (2.2.6)$$

or

$$\left(I - \frac{\tau}{2}A\right) U(t + \tau) = \left(I + \frac{\tau}{2}A\right) U(t).$$

in which the error in time is $O(\tau^3)$. Since the eigenvalues $\lambda_i, i = 1, 2, \dots, N$ of the matrix A are all different, the N eigenvectors w_i of A are linearly independent and can be used as

a basis for all N dimensional spaces of the vector g of the initial values. In other words, g can be expressed as

$$g = \sum_{i=1}^N \alpha_i w_i$$

where the $\alpha_i (i = 1, 2, \dots, N)$ are constants. Writing (2.2.6) over a second time step

$$\begin{aligned} U(t + 2\tau) &= (I - \frac{\tau}{2}A)^{-1}(I + \frac{\tau}{2}A)U(t + \tau) \\ &= \left[(I - \frac{\tau}{2}A)^{-1}(I + \frac{\tau}{2}A) \right]^2 U(t). \end{aligned}$$

In general after n -time steps (2.2.6) becomes

$$U(t + n\tau) = \left[(I - \frac{\tau}{2}A)^{-1}(I + \frac{\tau}{2}A) \right]^n U(t).$$

Putting $t = 0$ we get

$$\begin{aligned} U(n\tau) &= \left[(I - \frac{\tau}{2}A)^{-1}(I + \frac{\tau}{2}A) \right]^n U(0) \\ &= \left[(I - \frac{\tau}{2}A)^{-1}(I + \frac{\tau}{2}A) \right]^n \sum_{i=1}^N \alpha_i w_i \end{aligned}$$

Since $Aw_i = \lambda_i w_i$, $f(A) w_i = f(\lambda_i) w_i$; it follows that

$$U(n\tau) = \left[\frac{(I + \frac{\tau}{2}\lambda_i)}{(I - \frac{\tau}{2}\lambda_i)} \right]^n \sum_{i=1}^N \alpha_i w_i$$

Then the analytical solution of (2.2.6) may be written as

$$U(n\tau) = \sum_{i=1}^N \alpha_i \left[\frac{(1 + (\tau/2)\lambda_i)}{(1 - (\tau/2)\lambda_i)} \right]^n w_i, \quad (2.2.7)$$

Equation (2.2.7) shows explicitly that there is no stability restriction on τ for the Crank-Nicolson scheme since the growth factor associated with the eigenvectors w_i is

$$R_{1,1}(\tau\lambda) = (1 - \frac{\tau}{2}\lambda_i)^{-1}(1 + \frac{\tau}{2}\lambda_i).$$

This function is less than 1 in modulus for all τ , since all $\lambda_i < 0$. However, the amplification symbol of the method is

$$R_{1,1}(-z) = \frac{(1 + \frac{1}{2}z)}{(1 - \frac{1}{2}z)},$$

with $|R_{1,1}(-z)| < 1$ for all $z > 0$ and

$$\lim_{z \rightarrow \infty} R_{1,1}(-z) = -1$$

showing that the method is A_0 -stable. This will be the case if h is small for those λ_i corresponding to $i = N, N-1, \dots$ (since, for example, $\sin^2[N\pi/(2(N+1))] \simeq 1$ and $\lambda_N \simeq \frac{-4}{h^2}$).

Accordingly, one may expect to see components $\alpha_N w_N, \alpha_{N-1} w_{N-1}, \dots$ of the initial condition preserved at subsequent solution steps, but with alternating sign. For problems with smooth initial data such an effect may not be noticed. But, for initial conditions with discontinuities such discontinuities are preserved and distorted rather than damped out, large values of $\alpha_N, \alpha_{N-1}, \dots$ characterize discontinuous problems.

Because of this oscillatory behaviour for sufficiently large τ , it is commonly suggested that the time step τ should be restricted so as to prevent oscillatory solutions. This would imply preventing the numerator of the growth factor in (2.2.7) becoming negative which imposes the condition

$$\tau \leq \frac{-2}{\lambda_i}, \quad i = 1, 2, \dots, N,$$

which in turn imposes the condition (approximately)

$$\tau \leq \frac{h^2}{2} \tag{2.2.8}$$

(see [22]). However, Lawson and Morris [9] show that this restriction is not strictly necessary, oscillatory components are permissible provided they decay to zero faster than the decay of the primary components. Thus, to ensure that the higher frequency component, is oscillatory or not, and damped to zero faster than the lowest frequency component, the inequality

$$\left| \frac{1 + (\tau/2)\lambda_N}{1 - (\tau/2)\lambda_N} \right| < \left| \frac{1 + (\tau/2)\lambda_1}{1 - (\tau/2)\lambda_1} \right|$$

is required. That is,

$$\frac{-1 - (\tau/2)\lambda_1}{1 - (\tau/2)\lambda_1} < \frac{1 + (\tau/2)\lambda_N}{1 - (\tau/2)\lambda_N} < \frac{1 + (\tau/2)\lambda_1}{1 - (\tau/2)\lambda_1} \quad (2.2.9)$$

The right-hand side inequality in (2.2.9) is trivially satisfied, whereas the left-hand side inequality gives

$$\tau^2 \lambda_1 \lambda_N < 4$$

For large values of N ,

$$\begin{aligned} \lambda_1 &\simeq \frac{-4}{h^2} \sin^2 \frac{\pi}{2(N+1)} = \frac{-4}{h^2} \frac{\pi^2}{4(N+1)^2} \\ &= \frac{-\pi^2}{h^2(N+1)^2} = \frac{-\pi^2}{X^2} \end{aligned}$$

(since, for any small angle θ , $\sin \theta \cong \theta$), and

$$\lambda_N \cong \frac{-4}{h^2}.$$

Thus,

$$\begin{aligned} \tau^2 \cdot \frac{-\pi^2}{X^2} \cdot \frac{-4}{h^2} &< 4 \\ \tau^2 &< \frac{X^2 h^2}{\pi^2} \\ \tau &< \frac{Xh}{\pi} \quad (\text{approximately}). \end{aligned} \quad (2.2.10)$$

So that, to produce solutions which damp high frequency components at faster rate than the low frequency components we have a linear relation between τ and h (not second order as is commonly used) This restriction is less severe than that implied by (2.2.8) but still restricts τ to being small. However, as it will be discussed in the next section, higher order diagonal Padé approximations also suffer a time step restriction similar to that in (2.2.10).

2.3 Factorized Padé Approximations

High order diagonal Padé methods were implemented in an efficient manner for serial machines by Reusch *et al.* [14] In this case, since $Q_m(z) = P_m(-z)$, these have the form

$$R_{m,m}(z) = \frac{P_m(z)}{P_m(-z)} = \prod_{k=1}^m \frac{1 - z/c_k}{1 + z/c_k^*}, \quad (2.3.1)$$

where c_k are the roots of $P_m(z)$ and c_k^* is the complex conjugate of c_k . Numerical values of these roots are listed in [14] for $m = 1(1)11$. For the resulting algorithm, each substep of a given time step has the form of the Crank-Nicolson expression (2.2.6), that is,

$$(I + \frac{\tau}{c_k^*} A)y^k = (I - \frac{\tau}{c_k} A)y^{k-1}, \quad (2.3.2)$$

where $1 \leq k \leq m$, $y^0 = y(t)$, and $y(t + \tau) = y^m$. This approach incurs the extra cost of complex arithmetic but preserves A_0 -stability without forming matrix products, thereby ensuring that the condition number of each of the substeps remains tolerable. However, to ensure proper damping, a linear relation exists between τ and h similar to that in (2.1.10). Specifically, approximating the matrix exponential function (2.1.5) with the (m,m) Padé approximation requires

$$|R_{m,m}(\tau\lambda_N)| < |R_{m,m}(\tau\lambda_1)|, \quad (2.3.3)$$

as with the (1,1) Padé approximation.

Theorem 2.1 Khaliq, *et al.*[7] *For diagonal Padé approximants to e^z , and $w < v < 0$,*

$$|R_{m,m}(w)| < |R_{m,m}(v)|, \quad \text{if } wv < \min_{1 \leq k \leq m} |c_k|^2.$$

Proof. Using the factored form of $R_{m,m}(z)$ given in (2.3.1), we need to show that

$$\prod_{k=1}^m \left| \frac{1 - w/c_k}{1 + w/c_k^*} \right| < \prod_{k=1}^m \left| \frac{1 - v/c_k}{1 + v/c_k^*} \right|. \quad (2.3.4)$$

Inequality (2.3.4) is true provided

$$\left| \frac{1 - w/c_k}{1 + w/c_k^*} \right| < \left| \frac{1 - v/c_k}{1 + v/c_k^*} \right|, \quad k = 1(1)m,$$

or

$$\frac{(1 - w/c_k)(1 - w/c_k^*)}{(1 + w/c_k^*)(1 + w/c_k)} < \frac{(1 - v/c_k)(1 - v/c_k^*)}{(1 + v/c_k^*)(1 + v/c_k)}, \quad k = 1(1)m. \quad (2.3.5)$$

Denoting $c_k = \alpha_k + i\beta_k$, (2.3.5) becomes

$$\frac{-4\alpha_k(w - v)(vw - \alpha_k^2 - \beta_k^2)}{(v^2 + 2\alpha_k v + \alpha_k^2 + \beta_k^2)(w^2 + 2\alpha_k w + \alpha_k^2 + \beta_k^2)} > 0. \quad (2.3.6)$$

Noting that $\alpha_k < 0$ (2.3.6) will be true provided

$$vw < \alpha_k^2 + \beta_k^2 = |c_k|^2, \quad k = 1(1)m, \quad (2.3.7)$$

and the result follows.

Since

$$m + 1 \leq \min_{1 \leq k \leq m} |c_k|^2 \quad (2.3.8)$$

(see [15]), we get the following corollary :

Corollary 2.1 *For a diagonal Padé approximants to e^z , and $w < v < 0$,*

$$|R_{m,m}(w)| < |R_{m,m}(v)|, \quad \text{if } wv < (m + 1)^2.$$

with the use of the central discretization in space, the extension of time step restriction (2.2.10) corresponding to higher order diagonal Padé approximations now follows from Theorem 1 by setting $v = \tau\lambda_1$, and $w = \tau\lambda_N$, where λ_1 and λ_N are given in (1.2.7). Now, (2.3.3) is true if

$$\begin{aligned} wv &< \min_{1 \leq k \leq m} |c_k|^2 \\ \tau^2 \lambda_1 \lambda_N &\leq \min_{1 \leq k \leq m} |c_k|^2 \\ \tau^2 \frac{-\pi^2}{X^2} \frac{-4}{h^2} &\leq \min_{1 \leq k \leq m} |c_k|^2 \\ \frac{\tau^2}{h} &\leq \min_{1 \leq k \leq m} |c_k|^2 \frac{X^2}{4\pi^2} \\ \frac{\tau}{h} &\leq \min_{1 \leq k \leq m} |c_k| \frac{X}{2\pi} \quad (\text{approximately}). \end{aligned} \quad (2.3.9)$$

Analog of (2.3.9) corresponding to other spatial discretizations of this problem depend on the eigenvalues of the associated matrix.

Setting $m = 1$, the time step restriction given in inequality (2.2.10) results from (2.3.9). For $m = 2$ and $m = 3$, we get the following linear relations

$$\begin{aligned} \frac{\tau}{h} &< \sqrt{12} \frac{X}{2\pi} \\ &\approx 1.73 \frac{X}{\pi} \end{aligned} \quad (2.3.10)$$

and

$$\begin{aligned} \frac{\tau}{h} &< \frac{16 + c^{1/3} - (\sqrt{5} - 1)c^{2/3}}{4} \frac{X}{2\pi}, \quad c = 4\sqrt{5} + 4 \\ &\approx 2.32 \frac{X}{\pi} \end{aligned} \quad (2.3.11)$$

2.4 Partial Fraction Decomposition of Padé Approximations

Letting

$$R_{m,k}(z) = \frac{P_k(z)}{Q_m(z)} \quad (2.4.1)$$

denote the (m, k) Padé approximation to e^z . If $R_{m,k}(z)$ possesses q_1 real poles and $2q_2$ complex poles, where $q_1 + 2q_2 = m$, then, as noted by Zakian [25], $R_{m,k}(z)$ can be expanded in partial fraction form resulting in approximating the matrix exponential function in (2.2.5) by

$$e^{\tau A} \approx \sum_{i=1}^{q_1} w_i (\tau A - c_i I)^{-1} + 2 \sum_{i=q_1+1}^{q_1+q_2} \text{Re} \{w_i (\tau A - c_i I)^{-1}\}. \quad (2.4.2)$$

where the w_i ($i = 1, 2, \dots, q_1 + q_2$) are weights. The above expansion (2.4.2) is particularly useful in parallel computing environments because it can be used to solve a corresponding linear algebraic systems on processors operating concurrently. Using this expansion, parallel algorithms may be implemented on machines with $q_1 + q_2$ processors (only). For the diagonal (m, m) Padé approximants, the form of the partial fraction expansion to the matrix exponential

as suggested by Gallopoulos and Saad [2], becomes

$$e^\tau A \approx (-1)^m I + \sum_{i=1}^{q_1} w_i (\tau A - c_i I)^{-1} + 2 \sum_{i=q_1+1}^{q_1+q_2} \text{Re} \{w_i (\tau A - c_i I)^{-1}\}, \quad (2.4.3)$$

with the resulting methods being A_0 -stable and suffering the time step restriction .

Since the poles of the subdiagonal Padé approximations

$$R_{m,k}(z) = P_k(z)/Q_m(z)$$

are distinct, $R_{m,k}(z), m > k$, can be expanded in terms of partial fractions to give

$$R_{m,k}(z) = \frac{P_k(z)}{\prod_{i=1}^m (z - c_i)} = \sum_{i=1}^m \frac{w_i}{z - c_i}, \quad (2.4.4)$$

where

$$w_i = \frac{P_k(c_i)}{Q'_m(c_i)}, \quad (2.4.5)$$

Noting that, if c_s is a complex pole of $R_{m,k}(z)$ then so is c_s^* (its complex conjugate), and the expansion coefficients associated with this conjugate pair must also be conjugate. The corresponding contribution to (2.4.4) can be written

$$\frac{w_s}{z - c_s} + \frac{w_s^*}{z - c_s^*} = 2 \text{Re} \left(\frac{w_s}{z - c_s} \right). \quad (2.4.6)$$

and consequently, if $R_{m,k}$ possesses q_1 real poles and $2q_2$ complex poles, where $q_1 + 2q_2 = m$, then (2.4.4) has the form

$$R_{m,k}(z) = \sum_{i=1}^{q_1} \frac{w_i}{z - c_i} + 2 \sum_{i=q_1+1}^{q_1+q_2} \text{Re} \left(\frac{w_i}{z - c_i} \right). \quad (2.4.7)$$

Using this expanded form, the approximation to the matrix exponential function occurring in (2.2.5) is given in (2.4.2), and the resulting algorithm is given in the next section.

2.5 The Implementation

Algorithm developed in section 2.4 may be implemented on parallel computers in the following manner .

Algorithm 1

- 1) For $i = 1, 2, \dots, q_1 + q_2$, solve $(\tau A - c_i I)v_i = y(t)$ in parallel.
- 2) Compute

$$y(t + \tau) = \sum_{i=1}^{q_1} w_i v_i + 2 \sum_{i=q_1+1}^{q_1+q_2} Re(w_i v_i).$$

A similar algorithm arises for diagonal Padé approximations resulting from the approximation to the matrix exponential given in (2.4.3). When these algorithms resulting from the use of (m,k) Padé approximations $m \geq k$ are applied to (2.2.1), the principal part of the local truncation error (PLTE) is given by

$$PLTE : -\frac{1}{12}h^2 \frac{\partial^4 u}{\partial x^4} + C_{q+1}\tau^q \frac{\partial^{q+1} u}{\partial t^{q+1}}, \quad q = m + k. \quad (2.5.1)$$

where the error constants C_{q+1} together with the poles c_s are given in Table 1 for several values of $q \leq 4$. Note that all of the corresponding methods require the solution of at most two complex linear systems of equations. Implementing these in parallel on machine with at least two processors, as described by Algorithm 1, enables methods arising from L_0 – stable subdiagonal Padé approximations to be competitive with those arising from diagonal Padé approximations even on problems possessing smooth initial data.

error at the mesh point $(mh, n\tau)$ given by (2.5.1) with $C_{q+1} = \frac{1}{6}$. In (2.5.1) the component $-\frac{1}{12}h^2 \frac{\partial^4 u}{\partial x^4}$ is due to the use of (1.2.4) in (2.2.1); this term will appear in the local truncation error of every finite difference scheme arising from the use of Padé approximants to the matrix exponential in the recurrence relation (2.2.5). Applying (2.5.3) to the general mesh point $(mh, n\tau)$ at time $t = n\tau$ leads to a linear system of the form

$$G U(t + \tau) = B^n \quad (2.5.4)$$

where G is a quin-diagonal matrix of order N and is given by

$$G = \begin{bmatrix} g_4 & g_2 & g_3 & & & & 0 \\ g_2 & g_1 & g_2 & g_3 & & & \\ g_3 & g_2 & g_1 & g_2 & g_3 & & \\ & g_3 & g_2 & g_1 & g_2 & g_3 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & g_3 & g_2 & g_1 & g_2 \\ 0 & & & & g_3 & g_2 & g_4 \end{bmatrix} \quad (2.5.5)$$

with

$$\begin{aligned} g_1 &= 3p^2 + 2p + 1 \\ g_2 &= -(2p^2 + p) \\ g_3 &= \frac{p^2}{2} \\ g_4 &= \frac{5p^2}{2} + 2p + 1 \end{aligned} \quad (2.5.6)$$

and $p = \frac{\tau}{h^2}$. The vector B^n is given by $B^n = [B_1^n, B_2^n, \dots, B_N^n]^T$, where T denotes transpose. It is easily seen from (2.5.3) that the vector B^n is $U(t_n)$. The solution vector $U(t + \tau)$ of this linear system is found by the application of a quin-diagonal solver with real arithmetic at every time step.

Parallel implemmentation

To achieve the parallel implementation, (2.5.2) is expressed in its partial fraction form to give

$$U(t + \tau) = [k_1(I - r_1\tau A)^{-1} + k_2(I - r_2\tau A)^{-1}]U(t), \quad (2.5.7)$$

where r_1 and r_2 are the poles of the expression $(I - \tau A + \frac{1}{2}\tau^2 A^2)$ and are given by

$$\begin{aligned} r_1 &= \frac{1}{2}(1 + i) \\ r_2 &= \overline{r_1} \end{aligned} \quad (2.5.8)$$

and k_1 and k_2 are the constants resulting from the partial fraction splitting given by

$$\begin{aligned} k_1 &= \frac{1}{2}(1 - i) \\ k_2 &= \overline{k_1} \end{aligned} \quad (2.5.9)$$

The tridiagonal matrix $(I - r_1 \tau A)$ has the form

$$(I - r_1 \tau A) = \begin{bmatrix} a & d & & & & 0 \\ c & a & d & & & \\ & c & a & d & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & c & a & d \\ & & & & c & a \end{bmatrix} \quad (2.5.10)$$

where

$$\begin{aligned} a &= 1 + 2r_1 p \\ c &= -r_1 p \\ d &= -r_1 p \end{aligned} \quad (2.5.11)$$

The solution vector $U(t + \tau)$ may be obtained *via* the algorithm

$$\begin{aligned} \text{Processor } 1 : (I - r_1 \tau A)z &= k_1 U(t) \\ \text{Processor } 2 : (I - r_2 \tau A)w &= k_2 U(t) \\ U(t + \tau) &= z + w \end{aligned} \quad (2.5.12)$$

Here z and w are intermediate vectors to be determined and need not be stored once the solution, obtained using two tridiagonal solvers, is found.

The algorithm above involves the use of two processors to solve the two linear systems using complex arithmetic. This approach has the advantage of avoiding the difficulties arising from squaring the matrix A .

A Sequential algorithm

Alternatively, (2.5.2) may be written as

$$(I - \tau A + \frac{1}{2}\tau^2 A^2)U(t + \tau) = U(t), \quad (2.5.13)$$

which factorized as

$$(I - r_1 \tau A)(I - r_2 \tau A)U(t + \tau) = U(t), \quad (2.5.14)$$

and the solution vector $U(t + \tau)$ obtained *via* the two stage process

$$\begin{aligned} (I - r_1 \tau A)v &= U(t) \\ (I - r_2 \tau A)U(t + \tau) &= v, \end{aligned} \quad (2.5.15)$$

where the intermediate vector v need not be stored once the solution is determined. Here, also, two linear systems are solved using two tridiagonal solvers at every time step involving complex arithmetic.

An Alternative sequential algorithm

For the (2,0) Padé approximant, $q_1 = 0$ and $q_2 = 1$, and using (2.4.7) the solution vector $U(t + \tau)$ may be obtained as follows

$$U(t + \tau) = 2 \operatorname{Re}(v) \quad (2.5.16)$$

where v is determined by solving the tridiagonal system

$$(I - r_1 \tau A)v = k_1 U(t) \quad (2.5.17)$$

2.5.2 Fourth-order methods

Fourth-order accuracy in time may be achieved by approximating the matrix exponential in (2.2.5) by the (3,1) and (4,0) Padé approximants. These method are easily shown to be L_0 -stable.

Methods based on the (3,1) Padé approximant

Replacing now the exponential term in the recurrence relation (2.2.5) by its (3,1) Padé approximant gives

$$U(t + \tau) = (I - \frac{3}{4}\tau A + \frac{1}{4}\tau^2 A^2 - \frac{1}{24}\tau^3 A^3)^{-1}(I + \frac{1}{4}\tau A)U(t) \quad (2.5.18)$$

The resulting implicit, L_0 – stable scheme is given by

$$(I - \frac{3}{4}\tau A + \frac{1}{4}\tau^2 A^2 - \frac{1}{24}\tau^3 A^3)U(t + \tau) = (I + \frac{1}{4}\tau A)U(t), \quad (2.5.19)$$

which, on application to the general mesh point $(mh, n\tau)$ at time $t = n\tau$, leads to a linear system of the form

$$EU(t + \tau) = B^n. \quad (2.5.20)$$

where the coefficient matrix E is seven diagonal and is given by

$$\begin{bmatrix} e_5 & e_6 & e_3 & e_4 & & & & 0 \\ e_6 & e_1 & e_2 & e_3 & e_4 & & & \\ e_3 & e_2 & e_1 & e_2 & e_3 & e_4 & & \\ e_4 & e_3 & e_2 & e_1 & e_2 & e_3 & e_4 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & e_4 & e_3 & e_2 & e_1 & e_2 & e_3 & e_4 \\ & & & e_4 & e_3 & e_2 & e_1 & e_2 & e_3 \\ & & & & e_4 & e_3 & e_2 & e_1 & e_6 \\ 0 & & & & & e_4 & e_3 & e_6 & e_5 \end{bmatrix} \quad (2.5.21)$$

where

$$\begin{aligned} e_1 &= \frac{20p^3}{24} + \frac{3p^2}{2} + \frac{3p}{2} + 1, \\ e_2 &= -(\frac{15p^3}{24} + p^2 + \frac{3p}{4}), \\ e_3 &= \frac{6p^3}{24} + \frac{p^2}{4}, \\ e_4 &= -\frac{p^3}{24}, \\ e_5 &= \frac{14p^3}{24} + \frac{5p^2}{4} + \frac{3p}{2} + 1, \\ e_6 &= -(\frac{14p^3}{24} + p^2 + \frac{3p}{4}), \end{aligned} \quad (2.5.22)$$

and the vectors B^n are given by

$$\begin{aligned} B_1^n &= (1 - \frac{p}{2})U_1^n + \frac{p}{4}U_2^n \\ B_m^n &= \frac{p}{4}U_{m-1}^n + (1 - \frac{p}{2})U_m^n + \frac{p}{4}U_{m+1}^n \quad ; m = 2, \dots, N-1 \\ B_N^n &= \frac{p}{4}U_{N-1}^n + (1 - \frac{p}{2})U_N^n. \end{aligned} \quad (2.5.23)$$

The solution is determined by the application of seven-diagonal solvers.

A Parallel implementation

The partial fraction splitting of (2.5.18) takes the form

$$U(t + \tau) = [k_8(I - r_8\tau A)^{-1} + k_9(I - r_9\tau A)^{-1} + k_{10}(I - r_{10}\tau A)^{-1}]U(t). \quad (2.5.24)$$

Closed form expressions for r_m and k_m ($m = 8, 9, 10$) are not available and they have the approximate values

$$\begin{aligned} r_8 &= 0.38083387729959 \\ r_9 &= 0.18458306140401 - 0.27447791814622 i \\ r_{10} &= \overline{r_9}, \end{aligned} \quad (2.5.25)$$

and

$$\begin{aligned} k_8 &= 2.11011239736648 \\ k_9 &= -0.55506198683241 + 0.73103036838333 i \\ k_{10} &= \overline{k_9}. \end{aligned} \quad (2.5.26)$$

The solution vector may be determined as follows :

$$\begin{aligned} \text{Processor 1 : } (I - r_8\tau A)y &= k_8U(t) \\ \text{Processor 2 : } (I - r_9\tau A)q &= k_9U(t) \\ \text{Processor 3 : } (I - r_{10}\tau A)b &= k_{10}U(t) \\ U(t + \tau) &= y + q + b. \end{aligned} \quad (2.5.27)$$

Here again the three processors determine the intermediate vectors y, q and b concurrently with processor 1 using real arithmetic while processors 2 and 3 use complex arithmetic.

An Alternative parallel implementation

For the (3,1) Padé approximant, $q_1 = q_2 = 1$, therefore the algorithm may be implemented on a machine with two processors as follows :

$$\begin{aligned} \text{Processor 1 : } (I - r_8\tau A)y &= k_8U(t) \\ \text{Processor 2 : } G^{**} &= 2 \operatorname{Re}(G_2) \\ U(t + \tau) &= y + G^{**} \end{aligned} \quad (2.5.28)$$

where G_2 is obtained by solving the system $(I - r_9\tau A)G_2 = k_9U(t)$.

A sequential algorithm based on the (3,1)Padé approximant

Equation (2.5.18) may be factorized to given

$$(I - r_8\tau A)(I - r_9\tau A)(I - r_{10}\tau A)U(t + \tau) = (I + \frac{1}{4}\tau A)U(t) \quad (2.5.29)$$

and the solution determined *via* the three stage process

$$\begin{aligned} (I - r_8\tau A)y_1 &= (I + \frac{1}{4}\tau A)U(t), \\ (I - r_9\tau A)y_2 &= y_1, \end{aligned} \quad (2.5.30)$$

$$(I - r_{10}\tau A)U(t + \tau) = y_2,$$

where the intermediate vectors y_1 and y_2 need not be stored once the solution is found.

Methods based on the (4,0) Padé approximant

Approximating now the matrix exponential term in the recurrence relation (2.2.5) by the (4,0) Padé approximant gives

$$U(t + \tau) = (I - \tau A + \frac{1}{2}\tau^2 A^2 - \frac{1}{6}\tau^3 A^3 + \frac{1}{24}\tau^4 A^4)^{-1}U(t). \quad (2.5.31)$$

This suggest the fully implicit, L_0 - stable scheme

$$(I - \tau A + \frac{1}{2}\tau^2 A^2 - \frac{1}{6}\tau^3 A^3 + \frac{1}{24}\tau^4 A^4)U(t + \tau) = U(t). \quad (2.5.32)$$

applying (2.5.32) to the general mesh point $(mh, n\tau)$ at time $t = n\tau$ leads to a linear system of the form

$$F U(t + \tau) = B^n, \quad (2.5.33)$$

where the matrix F is nine-diagonal and is given by

$$F = \begin{bmatrix} f_6 & f_7 & f_8 & f_4 & f_5 & & & & 0 \\ f_7 & f_9 & f_2 & f_3 & f_4 & f_5 & & & \\ f_8 & f_2 & f_1 & f_2 & f_3 & f_4 & f_5 & & \\ f_4 & f_3 & f_2 & f_1 & f_2 & f_3 & f_4 & f_5 & \\ f_5 & f_4 & f_3 & f_2 & f_1 & f_2 & f_3 & f_4 & f_5 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & f_5 & f_4 & f_3 & f_2 & f_1 & f_2 & f_3 & f_4 \\ & & f_5 & f_4 & f_3 & f_2 & f_1 & f_2 & f_8 \\ & & & f_5 & f_4 & f_3 & f_2 & f_9 & f_7 \\ 0 & & & & f_5 & f_4 & f_8 & f_7 & f_6 \end{bmatrix} \quad (2.5.34)$$

with

$$\begin{aligned} f_1 &= \frac{70p^4}{24} + \frac{20p^3}{6} + 3p^2 + 2p + 1 \\ f_2 &= -\left(\frac{7p^4}{3} + \frac{5p^3}{2} + 2p^2 + p\right) \\ f_3 &= \frac{7p^4}{6} + p^3 + \frac{p^2}{2} \\ f_4 &= -\left(\frac{p^4}{3} + \frac{p^3}{6}\right) \\ f_5 &= \frac{p^4}{24} \\ f_6 &= \frac{42p^4}{24} + \frac{14p^3}{6} + \frac{5p^2}{2} + 2p + 1 \\ f_7 &= -\left(\frac{48p^4}{24} + \frac{14p^3}{6} + 2p^2 + p\right) \\ f_8 &= \frac{27p^4}{24} + p^3 + \frac{p^2}{2} \\ f_9 &= \frac{69p^4}{24} + \frac{20p^3}{6} + 3p^2 + 2p + 1. \end{aligned} \quad (2.5.35)$$

The vectors B^n are seen from (2.5.32) to be $U(t_n)$, and the solution will be determined *via* the application of nine-diagonal solver at each time step. The principal part of the local truncation error is given by (2.5.1) with $q = 4$ and $C_5 = \frac{1}{120}$.

Parallel implementation

Equation (2.5.31) may be expressed in its partial fraction form to give

$$U(t + \tau) = [g_1(I - c_1\tau A)^{-1} + g_2(I - c_2\tau A)^{-1} + g_3(I - c_3\tau A)^{-1} + g_4(I - c_4\tau A)^{-1}]U(t), \quad (2.5.36)$$

where

$$\begin{aligned} c_1 &= 4.2626656502702E - 2 + 0.39463295317211 \, i \\ c_2 &= \overline{c_1} \\ c_3 &= 0.45737334349730 + 0.23510048999854 \, i \\ c_4 &= \overline{c_3}, \end{aligned} \quad (2.5.37)$$

and

$$\begin{aligned} g_1 &= -0.2116960248671 - 0.20306415938099 \, i \\ g_2 &= \overline{g_1} \\ g_3 &= 0.62116960248677 - 0.59941529409522 \, i \\ g_4 &= \overline{g_3}. \end{aligned} \quad (2.5.38)$$

Closed form expressions for c_m and g_m ($m = 1, 2, 3, 4$) are not available. Four processors will be used to compute the solution concurrently as follows :

$$\begin{aligned} \text{Processor } 1 : (I - c_1\tau A)s &= g_1U(t), \\ \text{Processor } 2 : (I - c_2\tau A)f &= g_2U(t), \\ \text{Processor } 3 : (I - c_3\tau A)g &= g_3U(t), \\ \text{Processor } 4 : (I - c_4\tau A)m &= g_4U(t), \\ U(t + \tau) &= s + f + g + m. \end{aligned} \quad (2.5.39)$$

Here all the processors carry out complex arithmetic to solve the four linear systems each using a tridiagonal solver.

An alternative parallel implementation

Here $q_2 = 2$ and using (2.5.1), two processors may be used to implement the parallel algorithm based on the (4,0) Padé approximant as follows :

$$\begin{aligned} \text{Processor } 1 : d_1 &= 2\text{Re}(s) \\ \text{Processor } 2 : d_2 &= 2\text{Re}(g) \\ U(t + \tau) &= d_1 + d_2 \end{aligned} \quad (2.3.40)$$

where s and g are determined using processors 1 and 2 in (2.5.39) respectively.

A sequential algorithm based on the (4,0) Padé approximants

The sequential algorithm based on the (4,0) Padé approximant is obtained by expressing (2.5.32) in its complex factors to give

$$(I - c_1\tau A)(I - c_2\tau A)(I - c_3\tau A)(I - c_4\tau A)U(t + \tau) = U(t), \quad (2.3.41)$$

The solution vector $U(t + \tau)$ may be obtained *via* the algorithm

$$\begin{aligned} (I - c_1\tau A)s_1 &= U(t), \\ (I - c_2\tau A)s_2 &= s_1, \\ (I - c_3\tau A)s_3 &= s_2, \\ (I - c_4\tau A)U(t + \tau) &= s_3. \end{aligned} \quad (2.5.42)$$

This algorithm requires the application of four tridiagonal solvers with complex arithmetic to compute the solution at every time step.

2.6 Numerical Results

We consider a test problem from the literature which has high-frequency components in the solution.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 2, \quad t > 0, \quad (2.6.1a)$$

with boundary conditions

$$u(0, t) = u(2, t) = 0, \quad t > 0, \quad (2.6.1b)$$

and initial condition

$$u(x, 0) = 1, 0 \leq x \leq 2. \quad (2.6.1c)$$

The problem has a discontinuity between initial and boundary conditions and the exact solution is given by

$$u(x, t) = \sum_{n=1}^{\infty} [1 - (-1)^n] \frac{2}{n\pi} \sin\left(\frac{n\pi x}{2}\right) \exp\left(\frac{-n^2\pi^2 t}{4}\right). \quad (2.6.2)$$

Methods corresponding to diagonal Padé approximations through fourth-order were implemented on this example problem together with those corresponding to the same order subdiagonal Padé approximations referenced in Table 1. Letting $r = \frac{\tau}{h}$, the approximate time step restriction for the methods corresponding to (1,1) and (2,2) Padé approximations become

$$r < 0.637, \quad r < 1.103, \quad (2.6.3)$$

respectively. Table 2 contains the maximum errors between the exact solution and the approximate solution on various spatial meshes at time $t = 1.2$. Both τ and h are refined keeping their ratios constant on one side ($r = 1/2$) and then on the other side ($r = 2$) of the bound in (2.6.1).

TABLE 2A.
Second-order methods.

r	h	τ	(1,1)	(2,0)
1/2	0.1	0.05	3.48E-04	7.18E-04
	0.05	0.025	3.61E-04	1.84E-04
	0.025	0.0125	3.61E-04	4.71E-05
2	0.05	0.1	3.44E-01	1.74E-03
	0.025	0.05	2.44E-01	4.69E-04
	0.0125	0.025	2.44E-01	1.16E-04

TABLE 2B.
Fourth-order methods.

r	h	τ	(3,1)	(4,0)
1/2	0.1	0.05	2.66E-04	2.67E-04
	0.05	0.025	6.50E-05	6.72E-05
	0.025	0.0125	1.39E-05	1.51E-05
2	0.05	0.1	6.44E-05	7.13E-05
	0.025	0.05	1.10E-05	1.65E-05
	0.0125	0.025	2.49E-05	1.32E-05

A visual analysis of Table 2 demonstrates that for problems with incompatibility between initial and boundary data, the unrestricted subdiagonal Padé schemes offer better error behavior than the diagonal schemes when they fail to meet the condition on $\frac{\tau}{h}$ deduced from Theorem 1. This is contrary to what one might expect based solely on the comparison of the error constants (see Table 1) of diagonal and subdiagonal methods of the same order.

CHAPTER III

THE EXTRAPOLATION OF FIRST ORDER METHODS FOR PARABOLIC PDEs BASED ON (1,0) PADÉ APPROXIMATIONS

The Component of the local truncation error due to the approximation of the derivative by its central difference replacement will not be changed, however, the component due to the chosen Padé approximant can be improved by at least one power of τ by the process of *extrapolation* (see,[9],[20]).

This process is illustrated in this chapter for the finite difference schemes based on (1,0) Padé approximations in one space variable and in two space variables.

3.1 Extrapolation of the Fully Implicit Scheme

In this section a novel algorithm for a constant coefficient equation in one space variable, is analyzed.

Consider equation (1.3.5) written over a time interval 2τ so that

$$v^{(1)}(t + 2\tau) = (I - 2\tau A)^{-1}v(t). \quad (3.1.1)$$

and if (1.3.5) is applied twice, each over a time-interval τ , we have

$$\begin{aligned} v^{(2)}(t + 2\tau) &= (I - \tau A)^{-1}v(t + \tau) \\ &= (I - \tau A)^{-1}(I - \tau A)^{-1}v(t), \\ &= (I - \tau A)^{-2}v(t). \end{aligned} \quad (3.1.2)$$

Equations (3.1.1) and (3.1.2) are two different backward difference schemes for computing the solution at time $t + 2\tau$.

The binomial expansion of the matrix inverse in (3.1.1) and (3.1.2) produces

$$v^{(1)}(t + 2\tau) = (I + 2\tau A + 4\tau^2 A^2)v(t) + O(\tau^3), \quad (3.1.3)$$

$$v^{(2)}(t + 2\tau) = (I + 2\tau A + 3\tau^2 A^2)v(t) + O(\tau^3). \quad (3.1.4)$$

But the Maclaurin expansion of $\exp(2\tau A)$ in

$$v(t + 2\tau) = \exp(2\tau A)v(t),$$

result in

$$v(t + 2\tau) = (I + 2\tau A + 2\tau^2 A^2)v(t) + O(\tau^3). \quad (3.1.5)$$

Comparison of equations (3.1.3), (3.1.4), and (3.1.5) shows that neither (3.1.3) nor (3.1.4) is accurate to terms of order τ^2 . However, if we combine the expansions in (3.1.3) and (3.1.4) then by extrapolation, a vector v which is second order accurate in τ is produced. The algorithm for the extrapolation is therefore

$$\begin{aligned} v^{(1)}(t + 2\tau) &= (I - 2\tau A)^{-1}v(t), \\ v^{(2)}(t + 2\tau) &= (I - \tau A)^{-1}(I - \tau A)^{-1}v(t), \\ v(t + 2\tau) &= 2v^{(2)} - v^{(1)}. \end{aligned} \quad (3.1.6)$$

$v(t + 2\tau)$ is now a second order approximation to the solution $u(t + 2\tau)$.

The analysis of stability of the algorithm defined by (3.1.6) indicates that the method is stable for all positive τ , and the amplification symbol for the method is

$$\begin{aligned} R_{1,0}(-z) &= 2\left(\frac{1}{1+z}\right)^2 - \left(\frac{1}{1+2z}\right) \\ &= \frac{1 + 2z - z^2}{1 + 4z + 5z^2 + 2z^3}. \end{aligned}$$

with $|R_{1,0}(-z)| \leq 1$ for all $z \geq 0$. This implies that the extrapolated form of (1,0) method is stable for all $\tau > 0$; that is, the method is A_0 -stable. Moreover,

$$\lim_{z \rightarrow \infty} R_{1,0}(-z) = 0$$

so that (3.1.6) is L_0 – stable.

In Fig. 3.1 we have plotted the amplification symbols $R(-z)$ against z for the backward-difference scheme, (graph 1), the Crank-Nicolson method (graph 2) and the novel extrapolated scheme (graph 3). As it can be seen, the asymptotic behaviour of the backward-difference scheme produces a growth factor which tends to zero monotonically. In contrast, the extrapolated scheme has a growth factor which tends to zero asymptotically but has a small negative value for $z > 2(1 + \sqrt{2})$. This implies that small oscillations could theoretically appear but these should be damped with increasing t . The Crank-Nicolson method has a growth factor which tends asymptotically to -1 . Consequently errors which occur during the computation will tend to be damped at a considerably lower rate than the rate at which the analytical solution tends to zero.

3.2. Numerical Results I

We applied the numerical techniques (1.3.5), (1.3.7) and (3.1.6) to the model problem (2.6.1a), (2.6.1b) and (2.6.1c), which has an analytical solution given by (2.6.2). The resulting numerical solutions (with $h = 0.05$) were compared to the analytical one.

The computed solutions at time $t = 1.0$ are depicted in Fig. 3.2. In the figure, curve 1 is the graph of the analytical solution, curve 2 is the graph of the backward-difference solution (1.3.5), curve 3 is the graph of the Crank-Nicolson solution (1.3.7) and curve 4 is the graph of the extrapolated solution (3.1.6). In all these cases the value of the time step $\tau = 0.1$ was used.

As it can be seen from the figure, the Crank-Nicolson scheme is accurate in the interior of the region but has a large residual error near the boundary. If h were to be reduced, keeping τ constant, the amplitude of this residual error would increase. The backward difference scheme portrays a smooth behaviour but is inaccurate uniformly across the interval. In contrast the novel scheme has a smooth representation of the analytical solution over the whole region. It is considerably more accurate than the backward-difference scheme and is comparable with the Crank-Nicolson scheme in the interior of the region. We have tabulated (in Table 1) the maximum

errors (the difference between the analytical and numerical solutions) for three values of h for each of the three methods with $\tau = 0.1$. Note that the actual solution at $x = 1$ and $t = 1$ is about 0.1.

TABLE 1.
Errors in solving the model problem at $t = 1$.

Method	$h = 0.1$	$h = 0.05$	$h = 0.025$
Crank-Nicolson	$.56E - 01$	$.28E + 00$	$.55E + 00$
Backward-difference	$.33E - 01$	$.32E - 01$	$.32E - 01$
Extrapolation method	$.63E - 02$	$.61E - 02$	$.60E - 02$

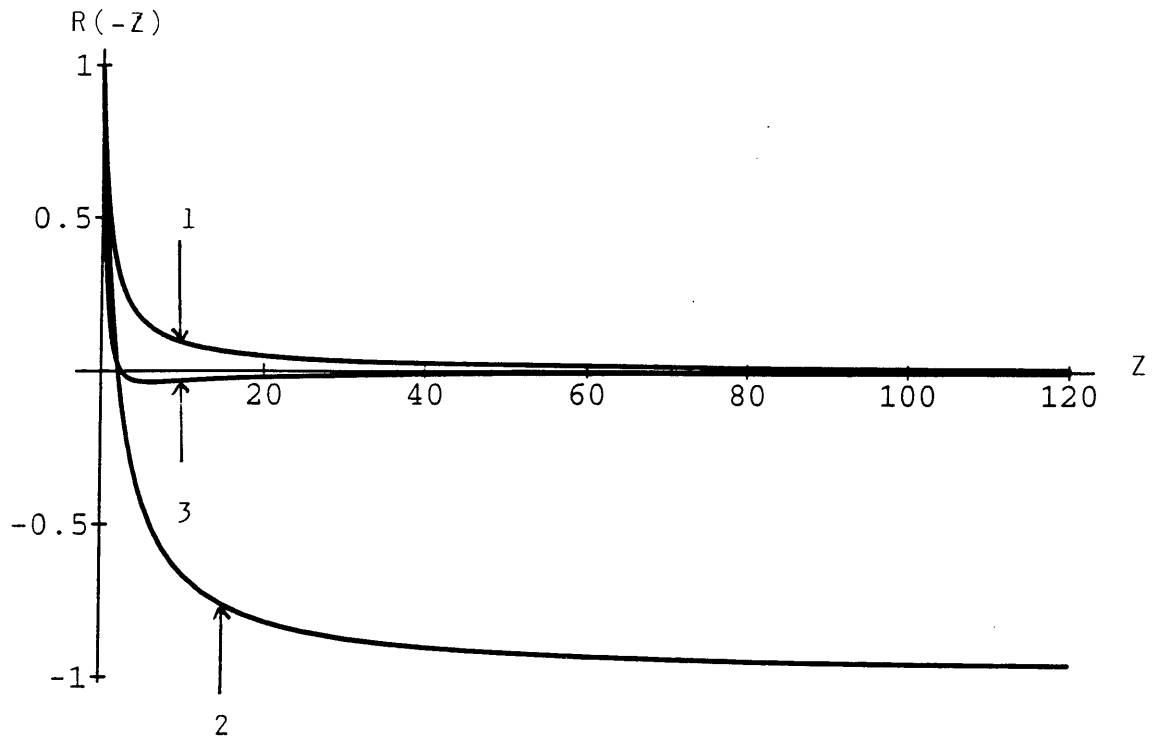


FIG. 3.1. Amplification factors for the schemes : Backward-difference (1) , Crank-Nicolson (2) , and extrapolation (3) .

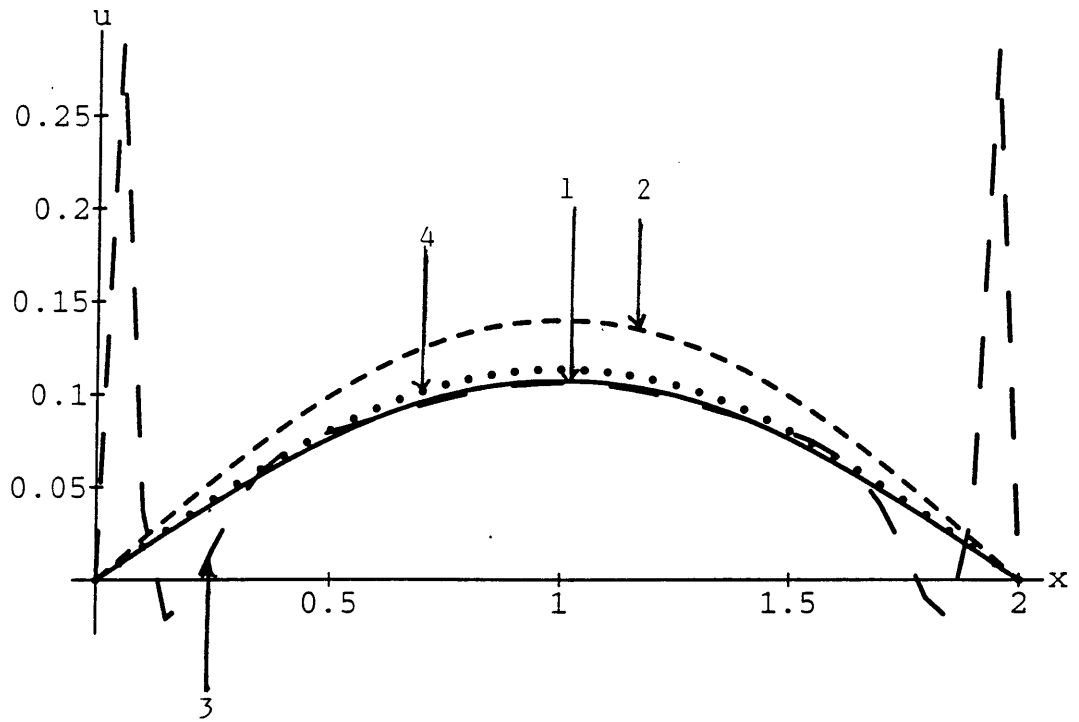


FIG. 3.2. Model problem solutions with $h = 0.05$ and $\tau = 0.1$: Fourier sum (1) , backward-difference (2) , Crank-Nicolson (3) , and extrapolation (4) .

3.3 The Extrapolated Method in Two Space Variables

In this section, the novel algorithm is generalized to two space variables. Thus, consider the simple heat equation in two space variables

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad 0 < x, y < 2, \quad (3.3.1)$$

with homogeneous Dirichlet boundary conditions ($u \equiv 0$) on the boundary $\delta\Omega$ of the square defined by the lines $x = 0, y = 0, x = 2, y = 2$, and initial conditions $u(x, y, 0) = g(x, y); (x, y) \in \Omega \times \delta\Omega$. Assuming further that $g(x, y) \neq 0$ for $(x, y) \in \delta\Omega$, so that discontinuities between initial conditions and boundary conditions may exist.

Replacing space derivatives in (3.3.1) by the corresponding central differences

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{h^2} u(x - h, y, t) - 2u(x, y, t) + u(x + h, y, t) + O(h^2),$$

and

$$\frac{\partial^2 u}{\partial y^2} = \frac{1}{h^2} u(x, y - h, t) - 2u(x, y, t) + u(x, y + h, t) + O(h^2),$$

we obtain a system of ordinary differential equations of the form

$$\frac{dv}{dt} = Av(t) \quad (3.3.2)$$

where the vector v of unknowns is of dimension N^2 , and the matrix A is $N^2 \times N^2$ given by

$$h^{-2} \begin{bmatrix} B & I & & & \\ I & B & I & & O \\ & \ddots & \ddots & \ddots & \\ & & I & B & I \\ O & & & I & B \end{bmatrix}.$$

where B is the tridiagonal matrix of order N given by

$$B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ 0 & & & 1 & -4 \end{bmatrix}.$$

and I is the identity matrix of order N .

The solution of (3.3.2) is given by

$$\begin{aligned} v(t + \tau) &= \exp(\tau A)v(t) \\ \text{where, } v(0) &= g, \quad \text{the vector of initial values.} \end{aligned} \tag{3.3.3}$$

As in section 1.3 a number of finite difference methods are obtained from (3.3.3) by replacing the exponential by its Padé approximations

The backward-difference approximation is obtained by using (1,0) Padé approximations . But, this involves solving systems of equations whose coefficient matrix has band width $2N$, only a few of its elements per row are nonzero. Consequently, the matrix A is split into constituent matrices A_1 and A_2 such that $A = A_1 + A_2$. The matrix A_1 arises from the use of the central difference in the x-direction, it is block diagonal with tridiagonal blocks and has the form :

$$h^{-2} \begin{bmatrix} C & & & & \\ & C & & & \\ & & \ddots & & \\ & & & C & \\ 0 & & & & C \end{bmatrix}.$$

where C is the tridiagonal matrix of order N given by

$$C = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{bmatrix}.$$

The matrix A_2 arises from the use of the central difference in the y-direction. It is block tridiagonal with diagonal blocks and has the form

$$h^{-2} \begin{bmatrix} -2I & I & & & \\ I & -2I & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & -2I & I \\ 0 & & & I & -2I \end{bmatrix},$$

where I is the identity matrix of order N .

Hence, (3.3.3) becomes

$$v(t + \tau) = \exp[\tau(A_1 + A_2)]v(t) \quad (3.3.4)$$

which may be approximated by

$$v(t + \tau) = \exp(\tau A_2) \exp(\tau A_1)v(t), \quad (3.3.5)$$

or

$$v(t + \tau) = \exp(\tau A_1) \exp(\tau A_2)v(t), \quad (3.3.5')$$

incurring an error $O(\tau^2)$. For each of the exponentials in (3.3.5), (1,0) Padé approximations then are proposed

$$v(t + \tau) = (I - \tau A_2)^{-1}(I - \tau A_1)^{-1}v(t)$$

If we let

$$v^* = (I - \tau A_1)^{-1}v(t)$$

then

$$v(t + \tau) = (I - \tau A_2)^{-1}v^*$$

hence, a split form of the implicit scheme is

$$(I - \tau A_1)v^* = v(t), \quad (I - \tau A_2)v(t + \tau) = v^* \quad (3.3.6)$$

where v^* is an intermediate vector.

This two step procedure has the advantage of being easily solved, when the components of the second step of (3.3.6) are reordered. In particular, if the second order derivatives in (3.3.1) are replaced by central differences then A is an $N^2 \times N^2$ matrix which has 5 nonzero bands. The matrices A_2 and A_1 may be written as tensor products of $N \times N$ tridiagonal matrices so that (3.3.6) may be solved by implementing sequences of tridiagonal matrix solvers.

The algorithm is first order accurate in time, unconditionally stable and has a growth factor which tends asymptotically to zero. Hence no residual boundary errors will be exhibited and the method is L_0 - stable.

In contrast to (3.3.5), (3.3.4) may be approximated to $O(\tau^2)$ accuracy by

$$v(t + \tau) = \exp\left(\frac{\tau}{2}A_1\right) \exp(\tau A_2) \exp\left(\frac{\tau}{2}A_1\right)v(t).$$

Then, by approximating the matrix exponential by (1,0), (1,1) and (0,1) Padé approximations , we get

$$v(t + \tau) = (I - \frac{\tau}{2}A_1)^{-1}(I + \frac{\tau}{2}A_2)(I - \frac{\tau}{2}A_2)^{-1}(I + \frac{\tau}{2}A_1)v(t)$$

Let

$$v^* = (I - \frac{\tau}{2}A_2)^{-1}(I + \frac{\tau}{2}A_1)v(t)$$

we get

$$v(t + \tau) = (I - \frac{\tau}{2}A_1)^{-1}(I + \frac{\tau}{2}A_2)v^*$$

Hence, we get the following implicit method

$$\begin{aligned} \left(I - \frac{\tau}{2}A_2\right)v^* &= \left(I + \frac{\tau}{2}A_1\right)v(t), \\ \left(I - \frac{\tau}{2}A_1\right)v(t + \tau) &= \left(I + \frac{\tau}{2}A_2\right)v^* \end{aligned} \tag{3.3.7}$$

Which is known as the Peaceman-Rachford method [13].

This algorithm is second order accurate in time, and the one dimensional amplification symbol of the method is given by

$$R_{1,1}(-z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}$$

with $|R(-z)| < 1$ for all $z > 0$ and $\lim_{z \rightarrow \infty} R(-z) = -1$. This implies that the method is unconditionally stable (A_0 -stable) but has the same restriction on the time step τ as in the case of the Crank-Nicolson method.

To obtain the two dimensional analogue of the extrapolated scheme described in the previous section, consider (3.3.6) written over two single steps as

$$\begin{aligned} (I - \tau A_1)v^* &= v(t), \quad (I - \tau A_2)v(t + \tau) = v^* \\ (I - \tau A_2)v^{**} &= v(t + \tau), \quad (I - \tau A_1)v^{(0)}(t + 2\tau) = v^{**}. \end{aligned} \tag{3.3.8}$$

System (3.3.8) is a symmetrized application of the method (3.3.6) in that the order in which the matrices A_1 and A_2 appear in the tridiagonal solutions alternates. Eliminating the intermediate vectors in (3.3.8) gives

$$\begin{aligned} v^{(0)}(t+2\tau) &= (I - \tau A_1)^{-1}(I - \tau A_2)^{-1}(I - \tau A_2)^{-1}(I - \tau A_1)^{-1}v(t) \\ &= (I - \tau A_1)^{-1}(I - \tau A_2)^{-2}(I - \tau A_1)^{-1}v(t) \end{aligned}$$

and expanding the matrix inverses to $O(\tau^2)$ we find

$$v^{(0)}(t+2\tau) = (I + \tau A_1 + \tau^2 A_1^2)(I + 2\tau A_2 + 3\tau^2 A_2^2)(I + \tau A_1 + \tau^2 A_1^2) + O(\tau^3)$$

which leads to

$$\begin{aligned} v^{(0)}(t+2\tau) &= [I + 2\tau(A_1 + A_2) + \tau^2(3A_1^2 + 3A_2^2 + \\ &\quad 2A_1A_2 + 2A_2A_1)]v(t) + O(\tau^3). \end{aligned} \quad (3.3.9)$$

If the Maclaurin expansion of $\exp(2\tau(A_1 + A_2))$ in (3.3.4) is written over a time interval 2τ we get

$$\begin{aligned} v(t+2\tau) &= [I + 2\tau(A_1 + A_2) + 2\tau^2(A_1^2 + A_2^2 \\ &\quad + A_1A_2 + A_2A_1)]v(t) + O(\tau^3). \end{aligned} \quad (3.3.10)$$

so that (3.3.8) is $O(\tau^2)$ accurate.

Writing equations (3.3.5), (3.3.5') over a time interval 2τ we get

$$v^{(1)}(t+2\tau) = \exp(2\tau A_1) \exp(2\tau A_2)v(t)$$

and

$$v^{(2)}(t+2\tau) = \exp(2\tau A_2) \exp(2\tau A_1)v(t)$$

then the matrix exponential functions are all replaced by (1,0) Padé approximations, giving,

$$v^{(1)}(t+2\tau) = (I - 2\tau A_1)^{-1}(I - 2\tau A_2)^{-1}v(t) \quad (3.3.11)$$

and

$$v^{(2)}(t+2\tau) = (I - 2\tau A_2)^{-1}(I - 2\tau A_1)^{-1}v(t). \quad (3.3.12)$$

Expanding the inverses in (3.3.11) and (3.3.12) to $O(\tau^2)$ we find that

$$\begin{aligned} v^{(1)}(t + 2\tau) &= (I + 2\tau A_1 + 4\tau^2 A_1^2)(I + 2\tau A_2 + 4\tau^2 A_2^2)v(t) + O(\tau^3) \\ &= [I + 2\tau(A_1 + A_2) + 4\tau^2(A_1^2 + A_1 A_2 + A_2^2)]v(t) + O(\tau^3). \end{aligned} \quad (3.3.13)$$

and

$$\begin{aligned} v^{(2)}(t + 2\tau) &= (I + 2\tau A_2 + 4\tau^2 A_2^2)(I + 2\tau A_1 + 4\tau^2 A_1^2)v(t) + O(\tau^3) \\ &= [I + 2\tau(A_1 + A_2) + 4\tau^2(A_1^2 + A_2 A_1 + A_2^2)]v(t) + O(\tau^3). \end{aligned} \quad (3.3.14)$$

Comparing (3.3.13), (3.3.14) with (3.3.10) it is seen that, in addition to $v^{(0)}(t + 2\tau)$, the approximations $v^{(1)}(t + 2\tau)$ and $v^{(2)}(t + 2\tau)$ are first order accurate in time, their errors being $O(\tau^2)$.

Now, consider the linear combination of the three first order schemes (3.3.8), (3.3.11) and (3.3.12), defined by

$$v(t + 2\tau) = 2v^{(0)} - \frac{1}{2}(v^{(1)} + v^{(2)}) \quad (3.3.15)$$

Applying this weighting to the expansions in (3.3.9), (3.3.13) and (3.3.14) and comparing with (3.3.10) we find that $v(t + 2\tau)$ defined by (3.3.15) is second order accurate. That is, we have extrapolated the first order methods to achieve second order accuracy.

This novel method (3.3.15) is unconditionally stable and has a growth factor similar to the one dimensional analogue

The novel algorithm requires 4 tridiagonal solutions per time step. This compares with the two required by the Peaceman-Rachford method. However, the time step τ is unrestricted in the novel scheme and consequently we may take time steps (at least) twice as large as the Peaceman-Rachford scheme, for a given h .

3.4. Numerical Results II

To demonstrate the numerical properties of the novel algorithm we computed the solution of the problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad 0 < x, y < 2, \quad (3.4.1)$$

subject to the initial conditions

$$u(x, y, 0) = \sin(\pi y/2), \quad 0 \leq x, y \leq 2;$$

and the boundary conditions

$$u(x, y, 0) = 0, \quad \forall (x, y) \in \delta\Omega, \quad \forall t > 0.$$

The initial function is shown in Fig. 3.3 The Fourier solution of this problem is

$$u(x, y, t) = \sin \frac{\pi}{2} y \sum_{n=1}^{\infty} [1 - (-1)^n] \frac{2}{n\pi} \sin\left(\frac{n\pi x}{2}\right) \exp\left(\frac{-\pi^2}{4}(n^2 + 1)t\right). \quad (3.4.2)$$

The maximum value of u at time $t = 1.0$ sec occurs for $x = 1$, $y = 1$ and is approximately 0.00916.

The solution was computed using the backward-difference scheme (3.3.8), the Peaceman-Rachford scheme (3.3.7) and the novel algorithm $\{(3.3.8), (3.3.11), (3.3.12) \text{ and } (3.3.15)\}$ using a fixed time step $\tau = 0.1$. The analytical solution at $t = 1.0$ sec is depicted in Fig. 3.4. The solution due to the backward-difference scheme (3.3.8) is depicted in Fig. 3.5 using $h = 0.05$. The general appearance of the solution is good although the inaccuracy due to the first order accuracy of the scheme is apparent as shown by results in Table 2.

TABLE 2.
*Errors in solving the two-dimensional
model problem at $t = 1$.*

Method	$h = 0.1$	$\tau = 0.1$	
		$h = 0.05$	$h = 0.025$
Backward-difference	$.64E - 02$	$.63E - 02$	$.63E - 02$
Peaceman-Rachford	$.47E - 02$	$.23E - 01$	$.46E - 01$
Novel scheme	$.94E - 03$	$.88E - 03$	$.87E - 03$

Figure 3.6 shows the results obtained by the Peaceman-Rachford scheme using $h = 0.05$.

It should be emphasized that the central portion of this figure is an accurate approximation to the solution so that rescaling has apparently decreased this portion of the solution in order to show the large peaks adjacent to the boundary on a graph of the same size. The numerical results obtained using the novel algorithm are depicted in Fig. 3.7 using $h = 0.05$. The maximum errors which occurred are summarized in Table 2.

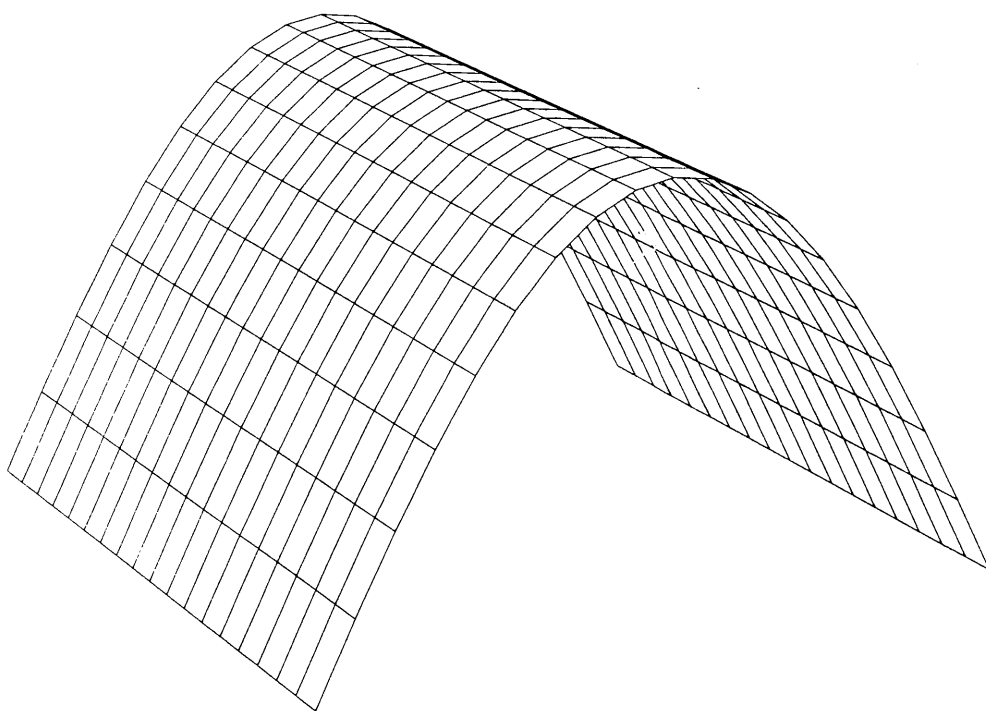


FIG. 3.3. *Initial data for the two-dimensional model problem.*

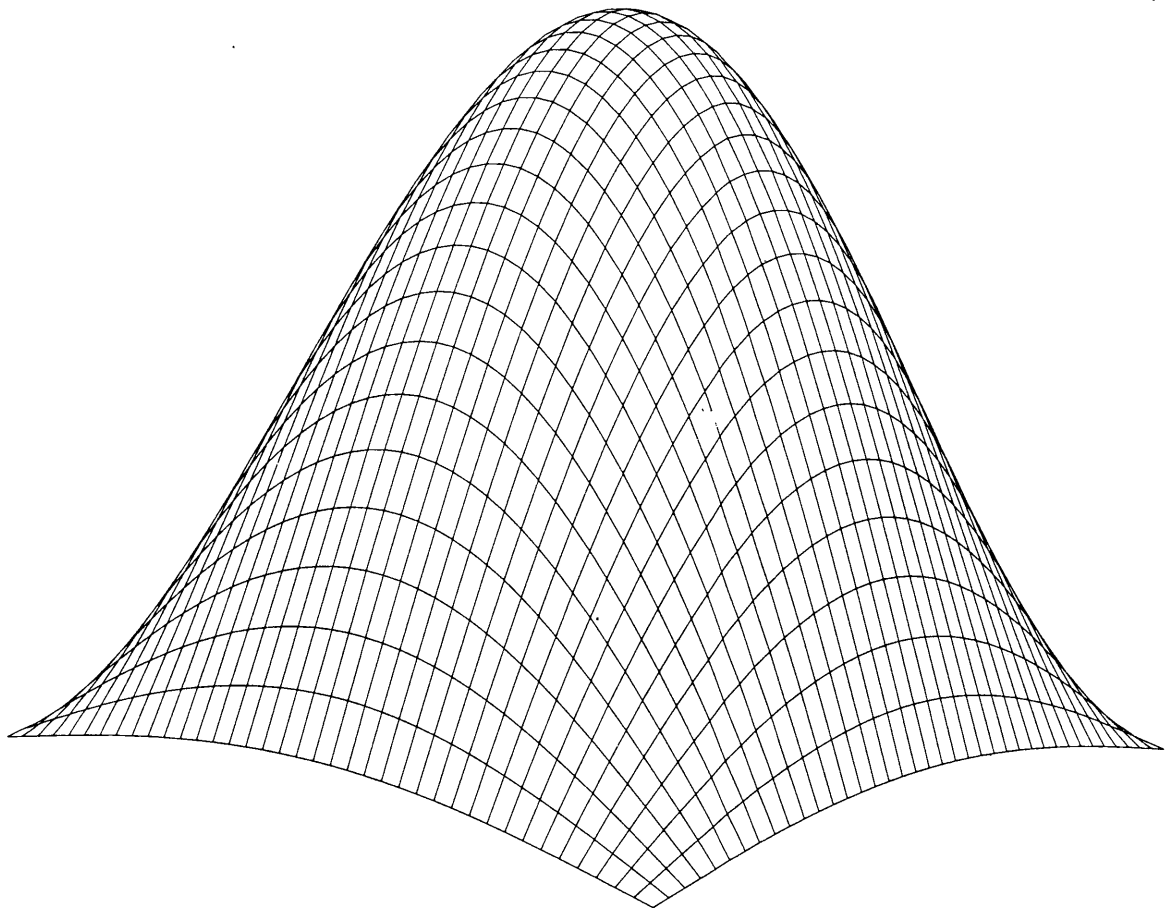


FIG. 3.4. *Analytical solution to the model problem at $t = 1.0$.*

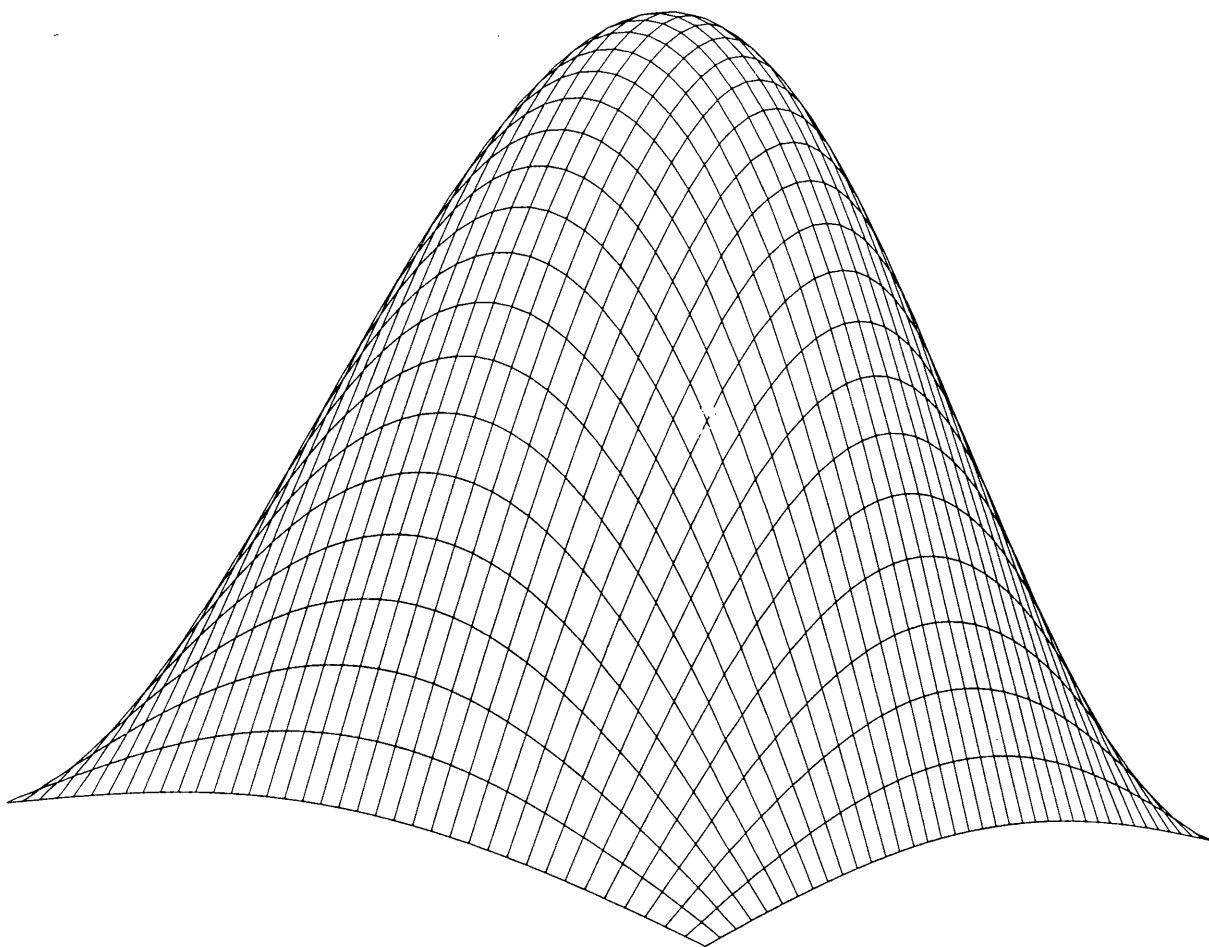


FIG. 3.5. *Backward-difference solution with $h = 0.05$ and $\tau = 0.1$.*

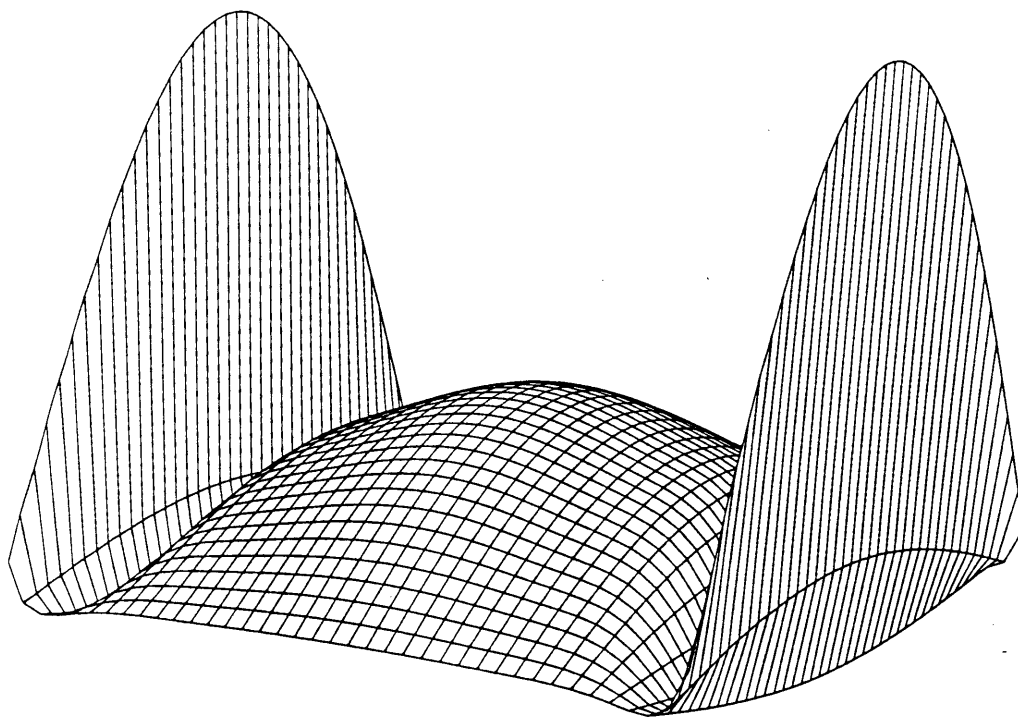


FIG. 3.6. *Peaceman-Rachford solution with $h = 0.05$ and $\tau = 0.1$.*

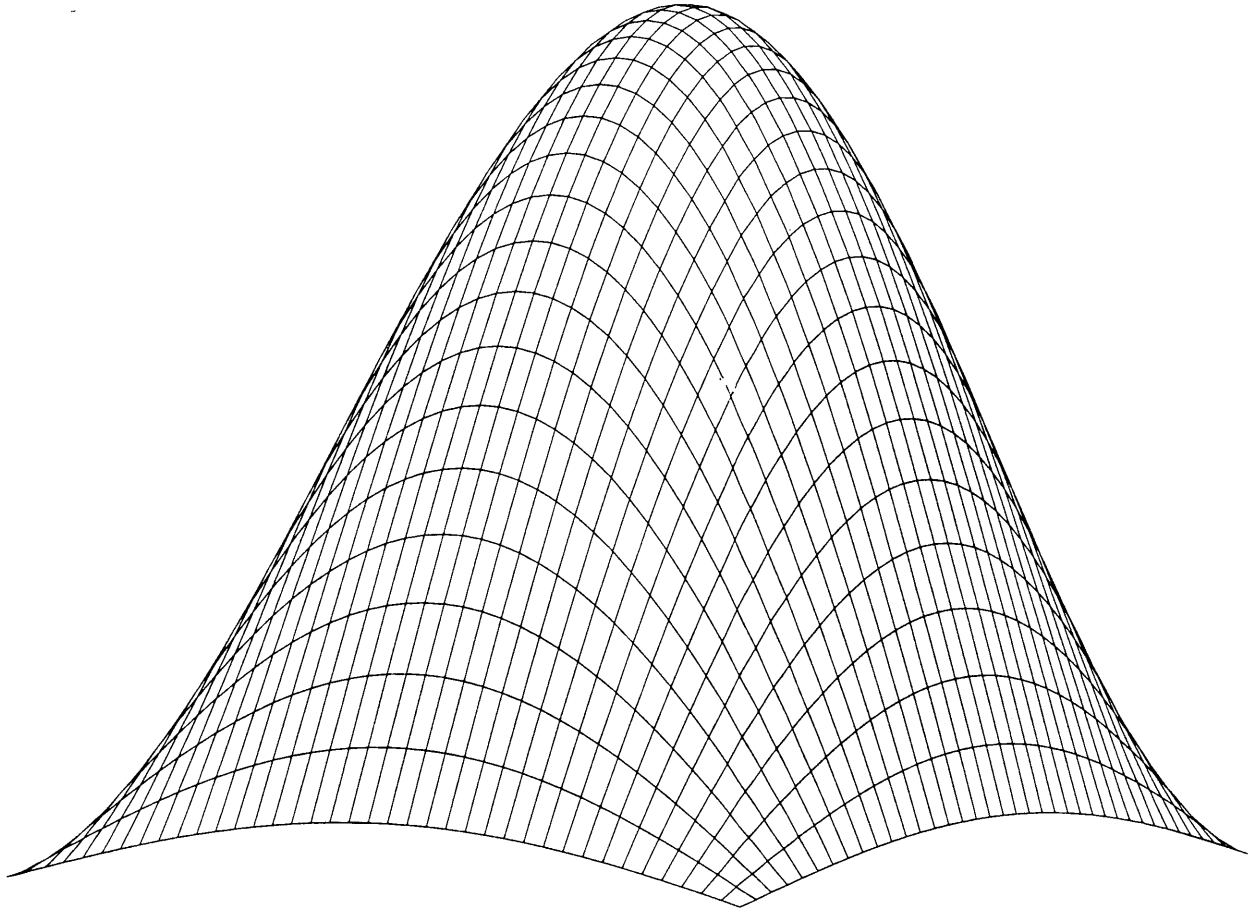


FIG. 3.7. *Extrapolation solution with $h = 0.05$ and $\tau = 0.1$.*

CHAPTER IV

TIME-DEPENDENT BOUNDARY AND INTERIOR FORCING IN LOCALLY ONE-DIMENSIONAL SCHEMES

4.1 Introduction

In this chapter we investigate some of the reasons for the difficulty of maintaining the accuracy in the numerical solutions of parabolic equations with time-dependent Dirichlet boundary conditions, when locally one-dimensional (LOD) techniques are employed, see for example, Swayne [18],[19].

The investigation arose from consideration of an alternative formulation for LOD algorithms for the solution of linear differential equations such as those used in the approximation of the heat equation, $u_t = \nabla^2 u$, by the method of lines (see Zafarullah [24]). For such equations, approximations may be derived by superimposing a uniform grid over the domain of interest. On such a grid, the spatial operator is replaced by a finite difference operator which produces an ordinary differential equation $(d/dt)u = Lu$. Thus over the given domain, the semidiscrete solution u is given by

$$u(t + \tau) = \exp(\tau L) u(t). \quad (4.1.1)$$

where τ is some convenient time-step.

It is possible to split the matrix L into two or more constituents, $L = L_1 + L_2$, where L_1 and L_2 result in physically meaningful problems which can be solved and recombined into an approximate solution to (4.1.1). One must assume that $L_1 L_2 \neq L_2 L_1$. This strategy has a payoff whenever the numbers of operations and the associated computing overhead for the one-dimensional strategy produces numerical solutions with the same accuracy in a shorter time than similar numerical methods associated with the “full” matrix

L. LOD methods are now classical (see Gourlay and Mitchell [5], Mitchell and Griffiths [10], and Yanenko [23]). However, it is known in the literature that time-dependent Dirichlet boundary conditions can reduced the efficiency of LOD methods. Methods which have local truncation error $O(\tau^3)$, are known to show sever deterioration in the error constant as far as $O(\tau^{0.25})$ (Yanenko, [23]) from the influence of time-dependent boundary conditions.

Swayne [18] proposed an approach which corrects the past defect, and permits flexibility in the evaluation of the forcing terms. There are many cases for which freedom in the location of forcing evaluation is advantageous. The forcing terms can only evaluated for main time-point $\{t_i, i = 1, \dots, N\}$ where accurate values of u are to be calculated.

The Process of developing numerical schemes involves the replacement of $\exp(\tau L)$ in (4.1.1) by combinations of $\exp(\tau L_1)$ and $\exp(\tau L_2)$ in such a way as to obtain order in the approximation to $\exp(\tau L)$. For example, it is known that

$$\exp(\tau L) = \frac{1}{2} [\exp(\tau L_1) \exp(\tau L_2) + \exp(\tau L_2) \exp(\tau L_1)] + O(\tau^3). \quad (4.1.2)$$

We say that (4.1.2) defines a **second-order** approximation to $\exp(\tau L)$ because $O(1)$, $O(\tau)$, and $O(\tau^2)$ terms match on both sides of (4.1.2).

A more common LOD splitting, defined by

$$\exp(\tau L) = \exp\left(\frac{\tau}{2} L_1\right) \exp\left(\frac{\tau}{2} L_2\right) \exp\left(\frac{\tau}{2} L_2\right) \exp\left(\frac{\tau}{2} L_1\right) + O(\tau^3), \quad (4.1.3)$$

results in numerical schemes which have poor convergence as $\tau \rightarrow 0$ near the boundaries of the solution domain when forcing is time dependent. Interior forcing terms are integrated to full accuracy. The schemes derived from (4.1.2) have the opposite shortcoming. Boundary conditions are accurately incorporated into the numerical solution. It is, rather, the forcing terms affecting equations in the interior of the solution domain which behave poorly. A scheme based on (4.1.2) is called **parallel alternating LOD (or PALOD)** method.

4.2 An Example of Boundary Forcing

In this section time-dependent Dirichlet boundary conditions for the linear heat conduction equation is considered.

Consider

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x, y, t)$$

with specified initial and boundary conditions. MOL reduces the PDE into inhomogeneous system of ODEs

$$\frac{d}{dt}u = Lu + f(t), \quad (4.2.1)$$

which has the following solution :

$$u(t) = \exp(Lt) \int_0^t \exp(-L\theta) f(\theta) d\theta.$$

After a time step τ the solution $u(t)$ becomes

$$u(t + \tau) = \exp(L(t + \tau)) \int_0^{t+\tau} \exp(-L\theta) f(\theta) d\theta$$

which can be written as

$$\begin{aligned} u(t + \tau) &= \exp(L(t + \tau)) \left[\int_0^t \exp(-L\theta) f(\theta) d\theta + \int_t^{t+\tau} \exp(-L\theta) f(\theta) d\theta \right] \\ &= \exp(\tau L) u(t) + \exp(L(t + \tau)) \int_t^{t+\tau} \exp(-L\theta) f(\theta) d\theta \end{aligned}$$

Now, let $\theta = t + s\tau$, we get

$$u(t + \tau) = \exp(\tau L) u(t) + \tau \exp(L(t + \tau)) \int_0^1 \exp(-L(t + s\tau)) f(t + s\tau) ds$$

so (4.2.1) has the following integral representation

$$u(t + \tau) = \exp(\tau L) u(t) + \tau \int_0^1 \exp(\tau(1 - s)L) f(t + s\tau) ds, \quad (4.2.2)$$

where $u(t)$ is known and $u(t + \tau)$ is to be computed.

The first term on the right-hand side of (4.2.2) is called the **complementary function** and it represents the solution of the unforced or **homogeneous** problem ($f = 0$). The second term of (4.2.2) is called the **particular integral** which satisfies the forced or **inhomogeneous** differential equation ($f \neq 0$). The known boundary values for the heat equation are incorporated into inhomogeneous terms through change in the normal definition of the coefficient matrix L and the complementary function term is given boundary values identically zero.

A simple domain is considered with a uniform grid, mesh size h , superimposed along (x, y) coordinate directions. The usual central difference approximation to $\nabla^2 u$ is applied on the grid to give a spatial semidiscretization of the original equation.

The discretized differential equation is :

$$\frac{d}{dt} U_{ij} = \frac{1}{h^2} \left(U_{i-1,j} - 2U_{ij} + U_{i+1,j} \right) + \frac{1}{h^2} \left(U_{i,j-1} - 2U_{ij} + U_{i,j+1} \right) + f(t, x_i, y_j)$$

The unknowns $\{U_{ij}, i = 1, 2, j = 1, 2\}$ together with the boundary values are laid out in a table thus :

$$\begin{pmatrix} . & U_{13} & U_{23} & . \\ U_{02} & U_{12} & U_{22} & U_{32} \\ U_{01} & U_{11} & U_{21} & U_{31} \\ . & U_{10} & U_{20} & . \end{pmatrix}.$$

If U_{ij} is a boundary point, the boundary condition is given by $f(t, x_i, y_j) = f_{ij}(t)$. If we organize the solution components into interior points

$$U_I = (U_{11}, U_{21}, U_{12}, U_{22})^T$$

and boundary points

$$U_{1,b} = (U_{01}, U_{31}, U_{02}, U_{32})^T, \quad U_{2,b} = (U_{10}, U_{20}, U_{13}, U_{23})^T$$

and define U by

$$U = \begin{pmatrix} U_I \\ U_{1,b} \\ U_{2,b} \end{pmatrix}.$$

The discretized differential equation can be written as

$$\frac{d}{dt}\mathbf{U} = \mathbf{L}(\mathbf{U} + f(t)), \quad (4.2.3)$$

with initial conditions given by

$$U(0) = (U_I(0)^T 0_{1 \times 8})^T. \quad (4.2.4)$$

The coefficient matrix $\mathbf{L}$ is built up from

$$\mathbf{L}^{11} = h^{-2} \begin{pmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{pmatrix} \quad (4.2.5)$$

and

$$\mathbf{L}^{12} = h^{-2} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.6)$$

Then

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}^{11} & \mathbf{L}^{12} \\ 0_{8 \times 12} & \end{pmatrix}$$

The forcing function is made up of

$$\mathbf{f}_1 = (0_{1 \times 4}, f_{01}, f_{31}, f_{02}, f_{32}, 0_{1 \times 4})^T$$

and

$$\mathbf{f}_2 = (0_{1 \times 8}, f_{10}, f_{20}, f_{13}, f_{23})^T,$$

and $\mathbf{f}$ is defined by

$$\mathbf{f}(t) = \mathbf{f}_1(t) + \mathbf{f}_2(t). \quad (4.2.7)$$

$\mathbf{L}_1$ is defined by

$$\mathbf{L}_1^{11} = h^{-2} \begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix}, \quad \mathbf{L}_1^{12} = h^{-2}(\mathbf{I}_{4 \times 4} 0_{4 \times 4})$$

and

$$\mathbf{L}_1 = \begin{pmatrix} \mathbf{L}_1^{11} & \mathbf{L}_1^{12} \\ 0_{8 \times 12} & \end{pmatrix}.$$

Finally, $L_2 = L - L_1$. We have the following :

$$Lf = L_1f_1 + L_2f_2, \quad (4.2.8)$$

and

$$L_1f_2 = L_2f_1 = 0. \quad (4.2.9)$$

Split (4.2.3) into two one-dimensional equations

$$\frac{1}{2} \frac{d}{dt} U = L_1(U + f_1), \quad (4.2.10)$$

and

$$\frac{1}{2} \frac{d}{dt} U = L_2(U + f_2). \quad (4.2.11)$$

This splitting dose not result in an increase in the effective size of the computation. We have not added m unknowns to the problem (m being the number of boundary points). Rather we are attempting to simplify splitting of some of the (known) boundary values in the one-dimensional subproblems.

Definition 4.2.1 Swayne [18] For a differential equation which may be split into locally one dimensional subproblems, **separable boundary forcing** is defined to be a forcing function $f(t)$ in $(d/dt)U = L(U+f(t))$ for which $L = L_1+L_2$ and $f(t) = f_1(t)+f_2(t)$ satisfy (4.2.8) and (4.2.9).

4.3 One Dimensional Approximations

Numerical methods from (4.2.2) are constructed by replacing the integral by a finite sum (quadrature rule). We write

$$\tau \int_0^1 \exp(\tau(1-s)L)f(t+s\tau)ds = \tau \left[W_1f(t+\tau s_1) + W_2f(t+\tau s_2) \right] + O(\tau^3). \quad (4.3.1)$$

Then the exponential in (4.2.2) is replaced by an approximation $E(\tau L)$, to give

$$U(t + \tau) = E(\tau L)U(t) + \tau [W_1f(t + \tau s_1) + W_2f(t + \tau s_2)]. \quad (4.3.2)$$

In order to derive a quadrature rule for the particular points $\{s_1, s_2\}$, we let $f(s) = 1$ and $f(s) = s$ in (4.3.1). The matrix argument τL is replaced by the scalar parameter z . Then if $f(s) = 1$, (4.3.1) gives

$$W_1(z) + W_2(z) = z^{-1}(\exp(z) - 1), \quad (4.3.3a)$$

and if $f(s) = s$, (4.3.1) gives

$$s_1 W_1(z) + s_2 W_2(z) = z^{-1} \left(z^{-1}(\exp(z) - 1) - 1 \right). \quad (4.3.3b)$$

Now let

$$M_0(z) = z^{-1}(\exp(z) - 1), \quad (4.3.4a)$$

and

$$M_1(z) = z^{-1}(M_0(z) - 1), \quad (4.3.4b)$$

so (4.3.3a) and (4.3.3b) become :

$$\begin{aligned} W_1(z) + W_2(z) &= M_0(z), \\ s_1 W_1(z) + s_2 W_2(z) &= M_1(z) \end{aligned} \quad (4.3.4')$$

Any exponential approximation $E(z)$ which is at least second order in z may be used to develop a method of LOD type using formula (4.1.2). The exponential approximation $P_{11}(z) = (1 - z/2)^{-1}(1 + z/2)$ will be examined. Replacing $E(z)$ in (4.3.4) by P_{11} we get

$$M_0(z) = z^{-1} \left[(1 - z/2)^{-1}(1 + z/2) - 1 \right] = \left(1 - \frac{z}{2} \right)^{-1},$$

and

$$M_1(z) = z^{-1} \left[\left(1 - \frac{z}{2} \right)^{-1} - 1 \right] = \frac{1}{2} \left(1 - \frac{z}{2} \right)^{-1}$$

The numerators of M_0 and M_1 are independent of the matrix argument, and so if we solve (4.3.4') we see that P_{11} is a special case for which $s_1 = s_2 = \frac{1}{2}$.

For the midpoint rule together with P_{11} , the integral in (4.3.1) becomes :

$$\begin{aligned} \tau \int_0^1 \exp(\tau(1-s)L) f(t+s\tau) ds &= \tau \left[W_1 f\left(t + \frac{\tau}{2}\right) + W_2 f\left(t + \frac{\tau}{2}\right) \right] \\ &= \tau \left[W_1 + W_2 \right] f\left(t + \frac{\tau}{2}\right) \\ &= \tau M_0(z) f\left(t + \frac{\tau}{2}\right) \\ &= \tau \left(1 - \frac{z}{2} \right)^{-1} f\left(t + \frac{\tau}{2}\right) \end{aligned}$$

hence (4.3.2) gives :

$$U(t + \tau) = (I - \frac{\tau}{2}L)^{-1}(I + \frac{\tau}{2}L)U(t) + \tau(I - \frac{\tau}{2}L)^{-1} f(t + \frac{\tau}{2})$$

which can be written in the form,

$$(I - \frac{\tau}{2}L)U(t + \tau) = (I + \frac{\tau}{2}L)U(t) + \tau f(t + \frac{\tau}{2}). \quad (4.3.5)$$

Noting that $(I - \frac{\tau}{2}L)^{-1}(I + \frac{\tau}{2}L) \equiv 2(I - \frac{\tau}{2}L)^{-1} - I$, then $U(t + \tau)$ becomes :

$$\begin{aligned} U(t + \tau) &= \left[2(I - \frac{\tau}{2}L)^{-1} - I \right] U(t) + \tau(I - \frac{\tau}{2}L)^{-1} f(t + \frac{\tau}{2}) \\ &= 2(I - \frac{\tau}{2}L)^{-1}U(t) - U(t) + \tau(I - \frac{\tau}{2}L)^{-1} f(t + \frac{\tau}{2}). \end{aligned}$$

If we let

$$U^* = (I - \frac{\tau}{2}L)^{-1}U(t) + \frac{\tau}{2}(I - \frac{\tau}{2}L)^{-1} f(t + \frac{\tau}{2})$$

then (4.3.5) can be written as :

$$(I - \frac{\tau}{2}L)U^* = U(t) + \frac{\tau}{2}f(t + \frac{\tau}{2}), \quad U(t + \tau) = 2U^* - U(t). \quad (4.3.6)$$

We may also apply $s_1 = 0$ and $s_2 = 1$ (a “trapezoidal rule ”) then (4.3.4') gives

$$\begin{aligned} W_2 = M_1 &= \frac{1}{2} \left(I - \frac{\tau}{2}L \right)^{-1} \\ W_1 = M_0 - M_1 &= \frac{1}{2} \left(I - \frac{\tau}{2}L \right)^{-1} \end{aligned}$$

and (4.3.1) gives

$$\begin{aligned} \tau \int_0^1 \exp(\tau(1-s)L) f(t + s\tau) ds &= \tau \left[W_1 f(t) + W_2 f(t + \tau) \right] \\ &= \tau \left[\frac{1}{2} (I - \frac{\tau}{2}L)^{-1} f(t) + \frac{1}{2} (I - \frac{\tau}{2}L)^{-1} f(t + \tau) \right] \\ &= \frac{\tau}{2} (I - \frac{\tau}{2}L)^{-1} \left[f(t) + f(t + \tau) \right] \end{aligned}$$

Using this result and the expression of P_{11} in (4.3.2) we get,

$$\begin{aligned} U(t + \tau) &= (I - \frac{\tau}{2}L)^{-1}(I + \frac{\tau}{2}L)U(t) + \frac{\tau}{2}(I - \frac{\tau}{2}L)^{-1}\left[f(t) + f(t + \tau)\right] \\ &= \left[2(I - \frac{\tau}{2}L)^{-1} - I\right]U(t) + \frac{\tau}{2}(I - \frac{\tau}{2}L)^{-1}\left[f(t) + f(t + \tau)\right] \\ &= 2(I - \frac{\tau}{2}L)^{-1}U(t) - U(t) + \frac{\tau}{2}(I - \frac{\tau}{2}L)^{-1}\left[f(t) + f(t + \tau)\right] \end{aligned}$$

Define U^* by

$$U^* = (I - \frac{\tau}{2}L)^{-1}U(t) + \frac{\tau}{4}(I - \frac{\tau}{2}L)^{-1}\left[f(t) + f(t + \tau)\right]$$

Then we have

$$(I - \frac{\tau}{2}L)U^* = U(t) + \frac{\tau}{4}(f(t) + f(t + \tau)), \quad U(t + \tau) = 2U^* - U(t). \quad (4.3.6')$$

Now, the LOD scheme defined by (4.1.2) is implemented using (4.3.5). Following (4.2.10) and (4.2.11), considering matrix arguments for the one-dimensional calculation to be $2L_1$ and $2L_2$, with time step $\frac{\tau}{2}$. The results, for boundary forcing, will be shown to be equivalent, within $O(\tau^3)$, to a two-dimensional two-point rule with $s_1 = \frac{1}{4}$ and $s_2 = \frac{3}{4}$. We proceed, setting $f_i(\alpha) = f_i(t + \alpha\tau)$, $i = 1, 2$.

The algorithm is the following

$$\left(I - \frac{\tau}{2}L_1\right)U_1 = \left(I + \frac{\tau}{2}L_1\right)U(t) + \tau L_1 f_1\left(\frac{1}{4}\right), \quad (4.3.7a)$$

$$\left(I - \frac{\tau}{2}L_2\right)U_2 = \left(I + \frac{\tau}{2}L_2\right)U(t) + \tau L_2 f_2\left(\frac{1}{4}\right), \quad (4.3.7b)$$

$$\left(I - \frac{\tau}{2}L_2\right)U_{21} = \left(I + \frac{\tau}{2}L_2\right)U_1 + \tau L_2 f_2\left(\frac{3}{4}\right), \quad (4.3.7c)$$

$$\left(I - \frac{\tau}{2}L_1\right)U_{12} = \left(I + \frac{\tau}{2}L_1\right)U_2 + \tau L_1 f_1\left(\frac{3}{4}\right), \quad (4.3.7d)$$

and finally,

$$U(t + \tau) = \frac{1}{2}(U_{12} + U_{21}). \quad (4.3.7e)$$

We substitute (4.3.7a)–(4.3.7d) into (4.3.7e) and collect terms containing $U(t)$, $f_i(\frac{1}{4})$, and $f_i(\frac{3}{4})$, respectively, we get

$$U(t + \tau) = EU(t) + \tau \left[T_1 f\left(\frac{1}{4}\right) + T_2 f\left(\frac{3}{4}\right) \right],$$

where

$$\begin{aligned} EU(t) &= E(\tau, L_1, L_2)U(t) \\ &= \frac{1}{2}(P_{11}(\tau L_1)P_{11}(\tau L_2) + P_{11}(\tau L_2)P_{11}(\tau L_1))U(t), \end{aligned} \quad (4.3.8)$$

$$\begin{aligned} T_1 f\left(\frac{1}{4}\right) &= \frac{1}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I + \frac{\tau}{2} L_2 \right) \left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1 f_1\left(\frac{1}{4}\right) \\ &\quad + \frac{1}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I + \frac{\tau}{2} L_1 \right) \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2 f_2\left(\frac{1}{4}\right), \end{aligned} \quad (4.3.9)$$

and

$$T_2 f\left(\frac{3}{4}\right) = \frac{1}{2} \left[\left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1 f_1\left(\frac{3}{4}\right) + \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2 f_2\left(\frac{3}{4}\right) \right]. \quad (4.3.10)$$

We see immediately that formula (4.3.8) defines a particular case of (4.1.2).

Now the moment equations (4.3.3) for (4.3.8) are expanded directly. Because L is a singular matrix, it is necessary to work with $M_0 \tau L$ and $M_1 \tau L^2$.

$M_0 \tau L \equiv E(\tau, L_1, L_2) - I$, is expanded using the partial fraction form of P_{11} :

$$\begin{aligned} E(\tau, L_1, L_2) - I &= \frac{1}{2} \left(2 \left(I - \frac{\tau}{2} L_1 \right)^{-1} - I \right) \left(2 \left(I - \frac{\tau}{2} L_2 \right)^{-1} - I \right) \\ &\quad + \frac{1}{2} \left(2 \left(I - \frac{\tau}{2} L_2 \right)^{-1} - I \right) \left(2 \left(I - \frac{\tau}{2} L_1 \right)^{-1} - I \right) - I, \end{aligned}$$

which simplifies to

$$\begin{aligned}
(E - I) &= 2\left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1} + 2\left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1} \\
&\quad - 2\left(\left(I - \frac{\tau}{2}L_1\right)^{-1} + \left(I - \frac{\tau}{2}L_2\right)^{-1}\right) \\
&= 2\left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1}\left[I - \left(I - \frac{\tau}{2}L_2\right)\right] \\
&\quad + 2\left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1}\left[I - \left(I - \frac{\tau}{2}L_1\right)\right]
\end{aligned}$$

hence,

$$(E - I) = \left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1}\tau L_2 + \left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1}\tau L_1$$

The first moment equation is applied to separable boundary forcing function $f \equiv f(t)$ (Definition 4.2.1) which leads to

$$\begin{aligned}
M_0\tau Lf &= (E - I)f \\
&= \left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1}\tau L_2f + \left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1}\tau L_1f \\
&= \left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1}\tau L_2(f_1 + f_2) \\
&\quad + \left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1}\tau L_1(f_1 + f_2) \\
&= \left(I - \frac{\tau}{2}L_1\right)^{-1}\left(I - \frac{\tau}{2}L_2\right)^{-1}\tau L_2f_2 + \left(I - \frac{\tau}{2}L_2\right)^{-1}\left(I - \frac{\tau}{2}L_1\right)^{-1}\tau L_1f_1.
\end{aligned} \tag{4.3.11}$$

Now $M_1\tau L^2$, which must be derived from $M_1\tau Lf(t)$ is examined. The singularity of L in the formulation of the heat equation, makes it impossible to assume that there exists a $g(t)$ satisfying $Lg = f$. However, for the semidiscretized heat equation, a solution exists for

$$L^2g(t) = Lf(t) \tag{4.3.12}$$

We need an expression for $g(t)$ only for $t-$ values corresponding to s_1 and s_2 , since g is again replaced by f in the derivation of M_1 applied to separable boundary forcing. For a fixed $t-$ value, if g is partitioned into components defined on the boundary points

g_b , and interior points g_I , then g_b may be freely chosen and g_I satisfies $L^{11}L^{11}g_I = L^{12}f_b - L^{11}L^{12}g_b$. Here L^{11} and L^{12} are given by (4.2.5) and (4.2.6), and f_b denotes the restriction of f on the boundary points. Choosing $g_b \equiv 0$, to get $L^{11}L^{11}g_I = L^{12}f_b$.

For separable forcing, we may similarly define components $g_1(t)$ and $g_2(t)$ of g satisfying $L_1^{11}L^{11}g_{1,I} = L_1^{12}f_{1,b}$ and $L_2^{11}L^{11}g_{2,I} = L_2^{12}f_{2,b}$ and note that

$$g(t) = g_1(t) + g_2(t)$$

g_1 and g_2 also satisfy

$$L_1(L_1 + L_2)g_1 = L_1f_1, \quad (4.3.13)$$

$$L_2(L_1 + L_2)g_1 = 0, \quad (4.3.14)$$

$$L_2(L_1 + L_2)g_2 = L_2f_2, \quad (4.3.15)$$

and

$$L_1(L_1 + L_2)g_2 = 0, \quad (4.3.16)$$

from (4.3.13)–(4.3.16) we obtain

$$(L_1L_2 + L_1^2)g = L_1f_1, \quad (4.3.17)$$

and

$$(L_2L_1 + L_2^2)g = L_2f_2, \quad (4.3.18)$$

From (4.3.4b) $M_1 z = M_0(z) - I$. Let $z = \tau L$ then

$$\begin{aligned} M_1 \tau L &= M_0 - I, \\ M_1 \tau L^2 &= (M_0 - I)L, \\ &= M_0 L - L, \\ &= \frac{1}{\tau}(E - I) - L, \\ &= \frac{\tau}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2^2 \\ &\quad + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1^2 \\ &\quad + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1 L_2 + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2 L_1. \end{aligned} \quad (4.3.19)$$

We apply $M_1\tau L^2$ to g to get

$$\begin{aligned} M_1\tau L^2 g &= \frac{\tau}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2^2 g \\ &\quad + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1^2 g \\ &\quad + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} L_1 L_2 g + \frac{\tau}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} L_2 L_1 g. \end{aligned}$$

Using the following substitutions deduced from (4.3.17) and (4.3.18)

$$L_1^2 g = \frac{1}{2} L_1^2 g + \frac{1}{2} L_1 f - \frac{1}{2} L_1 L_2 g,$$

$$L_1 L_2 g = \frac{1}{2} L_1 L_2 g + \frac{1}{2} L_1 f - \frac{1}{2} L_1^2 g,$$

$$L_2^2 g = \frac{1}{2} L_2^2 g + \frac{1}{2} L_2 f - \frac{1}{2} L_2 L_1 g,$$

and

$$L_2 L_1 g = \frac{1}{2} L_2 L_1 g + \frac{1}{2} L_2 f - \frac{1}{2} L_2^2 g.$$

Leads to

$$\begin{aligned} M_1\tau L f &= \frac{1}{4} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \tau L_2 f + \frac{1}{4} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \tau L_1 f \\ &\quad + \frac{1}{4} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \tau L_2 f \\ &\quad + \frac{1}{4} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \tau L_1 f + \tau R(t). \end{aligned} \tag{4.3.20}$$

Where $R(t)$ is given by :

$$\begin{aligned} R(t) &= \frac{1}{4} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(L_2 L_1 - L_2^2 \right) g + \frac{1}{4} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(L_1 L_2 - L_1^2 \right) g \\ &\quad + \frac{1}{4} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(L_2^2 - L_2 L_1 \right) g \\ &\quad + \frac{1}{4} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(L_1^2 - L_1 L_2 \right) g \end{aligned}$$

$R(t)$ further simplifies, when terms in $(L_z^2 - L_i L_j)$ are collected, to

$$R(t) = \frac{\tau L_1}{8} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(L_2^2 - L_2 L_1 \right) g \\ + \frac{\tau L_2}{8} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(L_1^2 - L_1 L_2 \right) g$$

$R(t)$ is an expression whose order is $O(\tau)$. The variation of $f(t)$ over the interval is assumed to be $O(\tau)$ (i.e. f is assumed to be linear for a two-point quadrature). As a result $M_1 \tau L f(t)$ contributes $O(\tau^2)$ to the numerical integration of the forcing, and the omission of $R(\tau)$ affects the solution only in the $O(\tau^3)$ terms.

The solution of (4.3.3a), (4.3.3b) for $s_1 = \frac{1}{4}, s_2 = \frac{3}{4}$ is

$$W_1(z) = 2M_1(z) - \frac{1}{2}M_0(z)$$

and

$$W_2(z) = \frac{3}{2}M_0(z) - 2M_1(z).$$

Equations (4.3.10), (4.3.11), and (4.3.20) may be substituted into

$$\tau T_1 f\left(\frac{3}{4}\right) = 2 \tau M_1 L f\left(\frac{3}{4}\right) - \frac{1}{2} \tau M_0 L f\left(\frac{3}{4}\right) + O(\tau^3)$$

to verify the correctness of the result, (4.3.9), (4.3.11) and (4.3.20) are substituted into

$$\tau T_2 f\left(\frac{1}{4}\right) = \frac{3}{2} \tau M_0 L f\left(\frac{1}{4}\right) - 2 \tau M_1 L f\left(\frac{1}{4}\right) + O(\tau^3).$$

In order to simplify this second expression, observe that the six terms resulting from the substitution of M_0 and M_1 simplify to :

$$\frac{3}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} - \frac{1}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} - \frac{1}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \\ = \frac{1}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I + \frac{\tau}{2} L_1 \right) \left(I - \frac{\tau}{2} L_2 \right)^{-1}$$

and

$$\begin{aligned}
& \frac{3}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} - \frac{1}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} - \frac{1}{2} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \\
& = \frac{1}{2} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I + \frac{\tau}{2} L_2 \right) \left(I - \frac{\tau}{2} L_1 \right)^{-1}.
\end{aligned}$$

When the forcing function is entirely due to Dirichlet boundary conditions, and no boundary terms are “shared” between the two systems of one-dimensional differential equations, the method based on (4.1.2) have local error $O(\tau^3)$ at the boundary.

4.4 Interior Forcing

The boundary forcing method of the previous section has the effect of halving the contributions of the $O(\tau)$ term in the boundary values. In this section the same result for interior forcing is induced. Suppose that there is an interior forcing term $f(t)$ in the differential equation (4.2.3).

Then the algorithm becoms

$$\left(I - \frac{\tau}{2} L_1 \right) U_1 = \left(I + \frac{\tau}{2} L_1 \right) U(t) + \frac{\tau}{2} f_1(t), \quad (4.4.1)$$

$$\left(I - \frac{\tau}{2} L_2 \right) U_2 = \left(I + \frac{\tau}{2} L_2 \right) U(t) + \frac{\tau}{2} f_2(t), \quad (4.4.2)$$

$$\left(I - \frac{\tau}{2} L_2 \right) U_{21} = \left(I + \frac{\tau}{2} L_2 \right) U_1 + \frac{\tau}{2} f_2(t + \tau), \quad (4.4.3)$$

$$\left(I - \frac{\tau}{2} L_1 \right) U_{12} = \left(I + \frac{\tau}{2} L_1 \right) U_2 + \frac{\tau}{2} f_1(t + \tau), \quad (4.4.4)$$

and finally,

$$U(t + \tau) = \frac{1}{2} (U_{12} + U_{21}). \quad (4.4.5)$$

Note that the weights and abscissae, do not relate intuitively to a known quadrature rule. In order to derive (4.4.1)-(4.4.4), we return to the original definition of the one dimensional quadrature

using (4.3.3a), (4.3.3b), and modify $M_1(z)$ by a factor of one half in the two-dimensional solution without modifying $M_0(z)$. Equations (4.4.1) and (4.4.2) are derived by altering the trapezoidal weights W_1 and W_2 by the addition of p_1 and p_2 satisfying

$$p_1 + p_2 = 0 \quad \text{and} \quad s_1 p_1 + s_2 p_2 = -M_1.$$

W_1 and W_2 in (4.4.3) and (4.4.4) are corrected by p_1 and p_2 satisfying

$$p_1 + p_2 = 0 \quad \text{and} \quad s_1 p_1 + s_2 p_2 = M_0 - M_1.$$

The flexibility of the forcing evaluation may now be increased. If (4.3.7) and (4.4.1)-(4.4.4) are solved together, then the interior forcing is incorporated in one dimensional two-point rules and the boundary forcing corresponding midpoint rule.

4.5. Numerical Results

The following test problem is examined using the standard finite difference approximation to one dimensional heat equations $\frac{1}{2}u_t = u_{xx}$ and $\frac{1}{2}u_t = u_{yy}$ on the two-dimensional problem $u_t = u_{xx} + u_{yy}$ on a uniform square mesh. The initial and boundary values are :

$$(t, x, y) \in [0, 1] \times (0, 1)^2,$$

$$u(0, x, y) = \sin \frac{\pi}{2}x \sin \frac{\pi}{2}y,$$

$$u(t, x, 1) = \exp\left(-\frac{\pi^2}{2}t\right) \sin \frac{\pi}{2}x,$$

$$u(t, 1, y) = \exp\left(-\frac{\pi^2}{2}t\right) \sin \frac{\pi}{2}y,$$

$$u(t, x, 0) = u(t, 0, y) = 0$$

The exact solution is

$$u(t, x, y) = \exp\left(-\frac{\pi^2}{2}t\right) \sin \frac{\pi}{2}x \sin \frac{\pi}{2}y.$$

In Table 1 the numerical results for the problem are reproduced using Swayne [18] method, labeled PALOD (Parallel Alternating LOD),

$$R(t) = \frac{\tau L_1}{8} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(L_2^2 - L_2 L_1 \right) g \\ + \frac{\tau L_2}{8} \left(I - \frac{\tau}{2} L_2 \right)^{-1} \left(I - \frac{\tau}{2} L_1 \right)^{-1} \left(L_1^2 - L_1 L_2 \right) g$$

with a locally one-dimensional midpoint rule for boundary forcing compared to a standard LOD, based on (4.1.3), at time $t = 1$. The step size referred to in the tables is that for the total calculation (four individual LOD steps plus averaging).

TABLE 1.
PALOD versus LOD (midpoint quadrature).
Maximum absolute error at $t = 1$.

PALOD (1,1) error				LOD error			
τ	$h = 0.1$	$h = 0.05$	$h = 0.025$	τ	$h = 0.1$	$h = 0.05$	$h = 0.025$
0.2	2.2E-03	8.9E-03	1.7E-02	0.2	1.1E-01	6.3E-01	2.9
0.1	6.6E-05	5.4E-06	2.2E-03	0.1	2.0E-02	2.1E-01	1.2
0.05	1.3E-05	7.5E-06	1.7E-05	0.05	6.6E-03	3.8E-02	4.2E-01
0.02	1.9E-06	2.1E-06	2.3E-06	0.02	3.5E-03	9.4E-03	4.4E-02

APPENDIX

The first twenty four entries of the Pade' Table for $f(z) = e^z$

e^z	$k=0$	$k=1$	$k=2$
$m=0$	1	$1+z$	$1+z+\frac{1}{2}z^2$
$m=1$	$\frac{1}{1-z}$	$\frac{1+\frac{z}{2}}{1-\frac{z}{2}}$	$\frac{1+\frac{2}{3}z+\frac{1}{6}z^2}{1-\frac{1}{3}z}$
$m=2$	$\frac{1}{1-z+\frac{1}{2}z^2}$	$\frac{1+\frac{z}{3}}{1-\frac{2}{3}z+\frac{1}{6}z^2}$	$\frac{1+\frac{z}{2}+\frac{1}{12}z^2}{1-\frac{z}{2}+\frac{1}{12}z^2}$
$m=3$	$\frac{1}{1-z+\frac{1}{2}z^2-\frac{1}{6}z^3}$	$\frac{1+\frac{1}{4}z}{1-\frac{3}{4}z+\frac{1}{4}z^2-\frac{1}{24}z^3}$	$\frac{1+\frac{2}{5}z+\frac{1}{20}z^2}{1-\frac{3}{5}z+\frac{3}{20}z^2-\frac{1}{60}z^3}$
$m=4$	$\frac{1}{1-z+\frac{1}{2}z^2-\frac{1}{6}z^3+\frac{1}{24}z^4}$	$\frac{1+\frac{1}{5}z}{1-\frac{4}{5}z+\frac{3}{10}z^2-\frac{1}{15}z^3+\frac{1}{360}z^4}$	$\frac{1+\frac{1}{3}z+\frac{1}{30}z^2}{1-\frac{2}{3}z+\frac{1}{5}z^2-\frac{1}{30}z^3+\frac{1}{360}z^4}$

e^z	$k=3$	$k=4$
$m=0$	$1+z+\frac{1}{2}z^2+\frac{1}{6}z^3$	$1+z+\frac{1}{2}z^2+\frac{1}{6}z^3+\frac{1}{24}z^4$
$m=1$	$\frac{1+\frac{3}{4}z+\frac{1}{4}z^2+\frac{1}{24}z^3}{1-\frac{1}{4}z}$	$\frac{1+\frac{4}{5}z+\frac{3}{10}z^2+\frac{1}{15}z^3+\frac{1}{120}z^4}{1-\frac{1}{5}z}$
$m=2$	$\frac{1+\frac{3}{5}z+\frac{3}{20}z^2+\frac{1}{60}z^3}{1-\frac{2}{5}z+\frac{1}{20}z^2}$	$\frac{1+\frac{2}{3}z+\frac{1}{5}z^2+\frac{1}{30}z^3+\frac{1}{360}z^4}{1-\frac{1}{3}z+\frac{1}{30}z^2}$
$m=3$	$\frac{1+\frac{1}{2}z+\frac{1}{10}z^2+\frac{1}{120}z^3}{1-\frac{1}{2}z+\frac{1}{10}z^2-\frac{1}{120}z^3}$	$\frac{1+\frac{4}{7}z+\frac{1}{7}z^2+\frac{2}{105}z^3+\frac{1}{840}z^4}{1-\frac{3}{7}z+\frac{1}{7}z^2-\frac{2}{105}z^3+\frac{1}{840}z^4}$
$m=4$	$\frac{1+\frac{3}{7}z+\frac{1}{14}z^2+\frac{1}{210}z^3}{1-\frac{4}{7}z+\frac{1}{7}z^2-\frac{2}{105}z^3+\frac{1}{840}z^4}$	$\frac{1+\frac{1}{2}z+\frac{3}{28}z^2+\frac{1}{84}z^3+\frac{1}{1680}z^4}{1-\frac{1}{2}z+\frac{3}{28}z^2-\frac{1}{84}z^3+\frac{1}{1680}z^4}$

References

- [1]. Crank J. and Nicolson P., *A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type*, MR8,409, Proc. Cambridge Philos. **43** (1947), 50–67.
- [2]. Gallopoulos E. and Saad Y., *On the parallel solution of parabolic equations*, University of Illinois, Urbana-Champaign, CSRD Report **854** (1989).
- [3]. Graves P.R. and Morris J. Ll, *Padé approximants and their applications*, Academic Press, London, 1973.
- [4]. Gourlay A.R. and Morris J.Ll., *The Extrapolation of First Order Methods for Parabolic PDEs II*, SIAM J. Numer. Anal. **17(5)** (1980), 641–655.
- [5]. Gourlay A.R. and Mitchell A.R., *On the structure of alternating direction implicit (A.D.I) and locally one dimensional (L.O.D) difference methods*, J. Inst. Math. Appl. **9** (1972), 80–90.
- [6]. Gumel A.B., *Parallel and sequential algorithms for second order parabolic equation with applications*, PhD thesis, Brunel University, U.K. (1993).
- [7]. Khaliq A.Q.M. , Twizell E.H. and Voss D.A., *On parallel algorithms for semidiscretized parabolic PDEs based on subdiagonal Padé approximation*, Numerical Methods for Partial Differential Equations **9(2)** (1993), 107–116.
- [8]. Lambert J.D., *Numerical Methods for Ordinary Differential System : The Initial-Value Problem*, John Wiley, London-New York, 1991.
- [9]. Lawson J.D. and Morris J.Ll., *The extrapolation of first order methods for parabolic partial differential equation I*, SIAM J, Numer. Anal. **15** (1978), 1212–1244.

- [10].Mitchell A.R. and Griffiths D.F., *The Finite Difference Method in Partial Differential Equations*, John Wiley, London-New York, 1980.
- [11].O'Brien C.G. , Hyman M.A. and Kaplan S., *A study of the numerical solution of partial differential equations*, J. Math. Phys. **29** (1951), 223–251.
- [12].Padé M.H., *Sur la représentation approchée d'une fonction par des fractions rationnelles*, (suppl), Ann. de l'Ecole Normal Supérieure **9** (1892).
- [13].Peaceman D.W. and Rachford H.H., *The numerical solution of parabolic and elliptic differential equations*, J. Soc. Indust. Appl. Math. **3** (1955), 28–41.
- [14].Reusch M.F. , Ratzan L. , Pomphrey N. and Park W., *Diagonal Padé approximations for initial value problems*, SIAM J. Sci. Stat. Comput. **9** (1988), 829–838.
- [15].Saff E.B. and Varga R.S., *On the sharpness of theorem concerning zero-free regions for certain sequences of polynomials*, Numer. Math. **26** (1976), 345–354.
- [16].Shampine L.F., *ODE solvers and the method of lines*, Numerical Methods for Partial Differential Equations **10** (1994), 739–755.
- [17].Smith G.D., *Numerical Solution of Partial Differential Equations*, Clarendon Press, Oxford, 1985 (third edition).
- [18].Swayne D.A., *Time dependent boundary and interior forcing in locally one dimensional schemes*, SIAM J. Sci. Stat. Comput. **8(5)** (1987), 755–767.
- [19].Swayne D.A., *Time dependent Dirichlet boundary conditions and fractional step methods*, Numer. Math. Singapore 1988, Birkhauser Verlag, Basel (1988), 443–455.
- [20].Twizell E.H., *Computational Methods for Partial Differential Equations*, Ellis Horwood, Chichester and John Wiley, New York, 1984.

- [21].Varga R.S., *On high-order stable implicit methods for solving parabolic partial differential equations*, J. Math. Phys. **40** (1961), 220–231.
- [22].Wood W.L. and Lewis R.W., *A Comparison of time marching schemes for the transient heat conduction equation*, Internat. J. Numer. Methods Engeg. **9** (1975), 679-689.
- [23].Yanenko N.N., *The Method of Fractional Steps*, Springer-Verlag, Berlin-New York, 1970.
- [24].Zafarullah A., *Application of the method of lines to parabolic differential equations with error estimates*, J. Assoc. Comput. Mach. **17** (1970), 294–302.
- [25].Zakian V., *Comment on rational approximants to the matrix exponential*, Electronic letters **7** (1971), 261–262.