COMPUTATION OF RATIONAL FUNCTIONS WITH MATRIX ARGUMENT WITH APPLICATION TO INITIAL-VALUE PROBLEMS

by

David A. Swayne

Research Report CS-75-14

Department of Computer Science

University of Waterloo
Waterloo, Ontario, Canada

April 1975

COMPUTATION OF RATIONAL FUNCTIONS WITH MATRIX ARGUMENT WITH APPLICATION TO INITIAL-VALUE PROBLEMS

by

DAVID A. SWAYNE

A Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

at the

UNIVERSITY OF WATERLOO

Waterloo, Ontario

Department of Applied Analysis

and

Computer Science

April 1975

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend it to other institutions or individuals for the purpose of scholarly research.

Signature David A. Swayne

ACKNOWLEDGEMENTS

I am deeply indebted to the people at the University of Waterloo with whom many stimulating discussions have resulted in a true educational experience. Particular mention is due my advisor, Professor J.D. Lawson, as well as Professors J.A. George and R.B. Simpson, the Mathematics Faculty Computing Facility, and the author's fellow graduate students at U.W. As well I appreciate the fortitude of my typist, Monique Krislock, and my wife Dianne and our families for their patience and confidence.

ABSTRACT

This thesis is concerned with the implementation of algorithms designed to numerically solve ordinary initial-value problems whose solutions are asymptotically stable. Those methods considered are themselves A-stable in the sense of Dahlquist, and involve the multiplication of a vector of initial conditions by a rational function of matrix argument. As an approximation to the exponential of z, the rational function is A-acceptable. Our purpose is to develop for a variety of cases a general computational scheme which is nearly optimal with respect to the three considerations of stability, computational complexity, and storage constraint.

The implementation schemes considered use explicitly only the denominator factors of the rational approximation, and, by a trivial extension to complex factors, include possible irreducible quadratic factors associated with complex conjugate poles. The redundancy of the complex elimination procedures is eliminated to provide a simple, stable alternative to forming quadratic factors explicitly. The feasibility of the alternative strategy of using rational approximations whose poles are real is also discussed, and an order-2 example is implemented.

Finally, particular rational approximations with complex conjugate poles are implemented, and the incorporation of forcing terms arising from non-autonomous systems is investigated.

The stability constraint is most important for differential equations which are known as stiff systems, since the relationship

between the stiffness of the differential system and the conditioning of the associated linear algebraic systems is a direct one. The computational scheme is thus oriented towards the evaluation of the displacement of the rational approximation from its asymptotic value.

TABLE OF CONTENTS

Page

1. Introduction

1.1 The Ordering of Computation 1

1.2 Stable Differential Equations 4

1.3 The Exponential Function for Matrix Argument 7

1.4 Approximations to the Exponential Function for Matrix Argument 9

1.5 Systems of Linear Equations 11

2. Previously Known Results

2.1 Numerical Solution to the Initial-Value Problem 16

2.2 Stabilized Methods 19

2.3 Chebyshev Approximation to Exp(z) and Applications 21

2.4 Algorithms for Evaluation of E(tA) 22

3. Optimal Ordering of Computation for Approximations with Real Poles

3.1 General Considerations 25

3.2 Implementation of the Trapezoidal Rule 26

3.3 Stability of ER-Trapezoidal Rule 29

3.4 The Stability of Mixed Operations 30

3.5 Extrapolation Techniques 35

3.6 The L21 Approximation 40

3.7 Applications to Non-Linear Systems 44

3.8 Linear Systems with Implicit Matrix Factors 46

4. Irreducible Quadratic Factors

4.1 Preliminary Considerations 55

4.2 Operation Times and Programming Overhead ... 55

4.3 The Computational Cost of Irreducible Quadratic Factors 58

4.4 The Existence of a Solution to the Complex Problem 61

4.5 The Sensitivity of Complex Elimination 64

4.6 The Cost of Complex Elimination 67

4.7 Round-off Errors in the Matrix Computations 71

4.8 Linear Systems with Implicit Factors (II) . 76

4.9 Conclusions 77

5. Specific Implementations

5.1 Details of Algorithm Construction 79

5.2 The Padé (2,0) Approximation 80

5.3 The Padé (2,1) Approximation 83

5.4 The Padé (2,2) Approximation 86

6. Conclusions and Numerical Results

6.1 The Decomposition of Rational Functions ... 91

6.2 Linear Systems with Implicit Matrix Factors (III) 92

6.3 Repeated Exponential Approximations 93

6.4 Conclusions and Numerical Results 99

Bibliography

CHAPTER 1

INTRODUCTION

1.1 The Ordering of Computation

The purpose of this thesis is to examine the structure of algorithms for the numerical solution of the initial value problem for systems which involve the computation of a rational approximation to the exponential function for matrix argument. The three main concerns are numerical accuracy and stability of the implementation, the relative computational speed, and the storage requirements as a function of the dimension and structure of the differential system.

Tradeoffs may exist among these considerations. Consider the familiar trapezoidal rule applied to the scalar differential equation

$$y' = qy,\ y(0) = y_0,\quad q < 0. \tag{1.1.1}$$

To obtain the solution at t = h, we first compute

$$y_{h/2} = (1 + hq/2)y_0$$

then

$$y_h = (1 - hq/2)^{-1} y_{h/2} \quad . \tag{1.1.2}$$

However, if the order of the computation is reversed, and we note that the first step in equation 1.1.2 need not be executed explicitly, we obtain the alternate implementation:

$$\begin{aligned} y^*_{h/2} &= (1 - hq/2)^{-1} y_0 \quad , \\ y_h &= 2y^*_{h/2} - y_0 \quad . \end{aligned} \tag{1.1.3}$$

The second mode of computation requires the temporary storage of y_0, while this is not a necessary feature of the first. As noted the first computation in equation 1.1.2 has been eliminated.

In circumstances where an algorithm with fixed step-size is applied to a constant-coefficient linear system, the coefficient matrix in the system:

$$y' = Ay \qquad 1.1.4$$

may be overwritten with a suitable decomposition of the matrix factor $(I - hA/2)$. In any program where step-size changes are possible this version might require increased storage, but since the initial value must be kept, and matrix-vector multiplication involves temporary storage, the two procedures have approximately the same storage requirements.

Stability considerations are particularly important in the numerical treatment of so-called stiff systems of differential equations. Systems of the form of equation 1.1.4 are called stiff when the coefficient matrix A has eigenvalues of widely different magnitudes, resulting in solution components whose time-behavior differs greatly. For our purposes, we suppose that the eigenvalues of A have negative real parts which differ in size. The effect of this situation is best illustrated by the familiar step-size constraint on the forward Euler's method for systems:

$$y_{n+1} = (I + hA)y_n \qquad 1.1.5$$

namely, $\frac{||hA||}{2} < 1.$

This is necessary to ensure that the solution components associated with the large eigenvalues do not grow in size indefinitely, when in fact they should tend to zero with increasing time. These components are the components which tend to zero the most quickly in the actual solution, and any successful approximation procedure has to mimic this behavior.

The implicit Euler's method, on the other hand is stable for all such systems, regardless of step-size.

$$(I - hA)y_{n+1} = y_n \qquad 1.1.6$$

In the formulation of the trapezoidal rule given by 1.1.2 the instability inherent in equation 1.1.5 is a source of numerical error. Even though the trapezoidal rule may be inaccurate for stiff problems because of its incorrect asymptotic form, this is a function of the vector y_0, and if the ordering of equation 1.1.3 is used, this effect can be minimized for certain initial values associated with small eigenvalues of A. These transient solution components, even if only present at round-off level, contaminate the first step of 1.1.2, and the second step must damp this. Noise in the direction of the slowly decaying solution persists as the solution advances in time.

For rational approximations of higher degree, it is proposed to obtain a solution by partial fraction decomposition of the rational function. The storage requirement is not significantly increased for homogeneous problems, and for forced linear systems where the forcing function is evaluated at a number of points proportional to the degree of the denominator of the rational approximation, it remains roughly the same. The primary saving is in speed of computation, since matrix-

vector operations associated with the numerator of the rational function are eliminated. Numerical stability is improved as well, since these matrices are of large norm for stiff problems.

1.2 Stable Differential Equations

The initial-value problem for a first-order nxn system of differential equations has the form

$$y' = f(t,y),\ y(0) = y_0, \qquad 1.2.1$$

where f is assumed to be such that the initial condition y_0 ensures a unique solution on an interval $0 \leq t \leq T$. The following definitions serve to define stable systems. For a more comprehensive treatment of this topic, the reader is referred to [11].

Definition 1.1 If $f(t,a) = 0$ for all $t \geq 0$ then $y(t) = a$ is a point solution of equation 1.2.1. We can assume, without loss in generality, that $a = 0$. Then $y(t) = 0$ is called the null solution of 1.2.1.

Definition 1.2 The null solution of equation 1.2.1 is stable if, given $\varepsilon > 0$, there exists a $\delta > 0$ such that for $0 < ||y_0|| < \delta$, $||y(t)|| < \varepsilon$ for all $t > 0$. If $||y(t)|| \to 0$ as $t \to \infty$, the null solution of 1.2.1 is said to asymptotically stable.

The notion of stability in the neighborhood of a point solution includes stability in the neighborhood of a known function $a = a(t)$, and for any particular system, the second case is reducible to the first.

Definition 1.3 Equation 1.2.1 is said to autonomous if f is a function

of y only, and nonautonomous otherwise.

Definition 1.4 Equation 1.2.1 is said to be linear if f can be written in the form

$$y' = A(t)y + r(t) \ , \qquad 1.2.2$$

where A(t) is an nxn square matrix of finite norm, and r(t) is a vector of forcing terms independent of the solution y.

If A is a constant matrix, we say that 1.2.2 is a linear constant coefficient system. (To avoid confusion, we will assume when referring to a linear system that, unless otherwise indicated, the system is a constant coefficient system.) A linear system for which $r(t) \equiv 0$ is called a homogeneous linear system. For any such first order linear system, there are n linearly independent solutions. The matrix differential equation associated with a homogenous linear system,

$$Y'(t) = A(t)Y(t) \ , \qquad 1.2.3$$

where Y(t) has non-vanishing determinant defines a fundamental matrix for the homogeneous system. If $Y(0) \equiv I$, the identity matrix, Y(t) is the principal matrix for the system. For constant coefficient systems, the principal matrix solution is defined by

$$Y(t) = \exp(tA), \text{ where}$$
$$\exp(tA) = I + tA + \cdots + \frac{t^n A^n}{n!} + \cdots \qquad 1.2.4$$

The solution to the initial value problem for linear systems can be written in terms of the principal matrix solution of the system. In particular, for constant coefficient systems, the solution in terms

of the principal matrix system takes the form

$$y(t) = \exp(tA)y_0 + \int_0^t \exp[(t-s)A]r(s)ds. \qquad 1.2.5$$

Terms in equation 1.2.5 which are of the form $\exp(tA)v$, v a vector are terms in the characteristic solution or complementary function. That function which satisfies the inhomogeneous problem for arbitrary initial conditions is known as a particular solution or particular integral.

Finally, following [11], we introduce the nonlinearity condition and the definition of asymptotic stability for general systems in terms of linear systems.

Theorem 1.1 Let

$$y' = Ay + r(t,y) \quad , \qquad 1.2.6$$

where A is a real constant matrix whose eigenvalues all have negative real parts. Let $r(t,y)$ be sufficiently well-behaved for the existence of a solution for $||y||$ small and $t \geq 0$. Suppose further that r satisfies the nonlinearity condition,

$$\lim_{||y||\to 0} \frac{||r(t,y)||}{||y||} = 0,$$

uniformly in $t \geq 0$. Then the null solution of equation 1.2.6 is asymptotically stable.

Proof See [11,p.314]. □

The study of constant coefficient systems leads to a solution procedure for systems of the more general form 1.2.1. When the

eigenvalues of the Jacobian matrix of 1.2.1 have negative real parts,

i.e., $\quad \text{Re } \lambda\left[\frac{\partial f_i}{\partial y_j}(t)\right] < 0.$

for t in a neighborhood of t_0, the matrix A can be chosen to be an approximation to the Jacobian evaluated at (t_0, y_0), the solution can be advanced stably to $t = t_1$ with a polynomial approximation to the nonlinearity term r(t,y), and the process repeated.

For constant coefficient systems with forcing, we attempt to implement an approach which exactly integrates the particular integral when r(t) is a polynomial. The error is solely in the evaluation of the terms involving the complementary function, and the method is equivalent to the replacement of exp(tA) by an approximation E(tA) after the integral term in equation 1.2.5 has been evaluated for polynomial r(t). The methods in this thesis may be extended to integrate systems of the form of equation 1.2.6 in this fashion.

1.3 The Exponential Function for Matrix Argument

The exponential function for matrix argument, defined by equation 1.2.4 is the principal matrix solution to equation 1.2.2 for constant matrix A. Some of its properties are discussed briefly.

Two nxn matrices A and J are said to be <u>similar</u> if there exists an nxn nonsingular matrix H such that

$$J = HAH^{-1} \qquad 1.3.1$$

The <u>Jordan canonical form</u> of a matrix A is a matrix J such that

$$J = \begin{pmatrix} J_1 & & 0 \\ & J_2 & \\ & \ddots & \\ 0 & & J_p \end{pmatrix}, \qquad 1.3.2$$

which is similar to A, where

$$J_i = \begin{pmatrix} \lambda_i 1 & \cdots & 0 \\ \ddots & \ddots & \\ & \lambda_i 1 & 0 \\ 0 & \cdots & \lambda_i \end{pmatrix} . \qquad 1.3.3$$

Thus each Jordan block of A is associated with one eigenvalue of A. More than one Jordan block may be associated with a particular eigenvalue. For example, all positive definite real symmetric matrices have 1x1 Jordan blocks, regardless of the multiplicity of the eigenvalues.

Definition 1.5 A matrix whose Jordan canonical form has all Jordan blocks of dimension 1x1 is called a matrix of simple structure.

Any matrix of simple structure has a complete set of eigenvectors. When the canonical form has blocks of dimension greater than one, there is only one eigenvector corresponding to the block. Such a system is termed defective.

The exponential function of a matrix is similar to the exponential of its Jordan form. For scalar t,

$$\exp(tJ) = H \exp(tA) H^{-1} , \qquad 1.3.4$$

and for a kxk Jordan block,

$$\exp(tJ_i) = \exp(t\lambda_i) \begin{pmatrix} 1 & t & & \frac{t^{k-1}}{(k-1)!} \\ & \ddots & & \\ & & 1 & t \\ 0 & & & 1 \end{pmatrix} . \qquad 1.3.5$$

In general any function f(A) defined on the spectrum of a matrix A is similar to f(J). The Jordan blocks J_i of A also satisfy the equation

$$f(tJ_i) = \begin{pmatrix} f(t\lambda_i) & \cdots & f^{k-1}(t\lambda_i)\, t^{k-1}/(k-1)! \\ 0 & \ddots & \ddots \\ 0 & 0 & f(t\lambda_i) \end{pmatrix} . \qquad 1.3.6$$

Theorem 1.2 If A is an nxn matrix whose eigenvalues have negative real parts

$$\lim_{t\to\infty} \exp(tA) = 0_{nxn} \quad .$$

If A has simple structure, then

$$||\exp(tA)|| < 1 \quad ,$$

in some supremum norm.

Proof

The result follows from the similarity of exp(A) to its Jordan form, and the fact that $\exp(\lambda t)p(t)$ goes to 0 for any polynomial p, as $t \to \infty$, for $\mathrm{Re}[\lambda] < 0$.

For the second part of the Theorem, we know that $|\exp(\lambda t)|<1$ for $\mathrm{Re}[\lambda] < 0$, and that for any $\varepsilon>0$ there exists a sup. norm such that the spectral radius of A and its norm differ by less than ε. □

The qualification that A have simple structure is necessary to avoid the possibility of the growth of off-diagonal terms in Jordan blocks of dimension greater than 1x1.

1.4 Approximations to the Exponential Function for Matrix Argument

It follows from equation 1.3.6 that any function of matrix argument is defined by the scalar values of the function on the spectrum of the matrix argument [18]. Thus, when investigating properties of a matrix-valued function, we may consider its properties for scalar argument z, which is assumed to lie in a domain containing the spectrum of the matrix.

For rational approximations to the exponential, denoted for scalar argument z by E(z) the following properties are of importance.

$$P_1: \text{ For } Re[z] < 0 \quad |E(z)| < 1. \qquad 1.4.1$$

$$P_2: \lim_{Re[z] \to -\infty} E(z) = 0.$$

Definition 1.5 A rational approximation to exp(z) with order k is a ratio of two polynomials of degree m and n, respectively, $0 \leq k \leq m + n$, such that for $z \to 0$

$$R_{m,n}(z) = P_n(z)/Q_m(z), \; Q_m(z) \neq 0, \qquad 1.4.2$$

$$P_n(z) - Q_m(z)e^z = O(z^{k+1})$$

Definition 1.6 If a rational approximation to exp(z) satisfies P_1, it is said to be A-acceptable.

Definition 1.7 If, in addition the approximation satisfies P_2, it is said to be L-acceptable.

Definition 1.8 A uniform rational approximation to the exponential function is a rational approximation on a domain D, such that for $\varepsilon > 0$,

$$\sup_{z \in D} |E(z) - \exp(z)| < \varepsilon . \qquad 1.4.3$$

For D a segment of the real line, a best uniform approximation to exp(z) is an approximation with minimum-maximum error [12]. Such an approximation is called a Chebyshev approximation to exp(z).

Definition 1.9 An algorithm for the numerical solution to the initial value problem is said to be A-stable if, when applied to $y' = qy$, $y_0 = 1$, $Re[q] < 0$, it defines an approximation $z(ih)$, $i = 1,2,3,\ldots$, such that

$|z(t)| \leq 1$ for all $t \geq 0$. The function $z(t)$ may be termed A-acceptable, considered as an approximation to $\exp(z)$.

Definition 1.10 If in addition the $z(t)$ is L-acceptable, the algorithm is said to be strongly A-stable.

The implicit Euler's method is an example of a method which is strongly A-stable, the trapezoidal rule is only A-stable, while Euler's method is neither. For stiff systems we will be considering the advantage of strongly A-stable methods.

1.5 Systems of Linear Equations

The implementation of the schemes that we are examining for the solution to the initial value problem involve repeated application of algorithms to solve linear systems of the form

$$Ax = b \quad , \tag{1.5.1}$$

where A is an nxn real non-singular matrix and b is a known nx1 column vector. The theoretical background is covered in Wilkinson [40] and Forsythe and Moler [17].

The usual definitions for matrix and vector norms will be assumed, particularly the Euclidean norm,

$$||x||_2 = (x^T x)^{\frac{1}{2}} \quad . \tag{1.5.2}$$

Given a vector norm, the subordinate or natural matrix norm is defined by the relation

$$||A|| = \sup_{||x||=1} ||Ax||.$$

A compatible matrix norm satisfies the weaker condition:

$$||Ax|| \leq ||A|| \; ||x||.$$

The spectral radius of a matrix is defined by

$$\rho(A) = \sup_i |\lambda_i| \ ,$$

λ_i an eigenvalue of A.

The singular values $\sigma_i, i = 1,\ldots,n$ of an nxn matrix A are defined by the relation

$$\sigma_i^2 = \lambda_i(A^*A) \ ,$$

where A^* denotes the conjugate transpose of the matrix A.

The Euclidean norm of a matrix A is defined by

$$||A||_2 = [\rho(A^*A)]^{\frac{1}{2}} \ .$$

For positive definite real symmetric matrices the Euclidean matrix norm is the same as the spectral radius. For arbitrary A, the Euclidean norm can be arbitrarily greater than the spectral radius, however. If for example,

$$A = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix},$$

$$\rho(A) = 1,$$

but

$$||A||_2 = O(\alpha^2).$$

The principal method we shall use in the solution of equations 1.5.1 arising from the denominator factors of rational approximations is Gaussian elimination. The coefficient matrix is factored into a lower triangular factor L and an upper triangular factor U. The solution for a particular right-hand side b is found by performing

$$Ly = b \ , \qquad 1.5.4$$

$$Ux = y \ .$$

A complete discussion of Gaussian elimination can be found, for

example in [17], [40]. Usually we will consider procedures which maintain stability through partial pivoting, or, for symmetric matrices, diagonal pivoting strategies.

The definition of the L and U factors varies with the form of Gaussian elimination procedure used. The essential features are the same, except for symmetric variants such as Cholesky factorization, and we will use one type, namely Gauss-Doolittle, to describe the algorithm.

Let

$$A = A_o = \begin{pmatrix} a_{11} & r_1^T \\ c_1 & M_o \end{pmatrix} .$$

Then for $u_{ii} = a_{ii}$

$$A_o = \begin{pmatrix} 1 & 0 \\ \frac{c_1}{u_{11}} & I_{n-1xn-1} \end{pmatrix} \begin{pmatrix} u_{11} & r_1^T \\ 0 & M_o - \frac{c_1 r_1^T}{u_{11}} \end{pmatrix} , \quad 1.5.5$$

and the successive decomposition steps may be stored in place in the matrix A. The factorization is carried out recursively on

$$A_i = M_{i-1} - \frac{c_i r_i^T}{u_{ii}} , \; u_{ii} \neq 0. \quad 1.5.6$$

The computational cost in terms of multiplications for the decomposition phase of Gaussian elimination for an arbitrary nxn matrix is $n^3/3 + O(n^2)$. The cost of substitution for a particular right-hand side is $n^2 + O(n)$, i.e. it is proportional to the cost for one matrix-vector multiplication.

There has been considerable work done to minimize the cost of Gaussian elimination for systems which have large dimension but have

many zero elements. We will be implicitly referring to the preservation of such sparsity in some later discussions in the thesis.

The following theorem analyzes the possible effect of a change in the right-hand side on the solution to 1.5.1.

Theorem 1.3 Let $Ax = b$ be a nonsingular nxn system of linear equations and let δb be a change in the right-hand side b. Then, if we denote by $||\delta x||$ the corresponding change in x,

$$\frac{||\delta x||}{||x||} \leq K(A) \frac{||\delta b||}{||b||}$$

where

$$K(A) = ||A||\,||A^{-1}|| \qquad 1.5.7$$

is called the condition number of A.

Proof See [39]. □

Note that this result is sharp, in the sense that equality is possible for arbitrary A and $||\delta b||$.

The class of matrices known as band matrices arise frequently in the semi-discretization of partial differential equations. They involve significantly less computational effort than general matrices when solving 1.5.1.

Definition 1.12 The semi-band-width of a square matrix A is defined as:

$$m = \sup_{a_{ij} \neq 0} |i - j|$$

The band width of a matrix is then $2m + 1$.

The solution of linear systems in t-digit floating point arithmetic places some limitation on the type of linear systems that can be solved, and affects the significance of the solution obtained. Defining the error by the technique of backward error analysis, we need the following two theorems.

Theorem 1.4 If x and y are any two t-digit base B floating point numbers, and * is one of +,-,.,/ then the floating-point result of x*y, fl(x*y), is given by

$$fl(x*y) = x*y(1+\delta),$$

where $|\delta| \leq u$, the unit round-off error, $0.5B^{1-t}$, or B^{1-t}, depending on whether the arithmetic is performed by rounding or chopping.

Proof See [17,p. 90]. □

Finally,

Theorem 1.5 If

$$x = (x_1 \ldots\ldots x_n)^T$$

$$y = (y_1 \ldots\ldots y_n)^T$$

are two t-digit floating point arrays,

$$|x^T y| = \sum_{i=1}^{n} x_i y_i (1 + (n + 2 - i)1.01\theta_i u) \qquad |\theta_i| < 1 .$$

Proof See [17,p. 93]. □

Remark: obviously, if $|x^T.y| << ||x||.||y||$, the floating point answer may have severe relative error.

Theorems 1.4 and 1.5 will be used in determining the stability of both the linear systems and the formation of quadratic polynomials of matrix argument.

CHAPTER 2

PREVIOUSLY KNOWN RESULTS

2.1 Numerical Solution of the Initial-Value Problem

Classically, there have been two main types of methods for solving the initial-value problem. They can be categorized as multistep methods and single step (Runge-Kutta, Taylor, and Hermite type) methods.

The general linear multistep method for the solution to the initial value problem at $t = (m + 1)h$ is expressed by the formula

$$y_{m+1} = \sum_{i=0}^{k} a_i y_{m-i} + h\sum_{i=1}^{k} b_i y'_{m-i}, \quad m = k,k+1,.... \qquad 2.1.1$$

Dahlquist [14] defines the concept of A-stability in terms of the solution to equation 1.1.1 for complex q with negative real part. A-stability relates to the fact that, as $m \to \infty$, the solution to 1.1.1 by an A-stable method tends to zero. He proved the following theorem for linear multistep methods.

Theorem 2.1 An explicit multistep method cannot be A-stable. The order of an A-stable multistep method cannot exceed two, and the trapezoidal rule has the smallest error coefficient of all A-stable linear multistep methods.

A method which is not A-stable may be used for systems provided suitable restrictions are placed on the time-step.

Definition 2.1 The region of stability of a method is the region S in the left-half complex plane within which hq must lie in order that

solutions to equation 1.1.1 by formula 2.1.1 tend to zero for large m.

Gear [19] has considered multistep methods for which only a small region of the left-half plane is missing from the region of stability. Methods such as Gear's, which are stable for most of the left-half complex plane are known as stiffly stable methods. The maximum order of stiffly stable methods is conjectured to be eleven [15].

The weaker condition, that y(nh) converge to the solution y(t) as $h \to 0$, $nh = t$, imposes less stringent conditions on equation 2.1.1. For a multistep method to be stable in this sense (Dahlquist [13]), the order cannot exceed k + 1 for k odd and k + 2 for k even. Modified multistep methods, which contain an additional term, $\beta h y'_{m-\theta}$, $\theta < 1$, [5] are stable in this weaker sense [13] for order 2k formulae. The maximum attainable order for modified multistep methods seems to be k = 7 or 8.

One class of single-step methods is based on Hermite interpolation at t = mh and t = (m + 1)h. The formula is given by

$$y_{m+1} = y_m + \sum_{i=1}^{k} C_{ik}^{m} y_m^{(i)} + \sum_{i=1}^{k} C_{ik}^{m+1} y_{m+1}^{(i)} \qquad 2.1.2$$

The coefficients of the maximal order 2k formulae are the polynomial coefficients of the numerator and denominator of the diagonal Padé approximation normalized to 1 for the constant term. One analysis and implementation of these formulae may be found in [34]. Since they reduce for y' = qy to the Padé diagonal approximation, they are A-stable [16].

Liniger and Willoughby [33] consider intermediate methods based on Hermite interpolation, in the sense that, for the scalar linear

problem, they reduce to rational approximations which lie between the diagonal Padé and the fully implicit Taylor expansion of the solution to the differential equation. The parameters in the interpolation formulae are chosen to fit the exponential function for a particular argument which at which the problem indicates that such a fit would be useful to stabilize the approximation for that argument range. For k = 1, the intermediacy condition reduces to the familiar stability condition for the weighted trapezoidal approximation

$$y_{m+1} = y_m + h\{\mu y'_m + (1 - \mu)y'_{m+1}\},$$

namely, $\mu \leq 1/2$.

The Runge-Kutta formulae are another class of single-step methods defined by the equations

$$y_{m+1} = y_m + h\sum_{i=1}^{\nu} w_i k_i,$$

where

$$k_i = f(t_m + c_i h,\ y_m + h\sum_{j=1}^{\nu} b_{ij}k_j), \qquad 2.1.3$$

and

$$c_i = \sum_{j=1}^{\nu} b_{ij}.$$

A Runge-Kutta process is explicit if $b_{ij} = 0$ for $j \geq i$, and implicit otherwise. Butcher [2] established that the highest attainable order for an explicit ν-stage process is

$$P(\nu) = \nu,\ \nu = 1,4,$$
$$\nu - 1,\ \nu = 5,6,7$$
$$\nu - 2,\ \nu = 8,9$$
$$P(\nu) \leq \nu - 2,\ \nu \geq 10$$

For implicit processes Butcher [3] established that the highest attainable order is 2ν. Butcher [4], Ehle [16], and Chipman [10] have investigated the implicit Runge-Kutta processes based on Gaussian, Radau, and Lobatto quadrature formulae, and identified them with diagonal, subdiagonal, and second subdiagonal Padé approximations, respectively. Gear [20] has shown that, given any rational approximation to $\exp(z)$, $E(z) = P(z)/Q(z)$, with distinct non-zero poles and with degree $P(z) \leq$ degree $Q(z) = n$, there exists an n-stage Runge-Kutta process which, when applied to $y' = qy$, results in the difference equation $y_{m+1} = E(qh)y_m$. However the order of the resulting process is usually much lower than that of the rational approximation.

2.2 Stabilized Methods. Lawson [29] has investigated the following stabilization transformation which when applied to existing methods has resulted in new A-stable methods. For $y' = f(t,y)$, he introduces the change of variable $z(t) = \exp(-tA)y(t)$. From consideration of the initial-value problem for $z(t)$, he obtains the formula

$$y(t) = \exp(tA)y(0) + \int_0^t \exp[(t-s)A]\{f(s,y(s)) - Ay(s)\}ds. \quad 2.2.1$$

Alternatively,

for $u(t) = f(t,y(t)) - Ay(t)$,

$$y(t + h) = \exp(hA)y(t) + h\int_0^1 \exp[(1 - \tau)hA]\cdot u(t + \tau h)d\tau. \quad 2.2.2$$

The variable change results in a reduction of the Lipschitz constant associated with the problem. It is now proportional to $||\partial \underline{f}/\partial \underline{y} - A||$.

A need not be the Jacobian. It is sufficient that the Jacobian matrix of the transformed problem, which is similar to $\partial f/\partial y - A$, have eigenvalues with negative real part. For constant coefficient systems, the transformed problem is just equation 1.2.5. The method for the original initial-value problem is then A-stable provided an A-acceptable approximation is used to implement the transformation.

Setting $t \equiv 0$ in equation 2.2.2, we can derive a recursion formula for the integral term.

Define

$$M_j(hA) \equiv \int_0^1 \exp\{(1-\tau)hA\}\tau^j d\tau.$$

Then $hA\{M_j(hA)\}$ may be integrated by parts to obtain

$$hA\{M_j(hA)\} = -\exp(\tau hA)\tau^j\Big|_0^1 + j\int_0^1 \exp\{(1-\tau)hA\}\tau^{j-1}d\tau.$$

It is an easy matter to verify that hA is a factor of the right hand side of this expression. In view of the remarks of section 1.3 we may consider instead the expressions as functions of a complex variable z. Accordingly, Lawson [29] derives a set of moment functions based on integration by parts of the particular integral for

$r(t) = t^m b$, $m = 0,1,\ldots,k-1$, where k is the order of the rational approximation E(z), of the form

$$M_0(z) = z^{-1}[E(z) - 1]$$

$$M_j(z) = z^{-1}[jM_{j-1}(z) - 1] \quad j = 1,2,\ldots,k-1 \qquad 2.2.3$$

Using these moments, a set of weight functions are produced for particular abscissae to yield for the inhomogeneous problem a

solution which is formally the same as that of the continuous case, with exp(z) replaced by E(z). The following two theorems establish the behavior of this construct [29].

Theorem 2.2 Let $E(z) - \exp(z) = O(z^{k+1})$, $z \to 0$ and let $\{W_i(hA)\}$ be computed from 2.2.3. Then the algorithm defined by

$$y_{m+1} = E(hA)y_m + h\sum_{i=1}^{k} W_i(hA)\cdot u(t_m + \alpha_i h) \qquad 2.3.1$$

is exact for particular integral of $y' = Ay + p(t)$, where $p(t)$ is an arbitrary vector polynomial of degree $k - 1$ or less and A is a real non-singular square matrix.

Theorem 2.3 Let the scheme of 2.3.1 be A-stable and exact for the particular integral of $y' = Ay + p(t)$, where A is a square matrix with eigenvalues in the left-half plane. Then $\lim_{n\to\infty}[y_n - y(t_n)] = 0$, regardless of stepsize h.

If $E(z) - \exp(z) = O(z^{k+1})$ holds, the particular integral is exactly integrated for polynomially forced problems of polynomial degree $\leq k - 1$. The error in the solution is then completely in the terms associated with the evaluation of the fundamental matrix in the complementary function.

One important aspect of this method of solution is that all the weights so produced have the same denominator factors. This is an advantage that this approach has over other methods such as [32].

Generalized multistep methods using this quadrature scheme appear in [27]. The υ-stage implicit Runge-Kutta processes mentioned in [4], [10], [16] exactly simulate the polynomial solutions of polynomially forced problems to degree $\leq \upsilon$, as was pointed out in [26].

2.3 Chebyshev Approximations to Exp(z) and Applications

One approach to the numerical solution of partial differential equations of heat-conduction type has been the semi-discretization in the spatial coordinates to produce, for linear problems, a constant-coefficient ordinary differential system

$$y' = Ay + f(t), \qquad 2.3.1$$

where A is a constant square matrix.

For such problems, Cody, Meinardus, and Varga have developed algorithms based on Chebyshev rational approximations to exp(-z) on $[0,\infty)$ which have been successfully used for constant r(t) to obtain a solution y(T) for the discretized problem in one time-step for any T. [12]

Lawson [30] has extended the usefulness of such methods by developing combined order-uniformity constrained Chebyshev approximations which integrate such problems for polynomial forcing.

2.4 Algorithms for Evaluation of E(tA)

For the implementation of generalized Runge-Kutta processes, a suggested method of evaluating exp(z) for matrix argument which would avoid the instability associated with a summation of a large number of terms in the Maclaurin expansion is the computation of a low-order expansion, which is valid for argument with small norm. The use of the series:

$$E(z) = 1 + \sum_{n=1}^{5} \frac{z^n}{n!} + 0.5625\,\frac{z^6}{6!} \qquad 2.4.1$$

which has a larger region of stability is recommended in [28]. This truncated expansion is used to provide the initial matrix argument

$\exp(2^{-m}tA)$ from which the approximation $\exp(tA)$ is calculated by m matrix-squaring operations:

$$\exp(2^{-k+1}hA) = E(2^{-k}hA)^2. \qquad 2.4.2$$

For a polynomial expansion of degree k, and a full matrix with dimension nxn, the cost of evaluating $E(tA)$ is $O((k + m)n^3)$. The same approximation, applied to the vector b as a sequence of matrix vector operations costs $kn^3 + 2^m n^2$ multiplicative operations. Hence, the choice of method depends on the larger of $\{mn, 2^m\}$.

If the matrix is sparse the exponential function or any approximation based on powers of the matrix greater than 1 fills in non-zero elements quickly, and the matrix-vector approach with

$$E(z) = 1 + z \qquad 2.4.3$$

preserves sparsity in the individual matrix-vector operations. For linear heat-conduction problems involving initial conditions which guarantee a rapidly-decaying solution, the standard Euler's method has been shown to be fairly efficient [7]. This is the simplest case of an argument-reduction scheme such as we have been discussing.

Another problem associated with such argument-reduction schemes is the inability of the norm-reduced exponential approximation to "see" the effect of components associated with small eigenvalues of the coefficient matrix. These methods are less well suited to following slowly-varying solutions over more extended time-periods.

If a method based on a rational approximation to $\exp(tA)$ is chosen to follow a solution component which persists over a large time interval, then the rational approximation must satisfy either Definition 1.5 or Definition 1.8 to ensure that the limitation on the time-step for the

initial-value problem is chosen from truncation error considerations rather than consideration of the stability of the rational function evaluation procedure.

The A-stable formulae of Section 2.1 all contain an implicit evaluation of an A-acceptable approximation $E(z)$. The generalized methods of Section 2.2 and the one-step global methods for forced linear systems of Section 2.3 have an explicit evaluation of a term $E(tA)$ in the solution procedure. In [8], Cavendish, Culham and Varga have implemented for some Chebyshev rational approximations the evaluation of $E(M^{-1}N)$, where the argument $M^{-1}N$ is calculated à priori. The method applied to such an argument necessarily implies that the matrix M be of particularly simple structure (in their example it is diagonal) to avoid matrix fill if the problem as posed has a sparse coefficient matrix.

CHAPTER 3

OPTIMAL ORDERING OF COMPUTATION FOR APPROXIMATIONS WITH REAL POLES

3.1 General Considerations

In the introduction, the effect of a change of computational ordering on the number of operations and the stability of the numerical solution to a linear initial-value system by the trapezoidal rule was discussed. In this chapter, these considerations are covered in more detail. The more general case of linear factors is examined with the trapezoidal rule as a model. As well, the development of a second-order method which is strongly A-stable, involving real poles is investigated, and one such method is examined in detail. This method is used as a model for implementation procedures which are to be followed for more complex methods based on rational approximations to exp(z).

The quadrature weights for a given rational approximation to exp(z) will in general have a non-constant numerator. The method of Section 3.6 avoids the problem of computing the numerators of the weight functions explicitly by a partial fraction factorization of the weights and the rational approximation itself, with the added simplification of the separation of the partial fraction factorization for the quadrature points chosen in the section.

Section 3.8 considers another problem, namely the treatment of linear systems of the form $My' = Ny + f(t)$ and demonstrates that the partial fraction factorization of the approximation and the weight functions permits efficient calculation of numerical solutions to this problem for quite general conditions on M, provided the denominator

factors are linear.

The treatment of mismatched initial-boundary values for the heat equation is interpreted as a case of implicit matrix argument, and the necessity of L-acceptability for this case is demonstrated.

3.2 Implementation of the Trapezoidal Rule

The trapezoidal rule, known also as the Crank-Nicholson formula in the study of parabolic partial differential equations, is based on the Padé (1,1) rational approximation to exp(z).

It is A-stable, but the rational approximation on which it is based is not L-acceptable, since

$$\lim_{\mathrm{Re}[z]\to-\infty} P_{11}(z) = -1 \ . \qquad 3.2.1$$

The rational approximation and its first two moments from equations 2.2.3 are:

$$P_{11}(z) = (1 - z/2)^{-1}(1 + z/2), \qquad 3.2.2$$

$$M_0(z) = (1 - z/2)^{-1} \ , \qquad 3.2.3a$$

$$M_1(z) = \tfrac{1}{2}(1 - z/2)^{-1}. \qquad 3.2.3b$$

For equation 2.3.1, the trapezoidal rule for quadrature points α_0 and α_1 is

$$(I - hA/2)y_{n+1} = (I + hA/2)y_n + h\{\frac{\frac{1}{2} - \alpha_1}{\alpha_0 - \alpha_1} f(t_n + h\alpha_0) +$$

$$+ \frac{\frac{1}{2} - \alpha_0}{\alpha_1 - \alpha_0} f(t_n + h\alpha_1)\} \ . \qquad 3.2.4$$

For $\alpha_1 = 1 - \alpha_0$, this simplifies to

$$(I - hA/2)\ y_{n+1} = (I + hA/2)y_n + \frac{h}{2}\{f(t_n + \alpha_0 h) + f(t_n + \alpha_1 h)\}. \qquad 3.2.4a$$

The usual quadrature points chosen on the interval [0,1] are {0,1} or {1/2}, the second abscissa being unnecessary for the midpoint rule for the Padé (1,1) approximation. For simplicity, we will refer to the inhomogeneous term as $f_{1/2}$. The computational ordering of equation 1.1.2 we will call TR, and the ordering implied by 1.1.3 will be given the designation ER. The algorithm 3.2.4 written in ER form is:

$$y^*_{n+1/2} = (I - hA/2)^{-1}\{y_n + hf_{1/2}/2\}$$

$$y_{n+1} = 2y^*_{n+1/2} - y_n \qquad 3.2.5$$

A third ordering, which has the same characteristics as TR, is given by the implicit Runge-Kutta model (IRK):

$$k_1 = (I - hA/2)^{-1}\{Ay_n + f_{1/2}\}$$

$$y_{n+1} = y_n + hk_1/2$$

If the two algorithms TR and ER are to be applied with constant stepsize, their minimum storage requirements are, for a problem with dimension k

TR: D(A) + D(LU) + k

ER: D(LU) + 2k

where D(.) denotes the storage for the computational step involving matrix A and the LU factorization of the rational function denominator, respectively.

Table 3.1 indicates the rough operations counts in terms of multiplicative operations for the matrix operations for the two different

implementations, TR and ER, for;

(1), nxn tridiagonal coefficient matrix with re-decomposition of the coefficient matrix at each step;

(2), nxn tridiagonal matrix using a stored LU factorization;

(3), a general nxn dense coefficient matrix for which the LU decomposition has been found.

Table 3.1

	(1)	(2)	(3)
TR	$8n + O(1)$	$6n + O(1)$	$2n^2 + O(n)$
ER	$6n + O(1)$	$4n + O(1)$	$n^2 + O(n)$

The real advantage in cost of ER over TR is somewhat less than the indication given by the table, as a significant overhead is encountered in inhomogeneous problems from the evaluation of forcing terms. The first column indicates a 25% saving in computational cost, while experiments have shown the saving to be more of the order of 12.5%, in the case of simple heat-conduction problems. This entry in Table 3.1 is roughly proportional to the factorization cost, rather than simply the back-substitution, and for a full matrix which has been factored in a pre-processing step the saving in both substitution cost and storage is more significant.

3.3 Stability of ER-Trapezoidal Rule

ER is so-named because it is an extrapolation procedure. To identify it with the TR form, and establish that no instability arises, ER may be viewed as a multilevel method.

Define $w_n = (y_n^T \mid v^T)^T$

where v is an arbitrary nx1 column vector. Then

$$\begin{pmatrix} (I - hA/2) & 0 \\ 2I & -I \end{pmatrix} \begin{pmatrix} y_{n+\frac{1}{2}} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & I \\ 0 & I \end{pmatrix} \begin{pmatrix} v \\ y_n \end{pmatrix}$$

and since (I - hA/2) is non-singular, by hypothesis,

$$\begin{pmatrix} I - hA/2 & 0 \\ 2I & -I \end{pmatrix}^{-1} = \begin{pmatrix} (I - hA/2)^{-1} & 0 \\ 2(I - hA/2)^{-1} & -I \end{pmatrix} .$$

ER is thus equivalent to

$$w_{n+1} = \begin{pmatrix} 0 & (I - hA/2)^{-1} \\ 0 & 2(I - hA/2)^{-1} - I \end{pmatrix} w_n \quad , \qquad 3.3.1$$

ie. $w_{n+1} = Mw_n$.

We establish the correspondence between the eigenvalues and eigenvectors of M and those of TR.

Assume that the matrix A has simple structure, and that its eigenvalue-eigenvector pairs are denoted by $\langle \lambda_i, z_i \rangle$, $i = 1, \ldots, k$. Then the matrix M has n eigenvectors $w^T = (\underline{0}^T, \underline{z}^T)^T$ associated with non-zero eigenvalues, and an nxn null space spanned by $(v^T, 0^T)^T$ where the collection of vectors v_i form a linearly independent set, spanning the null space of M.

Consequently,

$Mw = \mu w$ for non-zero μ implies that

$$\{-I + 2(I - hA/2)^{-1}\}z_i = \mu_i z_i$$

where the z-vectors are eigenvectors of A, and thus,

$$-1 + 2(1 - h\lambda_i/2)^{-1} = \mu_i \ . \qquad 3.3.2$$

These k eigenvalues are identical for the iteration matrix for TR,

$$(1 - hA/2)^{-1}(1 + hA/2)z_i = \mu_i z_i, \qquad 3.3.3$$

the remaining n eigenvalues being zero.

Thus, we have proven

Theorem 3.1 The marching procedures ER and TR have the same non-zero eigenvalues of their amplification matrices, and ER is a stable variant of TR.

3.4 The Stability of Mixed Operations

The implementation of TR trapezoidal rule requires the performing of the two operations:

$$x = Ab \qquad 3.4.1$$

$$A'x' = b' \qquad 3.4.2$$

in sequence.

Either operation can be written:

$$z = Pf \qquad 3.4.3$$

where 3.4.2 is assumed to have a non-singular coefficient matrix.

Instead of the exact solution of 3.4.3 we compute a quantity $z + \delta z$, which exactly satisfies

$$z + \delta z = P(f + \delta f). \qquad 3.4.4$$

Hence, we have

$$\delta z = P\delta f$$

where $$||\delta z|| \leq ||P||.||\delta f|| \qquad 3.4.5$$

and $||P||$ is computed in a suitable compatible matrix norm. As in Theorem 1.3, there can be exact equality in 3.4.5, if δf is in the maximizing direction of $||P||$ and the matrix norm is subordinate.

The reason for this section is the fact that Theorem 1.3 is a "worst case" theorem, in that the right-hand side and the perturbation on it are assumed to be in the direction of maximum relative error. Also, for such methods as the trapezoidal rule, the numerator factor can be singular, and so any sensitivity analysis involving the mixed operations must avoid inverses of numerator factors.

Continuing, we must construct such an analysis which doesn't have these problems.

Assume that the solution to equation 3.4.4 is non-zero. Then

$$||\delta z|| \leq \frac{||P||}{||z+\delta z||} ||\delta f|| ||z+\delta z|| \quad .$$

Apply the triangle inequality, and assume that $||z|| > ||\delta z||$ to obtain

$$\frac{||\delta z||}{||z||} \leq \frac{\gamma}{1-\gamma} \quad , \qquad 3.4.6$$

where

$$\gamma = \frac{||P||}{||z+\delta z||} ||\delta f|| \quad . \qquad 3.4.7$$

We prove the following lemma.

Lemma 3.1 For $||z|| > ||\delta z||$, there exists a θ, $|\theta| \leq 1$, such that

$$\frac{||P|| ||f||}{||z + \delta z||} = \frac{||P|| ||f||}{||z||} \frac{1}{1 + \theta \frac{||\delta z||}{||z||}} \qquad 3.4.8$$

Proof

$$||z|| - ||\delta z|| \leq ||z + \delta z|| \leq ||z|| + ||\delta z||,$$

$$-||\delta z|| \leq ||z + \delta z|| - ||z|| \leq ||\delta z||$$

Therefore, there exists a θ satisfying the hypothesis of the lemma, and

$$\frac{||z + \delta z||}{||P||\,||f||} - \frac{||z||}{||P||\,||f||} = \theta \frac{||\delta z||}{||P||\,||f||} \cdot \frac{||z||}{||z||}$$

Assuming $||P||$ and $||f||$ are nonzero, inversion of this relation establishes the lemma.

Rewriting equation 3.4.7 as

$$\gamma = \frac{||P||\,||f||}{||z + \delta z||} \cdot \frac{||\delta f||}{||f||} \qquad 3.4.9$$

we see that two <u>sensitivity ratios</u> have been defined:

$$K_c(P|f) = \frac{||P||\,||f||}{||z + \delta z||} \qquad 3.4.10$$

and

$$K(P|f) = \frac{||P||\,||f||}{||z||} \qquad 3.4.11$$

We note that if $\frac{||\delta z||}{||z||}$ is small,

$$K_c(P|f) \doteq K(P|f)$$

Also, it may be noted that $K(P|f)$ behaves exactly like the condition number $K(P)$, when $K(P)$ exists (Chapter 1), for instance, the following is true:

$$1 \leq K(P|f) \leq K(P) \qquad 3.4.12$$

The sensitivity ratio is not a substitute for the condition number associated with the linear equations problem. Rather, it is a tool for

comparing the computational schemes for sensitivity to perturbations in the input data. For a genuinely ill-conditioned linear system, provided a stable decomposition exists, the solution to equation 3.4.2 might have to be refined by iterative improvement [17]. The analysis presented here is to show that the best solution available in a fixed precision arithmetic is affected by the computational ordering. The trapezoidal rule is the model problem for the sensitivity of mixed operations, and specifically we consider the unforced linear system, $y' = Ay$, where A has eigenvalues with negative real part. Assume that the problem is ill-conditioned, i.e., A has at least one small eigenvalue and one large eigenvalue and h is such that both $I - hA$ and $I + hA$ have large norm.

The explicit step of TR can be computed to high accuracy, and neglecting $(1 - \gamma)$ in 3.4.6,

$$\frac{||\delta y_{n+\frac{1}{2}}||}{||y_{n+\frac{1}{2}}||} \leq K((I + hA/2)|y_n) \frac{||\delta y_n||}{||y_n||} \qquad 3.4.13$$

If y_n does not contain a significant component in the direction of the maximizing eigenvector, $K((I + hA/2|y_n)$ is large. If it does, $||y_{n+\frac{1}{2}}||$ is large, and $K(I - hA/2)^{-1}|y_{n+\frac{1}{2}})$ is large. Hence,

$$\frac{||\delta y_{n+1}||}{||y_{n+1}||} \leq K(I - hA/2)^{-1}|y_{n+\frac{1}{2}})K(I - hA/2|y_n)\frac{||\delta y_n||}{||y_n||}$$

is large, establishing the following theorem.

Theorem 3.2 For the TR ordering of the model problem, one of the substeps 3.3.1 or 3.3.2 for arbitrary y_n must be ill-conditioned if A is an ill-conditioned coefficient matrix.

Continuing, we prove

Theorem 3.3 If $K((I - hA/2)^{-1}|y_n)$ is small and $||y_n||/||y_{n+1}||$ is also not large, the ER computational ordering has a computed solution with low sensitivity to computational error.

Proof

From the second step in the ER algorithm

$$y_{n+\frac{1}{2}} = \tfrac{1}{2}(y_{n+1} + y_n)$$

and

$$\delta y_{n+\frac{1}{2}} = \tfrac{1}{2}\delta y_{n+1}.$$

Hence,

$$||\delta y_{n+1}|| \leq \frac{||\delta y_{n+\frac{1}{2}}||}{||y_{n+\frac{1}{2}}||} \cdot \left(1 + \frac{||y_n||}{||y_{n+1}||}\right)||y_{n+1}||. \quad 3.4.14$$

If we neglect the computational error in the expression

$$\frac{||y_n||}{||y_{n+1}||},$$

then we can deduce from equation 3.4.13 that if the first factor on the right-hand side of equation 3.4.14 is small, and the ratio $||y_n||/||y_{n+1}||$ is not large, then $||\delta y_{n+1}||/||y_{n+1}||$ is small. □

Conversely, since we can compute $y_{n+\frac{1}{2}}$, and hence y_{n+1} to full t-digit accuracy, a large value $||y_n||/||y_{n+1}||$ will indicate a measure of the unsuitably of the trapezoidal approximation as an approximation to exp(hA) in such cases.

The result of these two theorems can be summarized, together with the operations counts for the two variants of the trapezoidal rule for linear systems, by simply stating that in virtually all circumstances the avoidance of the explicit calculation of numerator factors is the best strategy. The situation of a transient problem, where the trapezoidal rule is not a good algorithm to apply, is not an exception. In such a case the sensitive step may be computed using techniques which guarantee an accurate solution, and the vector addition may be performed with little error.

3.5 Extrapolation Techniques

The technique of increasing the order of a method by extrapolation results in an increase in solution accuracy with a moderate amount of computational effort. The one-step two-step solution carried independently in time is an often-used method for estimating the error in a solution procedure. Some shortcomings in the use of local extrapolation are given in [36]. The local step-by -step extrapolation for the trapezoidal rule is a method of obtaining order-4 accuracy in a forced linear system, provided the time-step is such that the method isn't unstable for the problem.

The Padé (1,1) approximation in ER form for first one step and then two steps can be written for argument z as

$$P_{11}(z) = 2(1 - z/2)^{-1} - 1 , \qquad 3.5.1$$

and

$$P_{11}^{2}(z/2) = 4(1 - z/4)^{-2} - 4(1 - z/4)^{-1} + 1 \cdot \qquad 3.5.2$$

To extrapolate to obtain a fourth-order method, we use $(4/3)P_{11}^{2}(z/2) - (1/3)P_{11}(z)$ to obtain

$$E_{11}(z) = (1 - z/4)^{-2}(1 - z/2)^{-1}(1 - \frac{3}{16} z^2 - \frac{5}{96} z^3) \qquad 3.5.3$$

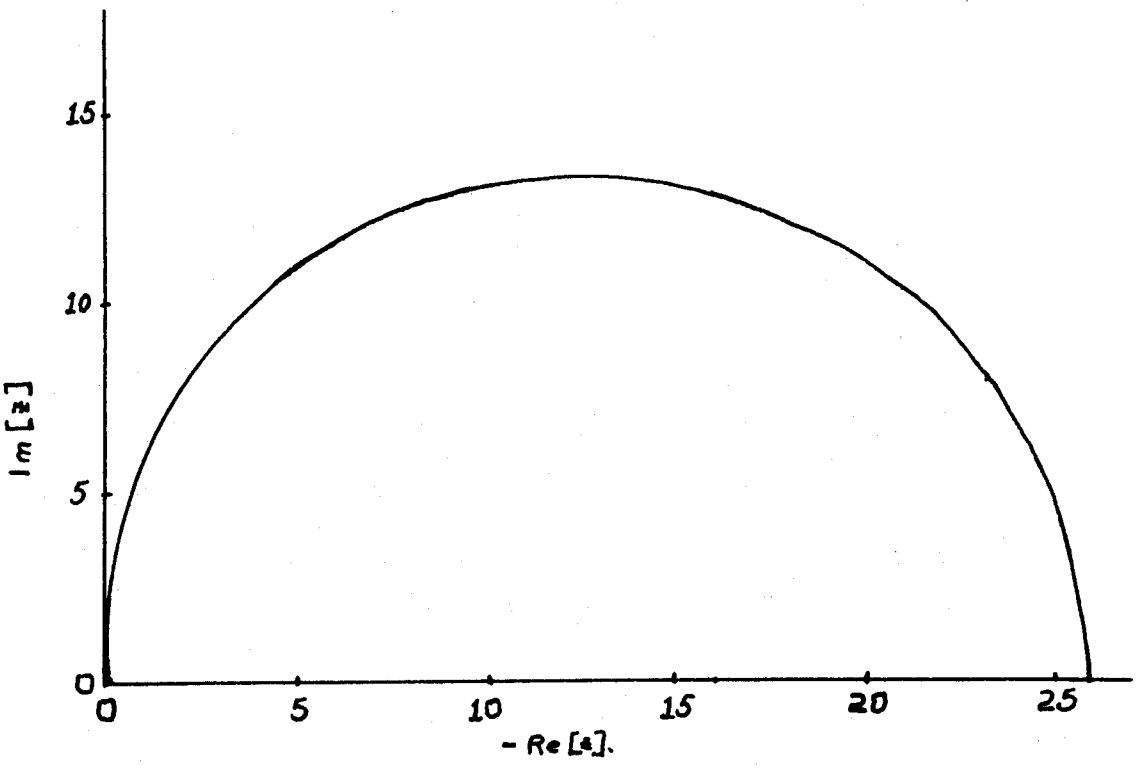
whose truncation error term for small argument is $1/320z^5$.

It is easy to verify that $E_{11}(z)$ is not A-stable, since

$$\lim_{\mathrm{Re}[z]\to\infty} E_{11}(z) = \frac{5}{3} \qquad 3.5.4$$

FIGURE 3.1

STABILITY REGION FOR LOCAL EXTRAPOLATION



(Figure is symmetric with respect to real axis.)

Figure 3.1 is a graph of the region of stability of $E_{11}(z)$ for $Re[z] < 0$. For heat-conduction problems, the condition $||hA|| \leq 25$ (approximately) results in a stable algorithm.

If the <u>average</u> of the two rationals is taken, rather than the extrapolated value, a strongly A-stable method results, since the averaged approximation is L-acceptable. The order of the averaged method is still two. The approximation is given by

$$A_{11}(z) = (1 - z/4)^{-2}(1 - z/2)^{-1}(1 - \frac{3}{16} z^2) , \qquad 3.5.5$$

with truncation error for small argument $-(5/96)z^3$. It is interesting to compare the two rational approximations to note that the term responsible for the source of non-A-stability can be simply isolated. From equations 3.5.4 and 3.5.5

$$\lim_{Re[z]\to-\infty} E_{11}(z) - A_{11}(z) = \frac{5}{3}$$

and

$$\lim_{Re[z]\to-\infty} A_{11}(z) = 0, \text{ hence, the term}$$

$$(1 - z/4)^{-2}(1 - z/2)^{-1}(\frac{5}{96} z^3)$$

is the source of the potential instability.

Local extrapolation of two Padé(1,1) solutions gives an approximation to the Padé(2,2) solution for a linear system:

$$(I - \frac{hA}{2} + \frac{h^2A^2}{12})u_{n+1} = (I + \frac{hA}{2} + \frac{h^2A^2}{12})\, u_n .$$

If we let u^*_{n+1} be the Padé(1,1) solution for stepsize h, and u^{**}_{n+1} be the solution for two steps of length h/2, then we denote by

u_{n+1} the extrapolated solution:

$$u_{n+1} = \frac{4}{3} u^{**}_{n+1} - \frac{1}{3} u^{*}_{n} \ .$$

Then u_{n+1} exactly satisfies

$$(1 - \frac{hA}{2} + \frac{h^2A^2}{12})u_{n+1} = (1 + \frac{hA}{2} + \frac{h^2A^2}{12})u_n + \frac{1}{36} h^2A^2(u^{**}_{n+1} - u^{*}_{n+1}) \ . \qquad 3.5.6$$

The [0,1] moment equations for $E_{11}(z)$ are, setting $q(z) = (1 - z/4)^2(1 - z/2)$:

$$\begin{aligned} q(z)M_o(z) &= 1 - \frac{z}{2} - \frac{z^2}{48} \quad ; \\ q(z)M_1(z) &= \frac{1}{2} - \frac{1}{3} z + \frac{z^2}{32} \ ; \\ q(z)M_2(z) &= \frac{1}{3} - \frac{1}{4} z + \frac{z^3}{32} \ ; \\ q(z)M_3(z) &= \frac{1}{4} - \frac{7}{32} z + \frac{z^3}{32} . \end{aligned} \qquad 3.5.7$$

For the nodes $\{0,\frac{1}{2},1\}$, (Simpson's rule), the weight functions derived from the moments are:

$$\begin{aligned} q(z)w_o &= \frac{1}{6} - \frac{z^2}{96} \quad ; \\ q(z)w_1 &= \frac{2}{3}(1 - \frac{z}{2}) \quad ; \\ q(z)w_2 &= \frac{1}{6} - \frac{z}{6} + \frac{z^2}{32} \ . \end{aligned} \qquad 3.5.8$$

A quick manipulation establishes the following two observations and a theorem.

Remark 3.1 For the Simpson nodes, the weights for $E_{11}(z)$ from the moment equation applied to the forcing terms for the inhomogeneous linear system $y' = Ay + f(t)$ agree with the weights derived from the application of the trapezoidal forcing when the extrapolation procedure is applied.

Theorem 3.4 When the trapezoidal rule is applied with the trapezoidal rule for forcing terms and extrapolated as in equation 3.5.3, polynomial forcing terms of polynomial degree ≤ 2 are exactly integrated.

Proof

For $i = 0$, 1, and 2, the Simpson nodes produce for $E_{11}(z)$ the same moments as the application of the trapezoidal rule with trapezoidal evaluation of inhomogeneous terms. But the moment equation for $i = 3$ is not satisfied by these weights:

$$\frac{1}{8} w_1 + w_2 = (1 - \frac{z}{4})^{-1}(\frac{1}{4} - \frac{5}{24} z + \frac{z^2}{32}) \neq M_3(z) \qquad \square$$

The mid-point rule quadrature for the trapezoidal rule doesn't extrapolate in this fashion.

Remark 3.2 The extrapolated mid-point rule for inhomogeneous linear systems integrates polynomial forcing for linear polynomials only.

There are no other obvious approximations which are based on linear combinations of trapezoidal approximations, and the author has been unsuccessful in obtaining any similar approximations using elements of the Padé table in linear combination. In the next section we discuss an order-2 approximation family with real poles which is strongly A-stable.

3.6 The L21 Approximation

We consider the class of rational functions defined by

$$R_{21}(z) = (1 - bz)^{-1}(1 - cz)^{-1}(1 + az), \qquad 3.6.1$$

for a real, $a > 0$

$\mathrm{Re}[b], \mathrm{Re}[c] > 0.$

The conditions that 3.6.1 matches the Maclaurin expansion of exp(z) through terms $O(z^3)$ are

$$\begin{aligned} a + b + c &= 1 \qquad \text{(linear)} \\ \frac{1}{2} - (b + c) + bc &= 0 \qquad \text{(quadratic)} \end{aligned} \qquad 3.6.2a$$

and, for the cubic term, the condition is

$$\frac{1}{6} - (b + c)/2 + bc = 0 \quad \text{(cubic)} \qquad 3.6.2b$$

The first two equations yield the following characterization of b and c in terms of a:

$$b = \frac{1}{2}(1 - a + \sqrt{a^2 + 2a - 1})$$

$$c = \frac{1}{2}(1 - a - \sqrt{a^2 + 2a - 1})$$

The leading truncation error term is then, from 3.6.2b

$$T(z) = (\frac{1}{6} - \frac{a}{2})z^3.$$

We note the following special cases, expressed as a function of the parameter a.

(1.) For a = 1/3, 3.6.1 is the Padé(2,1) approximation (T(z) = 0).

(2.) For a = 1/2, 3.6.1 reduces to the trapezoidal or Padé(1,1) approximation.

(3.) For a > 1/2, the approximation is not A-acceptable (c < 0).

(4.) For $a = 0$, we have the Padé(2,0) approximation.

(5.) For $\sqrt{2} - 1 \leq a \leq 1/2$, the approximation has real poles, and if $a \leq 1/2 - \delta$, for δ a small positive number, the approximation is L-acceptable.

(6.) For $a = \sqrt{2} - 1$, the approximation has repeated linear denominator factors.

It is this last case which we examine in detail, and we will call it the L21 approximation:

$$L21(z) = (1 - (1 - \frac{1}{\sqrt{2}})z)^{-2}(1 + (\sqrt{2} - 1)z) \qquad 3.6.3$$

For the L21 approximation, $T(z) = \{\frac{1}{6} - (\frac{\sqrt{2} - 1}{2})\}z^3$ and the coefficient of z^3 in $T(z)$ has a numerical value -0.04044 (approximately), which is about half that for the trapezoidal rule.

We now establish the stability of the L21 approximation.

Theorem 3.5 The linear approximation L21 is A-acceptable and L-acceptable.

Proof

Let $z = x + iy$ be a complex number such that $x < 0$.
Set $|z|^2 = x^2 + y^2$.

Define:

$$N(z) = 1 + (\sqrt{2} - 1)z$$

$$D(z) = (1 - (1 - \frac{1}{\sqrt{2}})z)^2.$$

Then,

$$|N(z)|^2 = 1 + (2\sqrt{2} - 2)x + (3 - 2\sqrt{2})|z|^2,$$

and $$|D(z)|^2 = |1 - (1 - \frac{1}{\sqrt{2}})z|^4,$$

ie. $$|D(z)|^2 = \{(1 - (2 - \sqrt{2})x + \frac{1}{2}(3 - 2\sqrt{2})|z|^2\}^2.$$

This expression can be rewritten

$$|D(z)|^2 = |N(z)|^2 - 2x + (\frac{1}{2}(3 - 2\sqrt{2})|z|^2 - (2 - \sqrt{2})x)^2.$$

Since $x < 0$, $|D(z)| \geq |N(z)|$ and the first part of the theorem is proved.

Continuing, since $|D(z)|^2 = |N(z)|^2 + P(z)$, and $P(z)$ is real and positive and $P(z) = 0|z|^4$ for large $|z|$, L21 is L-acceptable. □

From the moment equations, we obtain for constant and linear moments, if we set $q(z) = (1 - (1 - \frac{1}{\sqrt{2}}z)^2$:

$$q(z)M_0(z) = 1 - (1 - \frac{1}{\sqrt{2}})^2 z$$

$$q(z)M_1(z) = \frac{1}{2} - (1 - \frac{1}{\sqrt{2}})^2 z.$$

A particularly useful set of [0,1] abscissae for determining weight functions is the set $\{(1 - \frac{1}{\sqrt{2}}), (2 - \sqrt{2})\}$.

Solving the moment equations, we obtain

$$q(z)w_o = (1 - \frac{1}{\sqrt{2}})(1 + (\sqrt{2} - 1)z) ,$$

$$q(z)w_1 = \frac{1}{\sqrt{2}} (1 - (1 - \frac{1}{\sqrt{2}})z) .$$

Hence the algorithm becomes:

$$(I - (1 - \frac{1}{\sqrt{2}})hA)^2 y_{n+1} = (I + (\sqrt{2} - 1)hA)y_n$$

$$+ h\{w_o f_{n+(1-\frac{1}{\sqrt{2}})} + w_1 f_{n+(2-\sqrt{2})}\} . \qquad 3.6.5$$

For these particular abscissae, the formula reduces to the multiple step formula:

$$(I - (1 - \frac{1}{\sqrt{2}})hA)y^* = y_n + (1 - \frac{1}{\sqrt{2}})hf_{n+(1 - \frac{1}{\sqrt{2}})} ,$$

$$(I - (1 - \frac{1}{\sqrt{2}})hA)y^{**} = y^* + (1-\frac{1}{\sqrt{2}})h\, f_{n+(2-\sqrt{2})} , \qquad 3.6.6$$

$$y_{n+1} = (\sqrt{2} + 1)y^{**} - \sqrt{2}y^* .$$

The two stages in the algorithm become the implicit Euler's method at $t = \{n + (1 - \frac{1}{\sqrt{2}})\}h$. and $t = \{n + (2 - \sqrt{2})\}h$, respectively. The separation is achieved because the choice of abscissae produces weights which are factors of L21(z).

The L21 approximation error term has for argument with small norm an error whose coefficient is roughly half that for the trapezoidal rule. It is a preferable method in several situations. When the matrix is stiff, or when a solution is required for large time, the L-acceptability of L21 is important. When compared to the ER implementation of Crank-

Nicholson with mid-point evaluation of the forcing on the same time step, it is roughly 0.6 as efficient. This is the worst case, however, and when the two algorithms are compared over long time periods, or when L21 is compared to TR implementations of the trapezoidal rule, its efficiency in terms of stability and ease of implementation make it a useful algorithm.

3.7 Applications to Non-Linear Systems

To date, the separable form of the algorithm L21 has not been tried on nonlinear problems. However, the use of Lawson's linearization transformation and the separability condition yields plausible formulations for non-linear problems.

Recalling equation 2.2.2, for the system

$$y' = f(t,y),$$

we choose an appropriate linearization A so that $\frac{\partial f}{\partial y} - A$ has eigenvalues in the left-half plane in a t-interval containing $\{t_n, t_{n+1}\}$ and apply an algorithm based on a rational approximation such as L21, together with the evaluation of $f(t,y) - Ay$ as a forcing term.

We can apply multistep formulations, explicit or implicit, using previously determined values of y at t_n, t_{n-1}, etc., provided starting values are generated by a single-step method of the same order as the multistep formulae.

Lees [31] obtained a single-pass algorithm based on the trapezoidal rule, for nonlinear systems of the form

$$y' = P(y)y \qquad 3.7.1$$

which requires the prediction of $y(t_{n+\frac{1}{2}})$ using y-values on $\{t_{n-1}, t_n\}$ in order to evaluate the nonlinear part contained in the forcing function.

Using the trapezoidal rule with an approximation A to the Jacobian of f, we obtain,

for $u(t) = \{P(y(t)) - A\}y(t)$,

$$(I - \frac{hA}{2})y_{n+1} = (I + \frac{hA}{2})y_n + h(P(y_{n+\frac{1}{2}}) - A)y_{n+\frac{1}{2}} \quad . \qquad 3.7.2$$

If $\bar{y}_{n+\frac{1}{2}}$ replaces $y_{n+\frac{1}{2}}$ in the substitution in P(.), we obtain, upon writing equation 3.7.2 in ER form, and simplifying:

$$(I - \frac{h}{2} P(\bar{y}_{n+\frac{1}{2}}))y_{n+\frac{1}{2}} = y_n \quad ,$$

$$y_{n+1} = 2y_{n+\frac{1}{2}} - y_n \quad . \qquad 3.7.3$$

This is the essence of the Lees extrapolated Crank-Nicholson method, and the updated value of $\bar{y}_{n+\frac{1}{2}}$ is obtained from $\{y_{n-1}, y_n\}$ by linear extrapolation:

$$\bar{y}_{n+\frac{1}{2}} = \frac{3}{2} y_n - \frac{1}{2} y_{n-1} \quad .$$

In an analogous fashion, the L21 approximation with the quadrature points chosen in Section 3.6 may be written to solve equation 3.7.1.

To simplify the expression, let

$$n_1 = [n + (1 - \frac{1}{\sqrt{2}})] \ ,$$

$$n_2 = [n + (2 - \sqrt{2})] \ ,$$

$$n_1 - 1 = [n - 1 + (1 - \frac{1}{\sqrt{2}})] \ ,$$

and

$$n_2 - 1 = [n - 1 + (2 - \sqrt{2})] \ .$$

Then, using equations 3.6.6 and simplifying as in 3.7.3,

$$(I - (1 - \frac{1}{\sqrt{2}})hP(\bar{y}_{n_1}))y^*_{n_1} = y_n \ ,$$

$$(I - (1 - \frac{1}{\sqrt{2}})hP(\bar{y}_{n_2}))y^{**}_{n_2} = y^*_{n_1}, \qquad 3.7.4$$

$$y_{n+1} = (\sqrt{2} + 1)y^{**}_{n_2} - \sqrt{2}y^*_{n_1} \ ,$$

and

$$\bar{y}_{n_1} = (\sqrt{2} + 2)y^{**}_{n_2-1} - (\sqrt{2} + 1)y^*_{n_1-1} \ , \qquad 3.7.4a$$

$$\bar{y}_{n_2} = 2y_{n_1} - y_n \ . \qquad 3.7.4b$$

We note that the value $\bar{y}_{n_2}$ has second-order accuracy, since it is given by the Crank-Nicholson formula 3.7.3 with $\tilde{h} = (1 - \frac{1}{\sqrt{2}})h$.

If P is a function of t rather than y, both 3.7.3 and 3.7.4 have their coefficient matrices given analytically by $P(t_{n+\frac{1}{2}})$, and $\{P(t_{n_1}), P(t_{n_2})\}$ respectively.

In both cases, 3.7.4 behaves like an implicit Runge-Kutta formula, where y^* and y^{**} are the unknown quantities, rather than "derivative" values as in IRK methods.

3.8 Linear Systems with Implicit Matrix Factors

A more general class of linear differential systems is given by the equation:

$$My' = Ny + g(t) \ . \qquad 3.8.1$$

The matrix coefficients M and -N will be assumed to be positive definite symmetric for the time being, as this is sufficient to guarantee a stable coefficient matrix $M^{-1}N$ [38].

For these conditions on M and N the system is, formally,

$$y' = M^{-1}Ny + M^{-1}g(t) \qquad 3.8.2$$

Thus, the substitution into the rational function approximation E(z) is for argument $M^{-1}N$. If the rational approximation has only linear numerator and denominator factors, one approach which avoids the explicit calculation of M is to write the rational approximation in the form $q^{-1}pq^{-1}p$ ... where q represents a denominator factor, and p a numerator factor. We need not specifically exclude the factorization of irreducible quadratics into complex linear factors, although the feasibility of implementation in such cases is reserved for Chapter four. We observe that, if the number of numerator and denominator factors are the same,

$$(I - \alpha M^{-1}N)^{-1} = (M - \alpha N)^{-1}M$$

and $\qquad 3.8.3$

$$(I + \beta M^{-1}N) = M^{-1}(M + \beta N),$$

and the M terms cancel. This is also true for the forcing term, as the M associated with the forcing term acts in the same manner as for numerator factors, together with

Remark 3.3. For an A-acceptable approximation to exp(z), the weight functions derived from equations 2.2.3 have degree $\leq m - 1$, where m is the denominator degree of the rational approximation E(z).

The need to apply numerator and denominator factors alternately

increases the computational cost and/or storage requirements considerably. For the trapezoidal rule and L21, the partial fraction decomposition of the rational approximation and its weights effects a computational ordering which is useful in this case. Assuming that a linear combination of forcing terms is obtained from the partial fraction decomposition of the weights, and a term associated with the dependent variable y, either at t = mh or some intermediate (known) point can be written formally as

$$(I - \alpha M^{-1}N)y^{*} = y + \beta M^{-1}f \qquad 3.8.4$$

which may be rewritten

$$(M - \alpha N)y^{*} = My + \beta f. \qquad 3.8.5$$

Then, as in 3.8.3 the terms in the expansion of the partial fraction decomposition may be summed to yield the solution to 3.8.1 at the next level.

For the trapezoidal rule for implicit matrix argument the computational cost increases modestly in the ER form, and the computational advantages over the TR form are now substantial.

This is a somewhat different approach to the treatment of the implicit coefficient matrix factor M than in, e.g., [8]. It is not surprising to note that the ordinary three-point spatial discretization of the equation

$$\begin{aligned} u_t &= u_{xx} \quad \text{on } [0,1]\ x[0,T] \quad , \\ u(x,0) &= g(x) \quad , \\ u(0,t) &= f_0(t) \, , \\ u(1,t) &= f_1(t) \, . \end{aligned} \qquad 3.8.6$$

is of the form of equation 3.8.1. A full discretization in t and x applied to 3.8.6 can in fact be visualized as a specific case of equation 3.8.3, and one of the common coding procedures to solve the discretized problem by an implicit algorithm is in fact similar in structure to either 3.8.3 or 3.8.5. Apply the three-point spatial discretization at $t = t_n$, on the uniform mesh $i\Delta x$, $i = 0, 1, \ldots, (m + 1)$; $(m + 1)\Delta x = 1$, and use the implicit Euler's method, setting $r = \Delta t/\Delta x^2$.

$$(M - rN)u_{n+1} = Mu_n + \Delta t h(t_{n+1}) \qquad 3.8.7$$

where

$$M = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ & I_{m-1} & \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

$$N = \begin{pmatrix} -\Delta x^2 & 0 & \cdots & & 0 \\ 1 & -2 & 1 & 0 & \cdots \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots \\ & & 1 & -2 & 1 \\ & & & 0 & -\Delta x^2 \end{pmatrix}$$

and

$$h(t) = \begin{pmatrix} f_0(t) \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ f_1(t) \end{pmatrix} .$$

Note that $Mu_i = 0$ at $i = 1$ and $i = m + 2$. Not surprisingly, this

formulation lends itself to efficient coding of the tridiagonal equations solver in that the first and last rows of the coefficient matrix are not special cases with respect to the loop structure used to perform the substitution steps in solving 3.8.7.

The form of equation 3.8.1 gives rise to a class of singular perturbation problems.

If in the matrix M of equation 3.8.7 applied to the semi-discretization in space of 3.8.6, the {1, 1} and {n + 2, n + 2} elements are non-zero but order ε for some small $\varepsilon > 0$, the problem would be a classical singular perturbation problem. While it is beyond the scope of this thesis to consider the case of singular perturbation problems, it is of interest to note that the form of equation 3.8.5 is appropriate to finding the solution to the reduced system for 3.8.7 for ε approaching zero. It is a well-known method for singular perturbation problems to find the solution for the $\varepsilon = 0$ case and theorems about its applicability to the more general system exist [37]. Our concern is that, for the problems to which we apply methods designed to follow solutions which persist in time, the form of the linear systems which are free of M^{-1} will be better conditioned numerically than 3.8.4, as the application of M^{-1} to the coefficient matrix N would, for such cases, generate ill-conditioning that is avoidable through the use of 3.8.5.

Remark 3.4 For differential equations of the more general form 3.8.1, which have a well-conditioned matrix N associated with the homogeneous right-hand side, but an ill-conditioned matrix M, the coefficient

matrix of the linear system 3.8.7 will be well conditioned for large time-step.

What we have, in effect, is an equilibration of the coefficient matrix in 3.8.7. The effect of this technique is to numerically solve the reduced system when the time step is large relative to the singular terms.

A more difficult problem arises in equation 3.8.6 when

$$\lim_{t\to 0^+} f_0(t) \neq g(0)$$

and

$$\lim_{t\to 0^+} f_1(t) \neq g(1).$$

If we replace the coefficient matrix M by $\tilde{M}$, where

$$\tilde{M}_{11} = \tilde{M}_{m+2\ m+2} = \varepsilon$$

then the differential equation becomes, for the variable u_0 and u_{n+2},

$$u_i' = -\frac{1}{\varepsilon} u_i + \frac{1}{\varepsilon} f_i(t), \quad u_i(0) = g(x_i) \qquad x_i = 0 \text{ or } 1$$

i.e., dropping the subscript i,

$$u' = -\frac{1}{\varepsilon} u + \frac{1}{\varepsilon} f(t)$$

which has the solution

$$u(t) = \exp\left(-\frac{1}{\varepsilon} t\right) g + \int_0^t \exp\{-\frac{1}{\varepsilon}(t - s)\, \frac{f(s)}{\varepsilon}\, ds \qquad 3.8.8$$

When evaluating equation 3.8.8 using a rational approximation E(z) to exp(z), we have, for E(z) and a suitable quadrature formula derived from it, exactness for a particular integral of u(t) for polynomial f of degree $\leq$ k (the order of the quadrature formula). For

this case, the behavior of the solution to 3.8.8 must be examined as $\varepsilon \to 0$. To do this, we must first prove

Lemma 3.2. If $f(t)$ is a polynomial in t, then for t finite and $\varepsilon > 0$

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\int_0^t \exp\{-\frac{1}{\varepsilon}(t - s)\}f(s)ds = f(t).$$

Proof

Without loss of generality, assume that

$f(t) = t^j$, $j \geq 0$, and define

$$u_p^j(t) \equiv \frac{1}{\varepsilon}\int_0^t \exp\{-\frac{1}{\varepsilon}(t - s)\}s^j ds,$$

then for $j = 0$

$$u_p^0(t) = 1 - \exp(-\frac{t}{\varepsilon}),$$

and

$$\lim_{\substack{\varepsilon \to 0^+ \\ t \leq T}} [1 - \exp(-\frac{t}{\varepsilon})] = 1.$$

Assuming the lemma statement is true for $j > 0$, then

$$u_p^{j+1}(t) = \frac{1}{\varepsilon}\int_0^t \exp\{-\frac{1}{\varepsilon}(t - s)\}s^{j+1} ds$$

$$u_p^{j+1} = \exp\{-\frac{1}{\varepsilon}(t - s)\}s^{j+1}\Big|_0^t - j\varepsilon\{\frac{1}{\varepsilon}\int_0^t \exp\{-\frac{1}{\varepsilon}(t - s)\}s^j ds;$$

i.e.,

$$u_p^{j+1}(t) = t^{j+1} - j\varepsilon u_p^j(t)$$

and $$\lim_{\varepsilon \to 0} u_p^{j+1}(t) = t^{j+1}, \quad \text{for } t \leq T,$$

since $j\varepsilon u_p^j(t)$ is finite, by the induction hypothesis. □

Corollary to Lemma 3.2. If f(t) is a continuous function on [0,T], then Lemma 3.2 still holds.

Proof

Apply the Weierstrass approximation theorem and the hypothesis of the Lemma for polynomial forcing. □

Hence, as $\varepsilon \to 0$, the particular integral $u_p(t)$ approaches f(t).

When exp(z) is replaced by E(z) in equation 3.8.8, the solution

$$U(t) = U_c(t) + U_p(t)$$

is exact for $U_p(t)$ under the conditions described and $U_p(0) = f(0)$. However, for $U_c(t)$,

$$U_c(t) = \lim_{\varepsilon \to 0} E(-\frac{t}{\varepsilon})g,$$

and, if E(z) is not L-acceptable,

i.e. if $\lim_{Re[z] \to -\infty} E(z) = \gamma, \gamma \neq 0,$

then, denoting the exact solution by u(t)

$$\lim_{\varepsilon \to 0} U(t) - u(t) = \gamma g$$

Hence, we have proven

Theorem 3.6 For nearly-singular implicit argument, L-acceptability of the rational approximation E(z), and a quadrature scheme of order $\leq k$ is necessary and sufficient for exactness of the numerical solution to equation 3.8.8 for f(t) a polynomial of degree $\leq k$.

From Chapter 1, the values of the {1, 1} and {m+2, m+2} elements of any rational function with this particular matrix argument $M^{-1}N$, are

the scalar quantities given by the approximation $E(M^{-1}N)$ to the complementary function in equation 2.8.2. Two situations may occur here. If the approximation is L-acceptable or is a Chebyshev approximation to exp(z), the approximation may not be seriously in error. If, however, it is not asymptotically correct, there will be an error on the boundary of the order of the asymptotic error. The formalism which corrects this problem in equation 3.8.1 applies to any formula with linear factors, e.g., the trapezoidal rule and the L21 algorithm. The M^{-1}-free property of the generalized algorithms is in effect the solution procedure for finding the solution to the reduced system, as we have mentioned.

For problems which have implicit matrix factors which are nearly singular the application of L-acceptable methods becomes important here, unless the reduced equations can be solved free of M^{-1}, and for this case, non-L-acceptable methods require unrealistic step-size constraints to prevent incorrectness of the asymptotic solution to the singular portion of the problem from contaminating the solution. The example of this section is chosen because the cause of the singularity is obvious, but it demonstrates all of the features that any system with implicit argument might possess.

CHAPTER 4

IRREDUCIBLE QUADRATIC FACTORS

4.1 Preliminary Considerations

The case of irreducible quadratic denominator factors in rational approximations with matrix argument gives rise to two additional implementation considerations over those of approximations with linear denominator factors. The cost of forming the irreducible quadratic factor and the correctness of the floating-point representation of the resulting coefficient matrix when solving the resulting linear system are both difficult questions.

We will look at alternative ways of accomplishing the factorization, and compare the cost and accuracy of each strategy.

4.2 Operation Times and Programming Overhead

The traditional methods of evaluating the cost of a numerical algorithm operate on the assumption that the cost of multiplications and divisions together are proportional to the overall cost, provided the numbers of additive operations and multiplicative operations are roughly the same. This is a valid assumption to some extent, but is dependent on both machine characteristics and the code which is generated to handle such overhead as, for example, array indexing.

The following table lists the cost of the single and double-precision floating point operations on the IBM/360-75 and

HONEYWELL-6050, respectively.

Table 4.1 Operation Times (μsec)

	H6050[+]		IBM 360/75[++]	
	S.P.	D.P.	S.P.	D.P.
+/-	2.2	2.2	1.17	1.17
x	3.2	6.2	2.05	4
÷	7.52	12.3	6.92	6.92

A preliminary check of the table would indicate that, for both machines, additive and multiplicative operations in single precision are of the same order. The IBM/360 has the interesting characteristic that single and double precision divide operations have the same cost. The variation in cost between additions and multiplications lie between a factor of one and three.

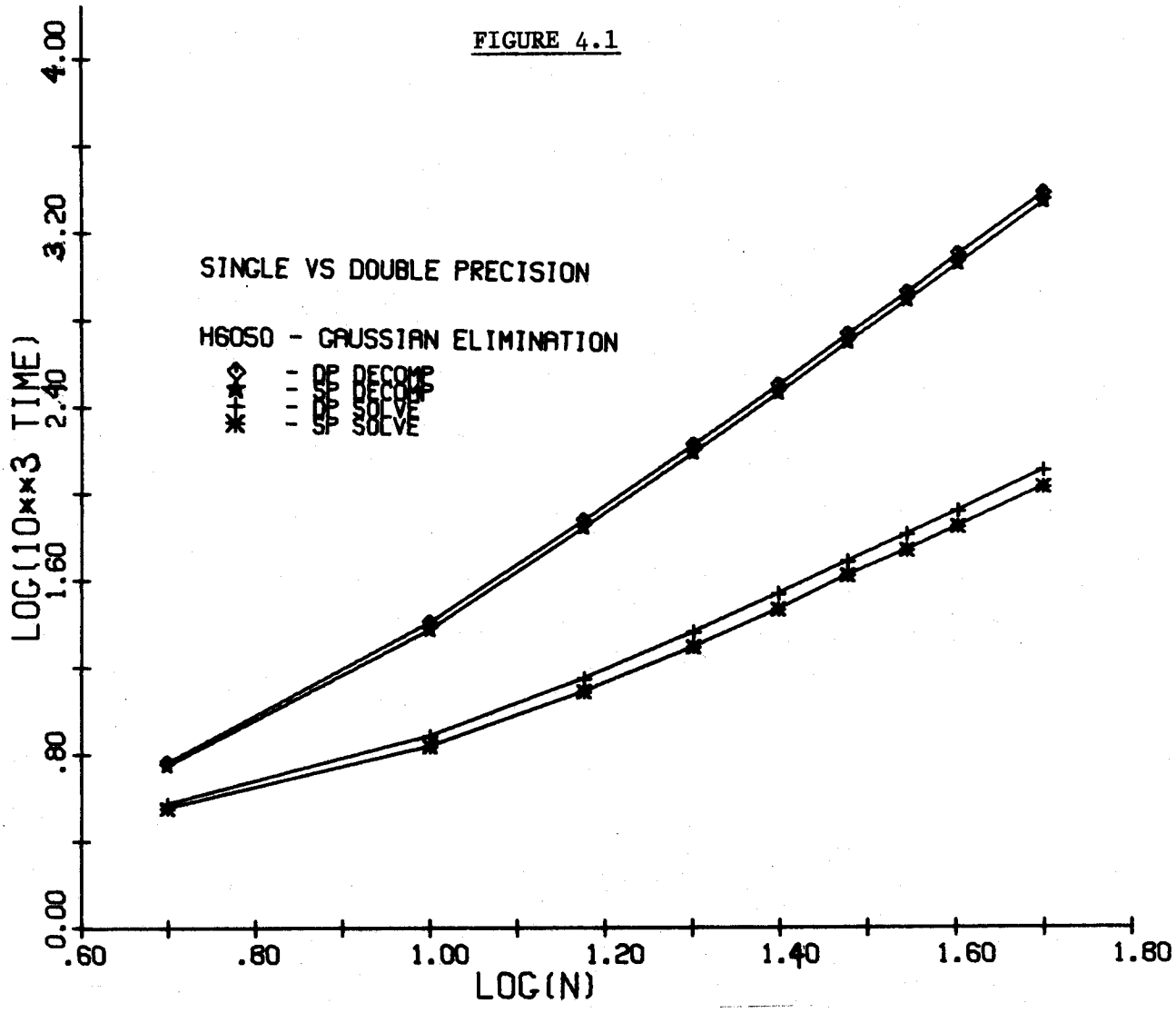
For a Gaussian elimination program, written in FORTRAN for example, there are other overhead costs. For dense coefficient matrix, the calculation of array pointers involves multiplicative operations, and the loading/storing of array elements tend to make the differences between single and double precision computation costs less than the table of operation times would indicate. Figure 3.1 is a graph of times in seconds for the decomposition and substitution phases for a typical linear equations solver, modelled after the FORTRAN code presented in [17]. For dimension 5 x 5 through 50 x 50, the code was

---

+,++
Source: IBM System/360 Model 75 Functional Characteristics
HONEYWELL Series 6000 Programming Reference Manual - Hardware

run completely in first single precision and double precision on the HONEYWELL-6050, using the time-sharing FORTRAN compiler at a time when minimum interference from other users could be anticipated. The time variations between single and double precision are of the order of 10% for problems of this dimension. For ill-conditioned systems of small dimension, the need to perform double-precision calculations to ensure a good result seems appropriate for a machine/compiler which is similar to the aforementioned, and the additional cost in the absence of storage constraints is minimal.

FIGURE 4.1



4.3 The Computational Cost of Irreducible Quadratic Factors

When considering evaluation procedures for rational functions of matrix argument associated with an approximation to exp(tA), there are two distinct computational considerations.

i) It is known à priori that the denominator matrix factors can be re-applied, as in constant coefficient systems for which it is known that the step-size need not be changed.

ii) The coefficient matrix is changed periodically, or the time-step changes; in either case the matrix factors must be re-evaluated.

Case i applies to methods based on approximations to exp(tA) for constant coefficient systems, for those instances when it is both feasible and economical to decompose and store the denominator factors. For problems of moderate dimension, and for sparse systems with modest fill-in characteristics (e.g. tridiagonal systems) Case i is applicable.

For systems of large dimension but with structural simplifications such as the Buneman algorithm applied to the multi-dimensional Poisson operator, or for variable matrix coefficients, Case ii is applicable. Global methods (Varga et al [12], or Lawson [30]) may fall into either category. They may operate in Case i mode for constant-coefficient systems with fixed step-length, or, where the speed consideration is marginal relative to the storage problems they may be of Case ii type. The problem of storage limitations in general, relegates a particular method to Case ii.

For Case i, decomposition time is essentially pre-processing time, and for Case ii the solution time must be estimated for both decomposition and substitution steps.

To conform with these separate cases estimates of cost based on operations counts will be broken up into three categories: initialization, factorization, and substitution. The synthesis of the costs to be evaluated into the broader picture will be carried out in Chapter 6.

The most general form of the equations we are to solve for irreducible quadratic factors of the rational function is given by

$$(t^2A^2 + \beta tA + \gamma I)y = \delta Ab_1 + \varepsilon b_2 \qquad 4.3.1$$

where β, γ, δ, and ε are real scalars, and the coefficient matrix for equation 4.3.1 is assumed nonsingular.

The first simplification that may be made for the problem is to assume that, for $t \neq 0$, equation 4.3.1 may be divided through by t^2, to make the scalar arguments in the rational function into simple functions of the variable t. This makes a change in the time-variable a more simple operation with respect to the matrix A: O(1) computation of time-changes at re-initialization time, and simpler representation of the matrix argument.

Equation 4.3.1 is rewritten

$$(A^2 + \beta A + \gamma I)y = \delta Ab_1 + \varepsilon b_2. \qquad 4.3.2$$

Observation 4.1 The overall computing cost of an irreducible quadratic factor is the cost in terms of real arithmetic operations to solve equation 4.3.2. This cost will be subdivided into (i) the initial overhead, (ii) the per-factor cost, and (iii) the substitution cost.

The following tables contain the cost estimates for the two examples we are considering, and for reference, the cost of a real tridiagonal decomposition and back-substitution.

Table 4.2

EQUATION 4.3.2 FOR NxN DENSE MATRIX

STEP TYPE	+,-	*	/
Initial Cost Overhead			
Matrix Squaring	$(n-1)^3$	n^3	0
Scalar Arguments	0	O(1)	0
Per Factor Cost			
Assembly of Quadratic	$n^2 + n$	$n^2 + n$	0
L-U Decomposition	$(\frac{1}{3}n^3 - \frac{n^2}{2} + \frac{7}{6}n)$	$(\frac{1}{3}n^3 - \frac{n^2}{2} + \frac{7}{6}n)$	$(\frac{n^2}{2} - \frac{n}{2})$
Substitution Cost per Factor			
Matrix-Vector Multiplication and Vector Assembly	$n^2 - n + 1$	$n^2 + 2n$	0
Substitution	$(n-1)^2$	$(n-1)^2$	n
Additional Initial Costs for Implicit Matrix Argument $M^{-1}N$			
L-U Decomposition of M	$(\frac{1}{3}n^3 - \frac{n^2}{2} + \frac{7}{6}n)$	$(\frac{1}{3}n^3 - \frac{n^2}{2} + \frac{7}{6}n)$	$(\frac{n^2}{2} - \frac{n}{2})$
Formation of $M^{-1}N$	$n(n-1)^2$	$n(n-1)^2$	n^2
Substitution $M^{-1}b$ (a per-factor cost)	$(n-1)^2$	$(n-1)^2$	n

Table 4.3

NxN TRIDIAGONAL COEFFICIENT MATRIX

Initial Cost Overhead	+,-	*	/
Matrix Squaring	4n - 2	9n - 4	0
Scalar Constants	0(1)	0(1)	0
Per Factor Cost			
Assembly of Quadratic	4n - 2	3n - 2	0
L-U Decomposition	4n - 7	4n - 7	2n - 1
Substitution Cost per Factor			
Matrix-Vector Multiplication	2(n - 1)	3n - 2	0
Vector Assembly	n	2n	0
Substitution	4n - 6	4n - 6	n

For later reference, the computational cost for a real symmetric tridiagonal nxn matrix is

Per Factor Cost	+	*	/
L-U Decomposition	n - 1	n - 1	n - 1
Substitution Cost	2(n - 1)	2(n - 1)	n

As has been noted, in general it is impossible to maintain sparsity for implicit matrix factors, when the matrices M and N do not commute or $M^{-1}N$ is full (Property P [21]).

4.4 The Existence of a Solution to the Complex Problem

Equation 4.3.1 has as coefficient matrix a denominator factor of a rational approximation whose poles are outside some region containing the

spectrum of the matrix A. The irreducible polynomial for the factor is

$$p(z) = z^2 + bz + c$$

or alternatively,

$$p(z) = (z - \gamma)^2 + \beta^2 \quad (\beta \neq 0) \ . \qquad 4.4.1$$

We first establish that the solution to 4.3.2 is unique.

Theorem 4.1 The linear system 4.3.2 has a unique solution for arbitrary non-zero right-hand side b if for λ an eigenvalue of A, $p(\lambda) \neq 0$.

Proof

From Chapter 1, P(A) is similar to its Jordan form,

$$P(A) = HP(J)H^{-1}$$

The diagonal elements of $(J - \gamma I)^2 + \beta^2 I$ are

$$(\lambda - \gamma)^2 + \beta^2$$

which are non-zero by hypothesis. P(J) is triangular, and thus has non-zero determinant. Therefore, the solution is non-zero and unique, and is denoted by

$$y = P(A)^{-1}b. \qquad \square$$

Theorem 4.2 Under the same hypothesis, the solution to the complex problem

$$\{A + \gamma I + i\beta I\}\{z_R + iz_I\} = b_R + ib_I$$

exists and is unique.

Proof

$$A + \gamma I + i\beta I = H(J + \gamma I + i\beta I)H^{-1}$$

If J is singular, then for some λ, $(\lambda - \gamma)^2 + \beta^2 = 0$, a contradiction.

We can assume for the remainder of this section that the irreducible quadratic is of the form

$$P(z) = z^2 + \beta^2, \beta \neq 0$$

and denote the associated matrix problem by

$$(A + i\beta I)(w_R + iw_I) = b_R + ib_I. \qquad 4.4.2$$

The following simple result, stated as a theorem, forms the basis for the use of complex elimination as an alternative to forming the quadratic matrix factor.

Theorem 4.3 For the complex linear system 4.4.2, the real and imaginary parts of the solution vector are equivalent to the real pair of systems

$$\begin{aligned} (A^2 + \beta^2 I)w_R &= Ab_R + \beta b_I \\ (A^2 + \beta^2 I)w_I &= Ab_I - \beta b_R \end{aligned} \qquad 4.4.3$$

Furthermore, there exist complex constants γ and δ such that the solution to equation 4.3.2 is given by w_R, where

$$(A + i\beta I)(w_R + iw_I) = \gamma b_1 + \delta b_2$$

Proof

Multiply 4.4.2 by $A - i\beta I$ and group real and imaginary parts. To obtain the second result, we recall that the right hand side of equation 4.3.2 has the same form as the right hand side of equation 4.4.3, with appropriate choice of γ and δ. □

4.5 The Sensitivity of Complex Elimination

The complex elimination algorithm can be formulated in terms of the equivalent real problem [1]

$$\begin{pmatrix} A & -\beta I \\ \beta I & A \end{pmatrix}\begin{pmatrix} w_R \\ w_I \end{pmatrix} = \begin{pmatrix} b_R \\ b_I \end{pmatrix} . \qquad 4.5.1$$

A quick manipulation establishes a correspondence between the singular values of the associated real and complex matrix factors,

$$(A - i\beta I)^T(A + i\beta I) = A^TA + \beta^2 I + i\beta(A^T - A)$$

and

$$\begin{pmatrix} A^T & +\beta I \\ -\beta I & A^T \end{pmatrix}\begin{pmatrix} A & -\beta I \\ \beta I & A \end{pmatrix} = \begin{pmatrix} A^TA + \beta^2 I & -\beta(A^T - A) \\ \beta(A^T - A) & A^TA + \beta^2 I \end{pmatrix}$$

The respective unitary and orthogonal matrices which diagonalize the complex and associated real forms are similarly related. If $U = U_R + iU_I$ for the complex problem then

$$Q = \begin{pmatrix} U_R & -U_I \\ U_I & U_R \end{pmatrix}$$

for the real problem.

Then we prove that for symmetric matrix argument both problems are well-conditioned in the Euclidean norm.

Theorem 4.4 If A is a symmetric matrix, then the singular values σ of A and $A + i\beta I$ are related by

$$\sigma^2[A + i\beta I] = \sigma^2(A) + \beta^2 . \qquad 4.5.2$$

Proof

The matrix $(A - i\beta I)^T(A + i\beta I)$ is, for symmetric A, equal to

the matrix

$$A^TA + \beta^2 I. \qquad \square$$

Corollary The 2-norm condition number for the complex problem for a real symmetric matrix A is given in terms of the singular values of A by

$$K_2(A + i\beta I) = \left(\frac{\sigma_1^2[A] + \beta^2}{\sigma_n^2[A] + \beta^2} \right)^{1/2} \qquad 4.5.3$$

Thus the problem for symmetric matrix argument can be seen to have exactly the square root of the condition number for that of the irreducible quadratic matrix with the same argument.

For arbitrary square matrix A, the condition number for the complex problem is not so simply related to the singular values of the matrix argument, as the following theorem shows.

Theorem 4.5 The complex matrix $A + i\beta I$ for arbitrary square matrix A can possess singular values which exceed the square root of the maximum and minimum eigenvalues of $A^TA + \beta^2 I$.

Proof

Let the maximum and minimum singular values of A^TA be represented by σ_1^2 and σ_n^2. Let z_1 and z_n be the corresponding real maximizing and minimizing vectors. Then,

$$z_1^T(A^TA + \beta^2 I + i\beta(A^T - A))z_1 = z_1^T(A^TA + \beta^2 I)z_1$$

and

$$z_n^T(A^TA + \beta^2 I + i\beta(A^T - A))z_n = z_n^T(A^TA + \beta^2 I)z_n.$$

Since z_1 and z_n are by definition real, a choice of complex vector w may induce greater values in the two expressions (with $.^T$ replaced by $.^*$, the complex conjugate).

For an example of a complex problem for which the conditioning of the matrix formed from the quadratic factor is better than that of the complex problem, let

$$A = \begin{pmatrix} 0 & \alpha \\ 0 & 0 \end{pmatrix} ,$$

then

$$A^2 + I = I$$

but $A + iI$ has roughly the same condition number as that of the real matrix

$$B = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix}$$

i.e.

$$K_2(A + iI) = O(\alpha^2) \qquad \text{for } \alpha >> 1. \qquad \square$$

In practice, such situations are unlikely, since we will not, in general, be working with nilpotent matrices or pure imaginary poles. The complex problem will usually be well-conditioned, relative to the matrix argument, at least.

Thus, the conditioning of the complex problem with respect to the associated real problem is possibly greater for general non-symmetric matrices. However, as the imaginary part of the matrix tends to zero, i.e. for large time step, we have

Theorem 4.6 In the limit as $t \to \infty$ for denominator factors of A-acceptable rational approximations, the maximum and minimum singular values of the complex problem become those of the coefficient matrix A.

Proof The result follows from equation 4.3.2, whose scalar arguments are inverse functions of t. $\square$

4.6 The Cost of Complex Elimination

The cost of performing Gaussian elimination in complex arithmetic may be compared to other alternative methods with respect to the number of real operations required to perform the same elimination steps on a real matrix of equivalent structure. The correspondence between the real and complex arithmetic then establishes the complexity of the algorithm.

The costs of complex operations in terms of equivalent real operations are summarized in Table 4.4 below. In addition to additive, multiplicative, and division costs, the cost of inverting a non-zero complex number is tabulated.

Table 4.4

COMPLEX OPERATIONS IN TERMS OF EQUIVALENT REAL OPERATIONS

Complex \ real	+,-	*	÷
+,-	2	0	0
*	2	4	0
÷	3	3	3
$1/\alpha$	1*	2	2

For real computations in which the number of additive and multiplicative operations is the same, the cost of the equivalent complex computation is four times that of the real computation. The cost of inversion

of a non-zero complex number is included, because the cost of a complex division is not comparable to the cost of a complex multiplication. Even though the real divisions can be seen to be more expensive than real multiplications in Table 4.1, the fact that they are hardware instructions and not performed via a subroutine results in a difference in speed which does not affect the complexity arguments. In the context of Gaussian elimination we have further justification for inverting the diagonal numbers which are normally used in division.

Remark 4.1 When Gaussian elimination is performed on a non-singular irreducible matrix A, the diagonal quantities are used at least twice in an elimination procedure; at least once in forward elimination, and once in the substitution of the right-hand side.

Comparing the table of equivalent real operations to perform the complex operation, we see that an upper bound on the cost of Gaussian elimination on an nxn complex matrix using the given assumption about the diagonal terms may be expressed in terms of the real factorization cost.

$$C_n(G) \leq 4(R_n(G) + n), \qquad 4.6.1$$

where $R_n(G)$, and $C_n(G)$ are the costs of the elimination algorithm in real and complex arithmetic, respectively.

There are a number of circumstances where the cost of the elimination for the quadratic is higher than for the complex elimination.

EXAMPLE 1

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

is a coefficient matrix (with 1's representing non-zero elements or blocks, and 0's representing zero elements or blocks) whose complex elimination algorithm has $O(n)$ computational cost, while the irreducible quadratic factor has $O(n^3)$ elimination cost and $O(n^2)$ substitution cost. In the case of block structure which is oriented towards maintenance of sparsity, this sparsity arrangement (see for example [21]) for the coefficient matrix of the differential equation is inappropriate for the irreducible quadratic factor with that coefficient matrix as argument. The quadratic matrix is, in the absence of any knowledge of the internal block structure, full.

In the absence of any other considerations excepting complexity, the complex elimination for a full matrix is more expensive than the formation of real quadratic factors for the case of linear systems $y' = Ay + f(t)$, for more than one irreducible quadratic factor in the rational function denominator. It is competive in at least one common situation other than the example just given, namely that of systems

with tridiagonal coefficient matrix A. For the principal reason that the initialization of complex elimination involves no matrix multiplication, the complex procedure can at least compete when the matrix squaring results in either increased initialization overhead, increased storage, or as in the example, increased computational complexity. The cost breakdown is given below for the elimination alternatives.

COMPLEX FACTORIZATION IN REAL OPERATIONS

	Real Quadratic		Complex Factorization	
	+ -	* /	+ -	* /
Initial Cost	4n	9n	0(1)	0(1)
Per Factor Cost	8n	9n	3n	8n
Substitution per Factor	7n	10n	10n	12n

where we neglect the 0(1) terms. The cost with respect to the solution of equation 4.3.2 is at least competitive for methods which require a frequent re-evaluation of the coefficient matrix or the time-steps (Case ii in section 4.3). A typical calculation for which two quadratic factors with the same matrix argument are to be used in the form of equation 4.3.2 would require for the real case 34n additive, and 37n multiplicative operations and the complex form would require 26n additive and 36n multiplicative operations. The saving in the complex elimination for tridiagonal matrices is accomplished by the fact that the inversion of the diagonal element is the only full complex multiplicative operation.

4.7 Round-off Errors in the Matrix Computations

The formation of the real irreducible quadratic factors is a source of computational error in a solution procedure which necessitates this step. If we denote the columns of A by c_i, $i = 1,\ldots,n$, and the rows by r_i^T, then

$$(A^2)_{ij} = r_i^T c_j,$$

and, using Theorem 1.5, the computed element a_{ij}^2 of A_{ij}^2 will have error which is bounded by

$$1.01(n + 1)u||r_i||_2||c_i||_2,$$

where u is the unit round-off error in the floating point arithmetic. The effect of this computational error is best illustrated for A positive definite symmetric. For eigenvalue-eigenvector pairs $\langle\lambda_k, z_k\rangle$, $||z_k|| = 1$, $k = 1,\ldots,n$ of the matrix A, A may be written

$$A = \sum_{k=1}^{n} \lambda_k z_k z_k^T . \qquad 4.7.1$$

The jth column of A is then

$$A_{\cdot j} = \sum_{k=1}^{n} \lambda_k \{z_k\}_j z_k$$

and the ith row is

$$A_{i\cdot} = \sum_{k=1}^{n} \lambda_k \{z_k\}_j z_k^T .$$

Hence, the ijth element of A^2 is given by the expression

$$A_{i\cdot}.A_{\cdot j} = \sum_{k=1}^{n} \lambda_k^2 \{z_k\}_i \{z_k\}_j , \qquad 4.7.2$$

and

$$|A_{i}.A._{j}| \leq \sum_{k=1}^{n} \lambda_k^2$$

since A is positive definite. If $\lambda_1 >> \lambda_n$ the component of the computed a_{ij}^2 element corresponding to λ_n may be negigible in equation 4.7.2. This will be more probably a drift in the computed λ_n^2 rather than a change in z_n, especially if λ_n is an isolated small eigenvalue. The λ_n solution component is, however, the persistent component in exponential solutions to

$$y' = -Ay \quad \text{(A positive definite)},$$

and error in the numerical solution is likely to occur if A is severely ill-conditioned.

On the other hand, there is no error in forming the matrix $A + i\beta I$ or its real equivalent:

$$\begin{pmatrix} A & -\beta I \\ \beta I & A \end{pmatrix}.$$

The error occurs in the reduction to complex L-U form.

One source of error in the solution results from the loss of significance in the elimination steps. To minimize this, a pivoting strategy needs to be devised which is inexpensive to use.

Let $A + i\beta = \tilde{M}_0$, 4.7.3

where $\tilde{m}_{ij} = a_{ij} + ib_{ij}$,

and construct the real matrix C corresponding to this complex matrix by replacing the ijth element of A by the 2x2 matrix

$$M_{ij} = \begin{pmatrix} a_{ij} & -b_{ij} \\ b_{ij} & a_{ij} \end{pmatrix}$$

The resulting matrix will look like

$$M_0 = \begin{pmatrix} M_{11} & M_{12} \cdots M_{1n} \\ M_{21} & M_{22} \cdots M_{2n} \\ \vdots & \vdots \\ M_{n1} & M_{n2} \cdots M_{nn} \end{pmatrix} \qquad 4.7.4$$

Now define

$$r_i = (a_{ii}^2 + b_{ii}^2)^{1/2}$$

$$c_i = a_{ii}/r_i$$

$$s_i = b_{ii}/r_i.$$

The matrix

$$G_{ii} = \begin{pmatrix} c_i & -s_i \\ s_i & c_i \end{pmatrix}$$

is an orthogonal matrix and

$$\begin{pmatrix} c_i & -s_i \\ s_i & c_i \end{pmatrix} \begin{pmatrix} r_i & 0 \\ 0 & r_i \end{pmatrix} = \begin{pmatrix} a_{ii} & -b_{ii} \\ b_{ii} & a_{ii} \end{pmatrix} \qquad 4.7.5$$

Now,

$$M_0 = \begin{pmatrix} c_1 & -s_1 & & \\ s_1 & c_1 & 0_{2x2(n-1)} & \\ & & & \\ 0_{2(n-1)x2} & & I_{2n-2x2n-2} & \end{pmatrix} \cdot M_0'$$

$$M_0' = \begin{pmatrix} r_1 & 0 & a_{12}' - b_{12}' & \cdots & a_{1n}' - b_{1n}' \\ 0 & r_1 & b_{12}' & a_{12}' & b_{1n}' & a_{1n}' \\ & & \text{(3rd through 2nth rows of } M_0\text{)} & & & \end{pmatrix}$$

and

$$\begin{pmatrix} a_{1i}' & -b_{1i}' \\ b_{1i}' & a_{1i}' \end{pmatrix} = \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} a_{1i} & -b_{1i} \\ b_{1i} & a_{1i} \end{pmatrix} . \qquad 4.7.6$$

We may now proceed with the first $4(n-1)$ elimination steps of the Gaussian elimination process, noting that only half of them are distinct.

$$M_0' = \begin{pmatrix} 1 & 0 & & \\ & & 0_{2x2(n-1)} & \\ 0 & 1 & & \\ & & & \\ \frac{\sigma_1}{r_1} & \frac{\sigma_2}{r_1} & I_{2n-2x2n-2} & \end{pmatrix} \begin{pmatrix} r_1 & 0 & \rho_1^T & \\ 0 & r_1 & \rho_2^T & \\ & & & \\ 0 & & M_0'' - \frac{\sigma_1\rho_1^T}{r_1} - \frac{\sigma_2\rho_2^T}{r_2} & \end{pmatrix}$$

Also, note that the two eliminations in the ith and i + 1st columns are disjoint, by virtue of the fact that the 2x2 matrix on the diagonal is a diagonal matrix. Hence the elements of ℓ_i and ℓ_{i+1} can be made less than one in absolute value by a partial pivoting strategy that searches for $\max_{i \geq j}\{|a_{ij}||b_{ij}|\}$. Such a strategy gives a slow growth in the upper triangular part of the matrix. Finally, the 2-norm of $a_{1j}' + ib_{1j}'$ is the same as the 2-norm of $a_{1j} + ib_{1j}$. The basic 2x2 matrices in the submatrix

$$M_0'' - \frac{\sigma_1\rho_1^T}{r_1} - \frac{\sigma_2\rho_2^T}{r_1}$$

step are in the i, jth position

$$M''_{ij} = \begin{pmatrix} m_{Rij} & -m_{Iij} \\ m_{Iij} & m_{Rij} \end{pmatrix} \qquad 4.7.7$$

where

$$m_{Rij} = a_{ij} - \frac{1}{r}(a_{i1}a'_{1j} - b_{i1}b'_{1j})$$

$$m_{Iij} = b_{ij} - \frac{1}{r}(b_{i1}a'_{1j} - a_{i1}b'_{1j}).$$

Now, let

$$\eta = ||(a_{i1} + ib_{i1})(a'_{1j} + ib'_{ij})||_2$$

then, since the rotation preserves the 2 norm

$$\eta \leq ||a_{i1} + ib_{i1}||_2||a'_{1j} + ib'_{1j}||_2$$

$$\eta \leq ||a_{i1} + ib_{i1}||_2||a_{1j} + ib_{1j}||_2 \ .$$

The pivoting strategy was designed to operate in the $||\ ||_\infty$ norm over the a_{ij}'s,

Hence

$$\max\{|a_{i1}|, |b_{i1}|\} \leq r,$$

$$||a_{i1} + ib_{i1}||_2^2 \leq 2r^2$$

$$||a_{i1} + ib_{i1}||_2 < \sqrt{2}\ r.$$

Therefore

$$\frac{\eta}{r} \leq \sqrt{2}||a_{1j} + ib_{1j}||_2$$

and

$$||M''_{ij}|| \leq (1 + \sqrt{2})\max\{||a_{1j} + ib_{1j}||_2, ||a_{ij} + ib_{ij}||_2\}.$$

The growth factor in the complex elimination is $(1 + \sqrt{2})^k$ where k is the number of elimination steps, rather than 2^k as in the real case, for an $||\ ||_\infty$ - norm pivoting strategy.

Provided pivoting is applied, where necessary, the inherent stability of the orthogonal transformation which is implicit in the complex division defines a stable procedure ([40], p 133). Experimentally, the complex elimination algorithm for positive definite A and $M = A + i\beta I$, appears to be at least as accurate in general as double precision calculation with the irreducible quadratic factor. This is particularly evident when A is so ill-conditioned that a stable L-U decomposition doesn't exist for $A^2 + \beta^2 I$.

4.8 Linear Systems with Implicit Matrix Factors (II)

For systems with implicit matrix factors, the complex elimination form provides, for a modest increase in cost, an algorithm which doesn't involve M^{-1}.

For $$My' = Ny + f(t) \qquad 4.8.1$$

we are required to solve systems with coefficient matrix $M^{-1}N$ (assuming M is non-singular). The systems to be solved have the general form

$$\{(M^{-1}N)^2 + 2a(M^{-1}N) + (a^2 + b^2)I\}y = b + M^{-1}f. \qquad 4.8.2$$

Unless M and N commute, there is no formulation of equation 4.8.2 which is free of $M^{-1}N$:

$$NM^{-1}N + 2aN + (a^2 + b^2)My = Mb + f \qquad 4.8.3$$

If $NM^{-1} = M^{-1}N$, we have

$$\{N^2 + 2aMN + (a^2 + b^2)M^2\}y = M^2b + Mf \qquad 4.8.3a$$

However, the complex form is M^{-1} - free:

$$\{M^{-1}N + (a + ib)I\}y = b + M^{-1}f \qquad 4.8.4$$

or, equivalently (for M non-singular)

$$\{N + (a + ib)M\}y = Mb + f. \qquad 4.8.4a$$

Equation 4.8.4a requires the additional computation N + aM in the initial decomposition phase. Also, some of the simplification in, for example, tridiagonal systems, is lost.

For sparse matrices the non-zero structure of M + γN relative to that of M and N is additive, and thus provides considerable simplification in maintaining sparsity. For nearly singular matrix argument M, we again have the numerical effect of solving the reduced system for large argument a(t) + ib(t).

4.9 Conclusions

The complex factorization form of implementing rational approximations for matrix argument has no one-to-one correspondence in cost with its irreducible quadratic real form. For sparse coefficient matrix argument, and especially for implicit matrix argument, where structural considerations greatly affect the real computational mode, there is a clear advantage in using the complex factorization. The special form of equation 4.3.2 dictates that, even in the case of a dense coefficient matrix, the substitution costs differ by at most a factor of two, and initialization costs between a factor of one and four. We can conclude that, carefully applied, the complex factorization is a serious competitor to the real quadratic implementations of the evaluation schemes.

CHAPTER 5

SPECIFIC IMPLEMENTATIONS

5.1 Details of Algorithm Construction

The Padé approximations of order ≥ 2 (excepting the trapezoidal approximation) and many of the Chebyshev rational approximations with or without order at t=0 have denominator factors which are almost always irreducible quadratics. Methods such as the complex elimination procedure of Theorem 4.4 are applicable to these approximations. In this chapter we implement in the complex form three of the Padé approximations, the Padé (2,0), the Padé (2,1) and the Padé (2,2). These approximations are particularly appropriate examples of the complex implementation because there is a single irreducible quadratic denominator factor. The close relationship with the optimal IRK methods provides useful sets of quadrature points on which to develop quadrature formulae.

Algorithm Construction

The procedure for deriving an algorithm for complex arithmetic is given in four steps.

1. For indeterminate argument z, the rational approximation and its moments from equations 2.2.3 are derived.

2. For particular abscissae on [0,1] the weights are generated from the appropriate VanderMonde system.

3. Theorem 4.3 is applied to the system to determine the complex constants which result in the solution to the homogeneous problem.

4. Complex constants $\alpha + i\beta$ are determined for each weight $W_k(z)$ to satisfy the linear system associated with the forced problem in one solution step.

5.2 The Padé (2,0) Approximation

The Padé (2,0) approximation is a second order L-acceptable approximation to exp(z). It has the form

$$P_{20}(z) = (1 - z + \frac{z^2}{2})^{-1} \qquad 5.2.1$$

The first neglected term in the error expansion is, from Chapter 3, $z^3/6$. The moments of order 0 and 1 are,

$$M_0(z) = (1 - z + \frac{z^2}{2})^{-1}(1 - \frac{z}{2})$$
$$M_1(z) = (1 - z + \frac{z^2}{2})^{-1}(\frac{1}{2} - \frac{z}{2}). \qquad 5.2.2$$

On the unit interval, we choose abscissae 0,1. The resulting weight functions are

$$W_0(z) = (1 - z + \frac{z^2}{2})^{-1}$$
$$W_1(z) = (1 - z + \frac{z^2}{2})^{-1}(\frac{1}{2} - \frac{z}{2}) \qquad 5.2.3$$

and the algorithm for matrix argument hA is

$$(I - hA + \frac{h^2A^2}{2})y_{n+1} = y_n + \frac{h}{2} f_n + h(\frac{1}{2} - \frac{hA}{2}) f_{n+1} \qquad 5.2.4$$

The irreducible quadratic factor may be factored into

$$(1 - z + \frac{z^2}{2}) = (1 - (\frac{1}{2} + \frac{i}{2})z)(1 - (\frac{1}{2} - \frac{i}{2})z)$$

and for the homogeneous problem, we have the form

$$(I - (\frac{1}{2} + \frac{i}{2})hA)y^* = y_n$$

$$y_{n+1} = y_R^* + y_I^* \qquad 5.2.5$$

This can be verified by the multiplication of the first equation in 5.2.5 by the complex conjugate factor.

For the inhomogeneous terms, the complex constants in step 4 of the algorithm construction are

$$\alpha_0 + i\beta_0 = \frac{1}{2}$$

$$\alpha_1 + i\beta_1 = \frac{i}{2}$$

The resulting inhomogeneous complex algorithm is then

$$(I - (\frac{1}{2} + \frac{i}{2})hA)y^* = y_n + \frac{h}{2}(f_n + if_{n+1}) \qquad 5.2.6$$

The coefficient matrix of this form may be put in the form $M + i\beta I$ by multiplication on the left by $1 - i$:

$$(I - hA - iI)y^* = (1 - i)y_n + h(\frac{1 - i}{2})f_n - h(\frac{1 - i}{2})f_{n+1}$$

The equivalent real problem is written as

$$\begin{pmatrix} I - hA & I \\ -I & I - hA \end{pmatrix}\begin{pmatrix} y_R^* \\ y_I^* \end{pmatrix} = \begin{pmatrix} y_0 \\ -y_0 \end{pmatrix} + \frac{h}{2}\begin{pmatrix} f_n - f_{n+1} \\ f_n + f_{n+1} \end{pmatrix} \qquad 5.2.7$$

We observe that the pre- and post-multiplication of the coefficient matrix by

$$Q = \begin{pmatrix} I_{nxn} & I_{nxn} \\ I_{nxn} & -I_{nxn} \end{pmatrix}$$

and Q results in the system

$$[QMQ^{-1}]Qy^{*} = Qb$$

i.e.,

$$\begin{pmatrix} I - hA & -I \\ I & I - hA \end{pmatrix} \begin{pmatrix} y_R^{*} + y_I^{*} \\ y_R^{*} - y_I^{*} \end{pmatrix} = \begin{pmatrix} 0 \\ 2y_n \end{pmatrix} + h \begin{pmatrix} f_1 \\ f_0 \end{pmatrix} \qquad 5.2.8$$

Setting

$$y_R^{*} + y_I^{*} = y_R^{**}$$

$$y_R^{*} - y_I^{*} = y_I^{**}$$

then $y_{n+1} = y_R^{**}$, and 5.2.8 is in the computational form given by Theorem 4.3.

Returning to the system in complex form, we have

$$(I - hA + iI)y^{**} = 2iy_n + hf_{n+1} + ihf_n \qquad 5.2.9$$

or, equivalently,

$$((I - hA)^2 + I)y^{**} = 2y_n + 2i(I - hA)y_n +$$

$$+ [\{(I - hA)f_{n+1} + f_n\} + i\{-f_{n+1} + (I - hA)f_n\}]. \qquad 5.2.10$$

The imaginary part of the solution vector in equation 5.2.9 is the solution by the Padé (2,0) approximation to a system with the same coefficient matrix A, with initial conditions $\tilde{y}_{n-1}$ given implicitly by the equation

$$\tilde{y}_{n-1} = (I - hA)y_n$$

and the weights for the solution to the inhomogeneous problem are {1,2},

so that this extra solution may be considered to be on the previous time-interval. It is possible that methods may develop from this observation about the Padé (2,2) approximation which operate in predictor-corrector mode for non-linear systems.

The transformation back to the original problem definition in complex terms yields the observation that the real part of the solution to equation 5.2.6 satisfies the boundary conditions, and the imaginary part of this solution represents a solution with homogenious boundary conditions.

For semi-discretized parabolic partial differential equations, an attempt to split the coefficient matrix in higher space dimensions may be able to make use of this observation to facilitate the problem of local matching of time-dependent boundary conditions when using this approximation.

5.3 The Padé (2,1) Approximation

The Padé (2,1) approximation is a third order approximation to exp(z), which is both A-acceptable and L-acceptable. It takes the form

$$P_{21}(z) = (1 - \frac{2}{3} z + \frac{z^2}{6})^{-1}(1 + \frac{1}{3} z) \qquad 5.3.1$$

Denoting the denominator by

$$q(z) = (1 - \frac{2}{3} z + \frac{z^2}{6}),$$

its first three moments are

$$q(z)M_0(z) = (1 - \frac{z}{6}),$$

$$q(z)M_1(z) = (\frac{1}{2} - \frac{z}{6}),$$

and

$$q(z)M_2(z) = (\frac{1}{3} - \frac{z}{6}). \qquad 5.3.2$$

The denominator factors into

$$(1 - \frac{2}{3}z + \frac{z^2}{6}) = (1 - (\frac{1}{3} + \frac{i}{3\sqrt{2}})z)\ (1 - (\frac{1}{3} - \frac{i}{3\sqrt{2}})z)$$

and for the homogeneous solution the complex factorization procedure can be written for $y^* = y_R + y_I$

$$(I - (\frac{1}{3} + \frac{i}{3\sqrt{2}})hA)y^* = y_n$$

$$y_{n+1} = y_R + 2\sqrt{2}\, y_I \qquad 5.3.3$$

There are a number of choices of quadrature points which yield implicit Runge-Kutta formulations for this approximation but, for linear systems, one which yields a minimum of function evaluations for the inhomogeneous terms is the set {1/3, 1}, i.e., the implicit Radau formulae. For the scalar Radau quadrature formula, the weights for these abscissae are 3/4 and 1/4.

Solving for the numerators of the moments, for the abscissae, 1/3, α, and 1, $\alpha \neq 1/3$ or 1, we obtain $W_\alpha \equiv 0$ and

$$q(z)\ W_{1/3} = \frac{3}{4}$$

$$q(z)W_1 = (\frac{1}{4} - \frac{z}{6}). \qquad 5.3.4$$

In terms of the complex factorization, the solution to the linear inhomogeneous problem with this quadrature is

$$(I - (\frac{1}{3} + \frac{i}{3\sqrt{2}})hA)y^* = y_n + \frac{h}{3}(\frac{5}{4} f_{n+1/3} - \frac{1}{4} f_{n+1}) +$$

$$+ i \frac{h}{3\sqrt{2}}(f_{n+1/3} + f_{n+1})/2 \qquad 5.3.5$$

$$y_{n+1} = y_R^* + 2\sqrt{2}\, y_I^*$$

which simplifies considerably if written in the form

$$(2I - hA - \sqrt{2}\, iI)y^* = (2 - \sqrt{2}\, i)y_n + hf_{n+1/3} +$$

$$+ i \frac{h}{2\sqrt{2}}(f_{n+1} - f_{n+1/3}) \qquad 5.3.6$$

This formula may be transformed, in a similar fashion to the Padé (2,0), into an implicit Runge-Kutta form, via the transformation

$$QMQ^{-1}(Qy) = Qb,$$

where

$$Q = \begin{pmatrix} I & 2\sqrt{2}I \\ I & 0 \end{pmatrix}$$

to give

$$\begin{pmatrix} \frac{5}{2} I - hA & \frac{9}{2} I \\ -\frac{1}{2} I & \frac{3}{2} - hA \end{pmatrix} \begin{pmatrix} y_{n+1} \\ \tilde{y}_{n+1/3} \end{pmatrix} = \begin{pmatrix} -2y_n \\ 2y_n \end{pmatrix} + h \begin{pmatrix} f_{n+1} \\ f_{n+1/3} \end{pmatrix}$$

The system in this form is similar to one of the variants of the IRK implementations in [10] based on the Padé (2,1) approximation.

The main advantage of this procedure is that, for constant coefficient forced linear systems, there is one less function evaluation than normally necessary for the exactness property for quadratic polynomial forcing terms. A look at the weights for the {0, 2/3, 1}

abscissae,

$$q(z)W_0 = \frac{1}{4}$$

$$q(z)W_{2/3} = \frac{3}{4}$$

$$q(z)W_1 = -\frac{z}{6}$$

establishes that, in general, the number of weights necessary to exactly integrate polynomial forcing of degree k is k + 1 in general, and the example of the {1/3, 1} abscissae is not indicative of the general case.

5.4 The Padé (2,2) Approximation

The Padé (2,2) approximation appears to provide the highest order formula based on rational approximations from the Padé table which is both A-acceptable and for which a reduced set of quadrature points exists which exactly integrate the maximum degree polynomial forcing using the algorithm construction of section 5.1. This approximation is not L-acceptable, since

$$\lim_{\mathrm{Re}[z]\to -\infty} P_{22}(z) = 1.$$

The approximation and its first four moments are:

$$P_{22}(z) = (1 - \frac{z}{2} + \frac{z^2}{12})^{-1}(1 + \frac{z}{2} + \frac{z^2}{12}),$$

$$M_0(z) = (1 - \frac{z}{2} + \frac{z^2}{12})^{-1}, \qquad 5.4.1$$

$$M_1(z) = M_0(z)(\frac{1}{2} - \frac{z}{12}),$$

$$M_2(z) = M_0(z)(\frac{1}{3} - \frac{z}{12}),$$

$$M_3(z) = M_0(z)(\frac{1}{4} - \frac{z}{12}) \qquad 5.4.1$$

For the Gauss-Legendre points $\{\frac{3 - \sqrt{3}}{6}, \frac{3 + \sqrt{3}}{6}\}$ on [0,1], the weights for the matrix quadrature are:

$$W_{g_1}(z) = M_0(z)(\frac{1}{2} + \frac{\sqrt{3}}{12} z)$$

$$W_{g_2}(z) = M_0(z)(\frac{1}{2} - \frac{\sqrt{3}}{12} z) \qquad 5.4.2$$

When applied to the scalar equation $y' = qy + (mt^{m-1} - t^m q)b$ on [0,h] for m = 0,1,2,3, we find that this quadrature formula integrates the polynomial solution exactly only to polynomial degree 2, and for m = 3,

$$y(h) = (1 - q\frac{h}{2} + \frac{12}{144} q^2h^2)^{-1}(1 - q\frac{h}{2} + \frac{10}{144} q^2h^2) \cdot h^3 b.$$

i.e. the solution has error

$$E(h) = (1 - \frac{qh}{2} + \frac{q^2h^2}{12})^{-1} \frac{1}{72} q^2h^5 b$$

giving insight into the difference between order and exactness for linear systems.

One approach to gaining full polynomial degree for the quadrature is to add the points {0,1} to the Gaussian points, to produce the weights

$$W_0(z) = M_0(z)(-\frac{z}{12})$$

$$W_{g_1}(z) = M_0(z)/2$$

$$W_{g_2}(z) = M_0(z)/2$$

$$W_1(z) = M_0(z)(\frac{z}{12})$$

which are of a particularly simple form.

However the Simpson nodes produce an algorithm which is exact for cubic polynomial forcing terms.

$$W_0(z) = M_0(z)(\frac{1}{6} + \frac{z}{12})$$

$$W_{1/2}(z) = M_0(z)(\frac{2}{3}) \qquad 5.4.3$$

$$W_1(z) = M_0(z)(\frac{1}{6} - \frac{z}{12}).$$

The proof of this assertion is by substitution to verify that

$$M_3(z) = \frac{1}{8}\ W_{1/2}(z) + W_1(z).$$

Thus, one function evaluation is saved and the quadrature is exact to full capability for inhomogeneous terms. In the absence of storage restrictions the end-point function values may be stored in an array for re-use in the next step, resulting in a long-term reduction in the number of function evaluations and for more general non-linear systems a formula which is comparable to the IRK formula in the number of substitutions.

The complex form which reduces to the solution of the real problem requires an extra step, as in the Padé (1,1). The real irreducible quadratic formula for a homogeneous system has the form:

$$(I - \frac{hA}{2} + \frac{h^2A^2}{12})\bar{y} = hAy_n$$

$$y_{n+1} = y_n + \bar{y} \qquad 5.4.4$$

To develop the complex quadratures for the two algorithms of this section, we apply the construct of Section 5.1 to obtain for the

Gaussian nodes

$$(I - (\frac{1}{4} + \frac{i}{4\sqrt{3}})hA)y^* = y_n + h\{(\frac{\sqrt{3}+2}{8\sqrt{3}} + \frac{i}{8\sqrt{3}})f_{g_1} + (\frac{\sqrt{3}-2}{8\sqrt{3}} + \frac{i}{8\sqrt{3}})f_{g_2}\} \quad 5.4.5$$

$$y_{n+1} = y_R^* + 4\sqrt{3}\, y_I^*$$

and for the Simpson nodes

$$(I - \frac{hA}{2} + \frac{h^2A^2}{12})y^{**} = y_n + h\{(\frac{1}{8} + \frac{i}{24\sqrt{3}})f_n + (\frac{1}{6} + \frac{i}{6\sqrt{3}})f_{n+1/2} + (-\frac{1}{24} + \frac{i}{24\sqrt{3}})f_{n+1}\} \quad 5.4.6$$

$$y_{n+1} = y_n + 4\sqrt{3}\, y^{**}$$

The algorithm constructed from the Simpson nodes represents a generalization of a modified multistep method [5], for which the order 2k formulae are known to be stable. As a result, the A-stability and exactness property of this generalized method appear attractive for nonlinear problems. However, the generalization doesn't appear to follow through for the Padé (3,3) approximation, which, for reference, appears below, along with its moments.

$$P_{33}(z) = (1 - \frac{z}{2} + \frac{z^2}{10} - \frac{z^3}{120})^{-1}(1 + \frac{z}{2} + \frac{z^2}{10} + \frac{z^3}{120})$$

$$q(z)M_0(z) = 1 + \frac{z^2}{60}$$

$$q(z)M_1(z) = \frac{1}{2} - \frac{z}{12} + \frac{z^2}{120}$$

$$q(z)M_2(z) = \frac{1}{3} - \frac{z}{12} + \frac{z^2}{120}$$

$$q(z)M_3(z) = \frac{1}{4} - \frac{3z}{40} + \frac{z^2}{120}$$

$$q(z)M_4(z) = \frac{1}{5} - \frac{z}{15} + \frac{z^2}{120} \qquad 5.4.7$$

$$q(z)M_5(z) = \frac{1}{6} - \frac{7z}{120} + \frac{z^2}{120}$$

For the Padé (3,3) approximation, neither the Gaussian nodes nor the modified multistep methods appear to have particularly useful quadratures associated with exactness for the particular integral .

It would appear that one should use equally spaced nodes for the higher order methods, either as one-step or multi-step formulae for linear problems.

CHAPTER 6

CONCLUSIONS AND NUMERICAL RESULTS

6.1 The Decomposition of Rational Functions

The computational scheme for an arbitrary rational function approximation $E_{mn}(z) = P_n(z)/Q_m(z)$ to $\exp(z)$ with order k requires the evaluation of E_{mn} and up to k matrix-valued weight functions. These rational functions all share the denominator $Q_m(z)$. The only efficient mode of computation which involves a single substitution for each denominator factor is a partial fraction decomposition of the rational functions, since for stability we do not want to evaluate the full numerator and polynomials, and for sparse systems we wish to avoid fill-in in the denominator terms.

Accordingly, the rational functions are decomposed into partial fractions:

$$\frac{P_n(z)}{Q_m(z)} = \alpha + \frac{R_k(z)}{Q_m(z)}$$

where $k \leq \min\{n, m-1\}$,

$$\frac{P_n(z)}{Q_m(z)} = \alpha + \sum_{i=1}^{k_m} \frac{c_i(z)}{q_i(z)}$$

where for distinct linear factors

$$c_i(z) = c_i$$
$$q_i(z) = z + d_i \qquad 6.1.1$$

and for repeated linear factors

$$c(z) = c_{ik} \quad k = 1,\ldots,m_i \text{ (the multiplicity of the factor)},$$
$$q_i(z) = (z + d_i)^k \qquad 6.1.2$$

and for irreducible quadratic factors

$$c_i(z) = e_i z + f_i$$
$$q_i(z) = z^2 + g_i z + h_i. \qquad 6.1.3$$

Then vector solutions for the partial fraction corresponding to each denominator factor may be obtained for each weight and E(z) itself, the substitution can be completed, and the result accumulated in a solution vector.

The work is thus proportional to the degree of the denominator and except for function evaluation overhead is, to a first approximation, independent of the number of weight functions carried in the computation.

6.2 Linear Systems with Implicit Matrix Factors (III)

For stable systems whose spectrum is real the combined order-uniform approximations derived by Lawson [30] are approximations E(z) to $\exp(\frac{-z}{1-z})$. The conversion to $\exp(-x)$ for the purpose of computation is accomplished by the transformation

$$z = \frac{x}{x-1} \quad . \qquad 6.2.1$$

The factors, for matrix argument may be substituted directly as for implicit matrix argument (Section 4.8) for if

$$p(z) = z + (a + ib)$$
$$p^*(x) = (\frac{x}{x-1}) + (a + ib)$$

For matrix argument hA, equation 4.8.4a becomes

$$\{hA + (a + ib)(hA - I)\}y = (hA - I)b + f \qquad 6.2.2$$

There is no simple correspondence between the implicit matrix

factors in (I) and (II) and this case, in terms of the original differential equation, but the restriction of the original approximation to the interval [0,1] results in more uniform scalar coefficients with less chance for error due to cancellation than if the interval is transformed to $[0,\infty)$.

6.3 Repeated Exponential Approximations

Given the simple exponential approximation E(z) for which a satisfactory error bound on $[0,\infty)$ exists, the approximation $E^2(z/2)$ may have an improved error bound on $[0,\infty)$. In particular, the A-acceptable subdiagonal Padé approximations, L21, and the averaged Padé approximation

FIGURE 6.1

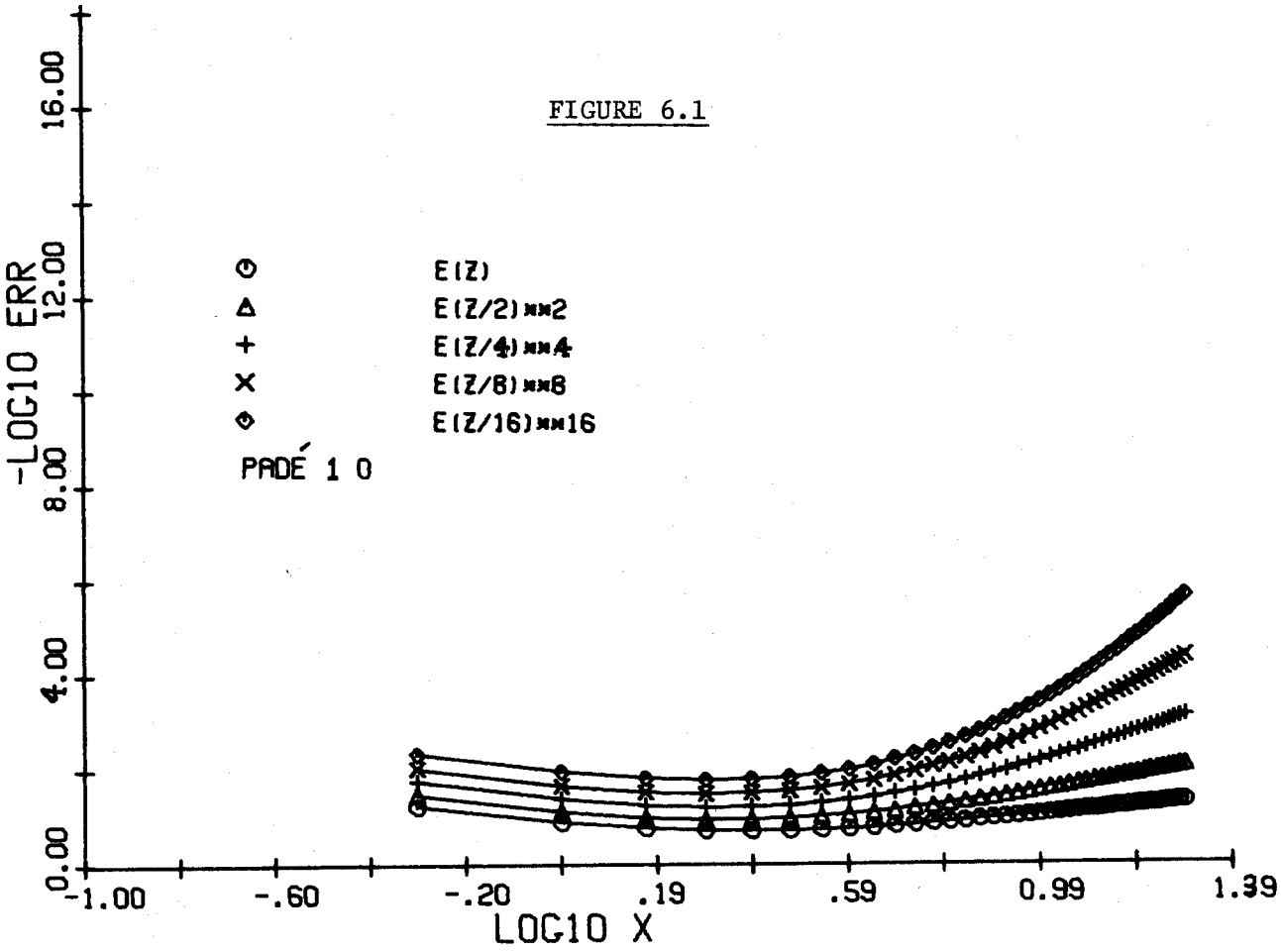
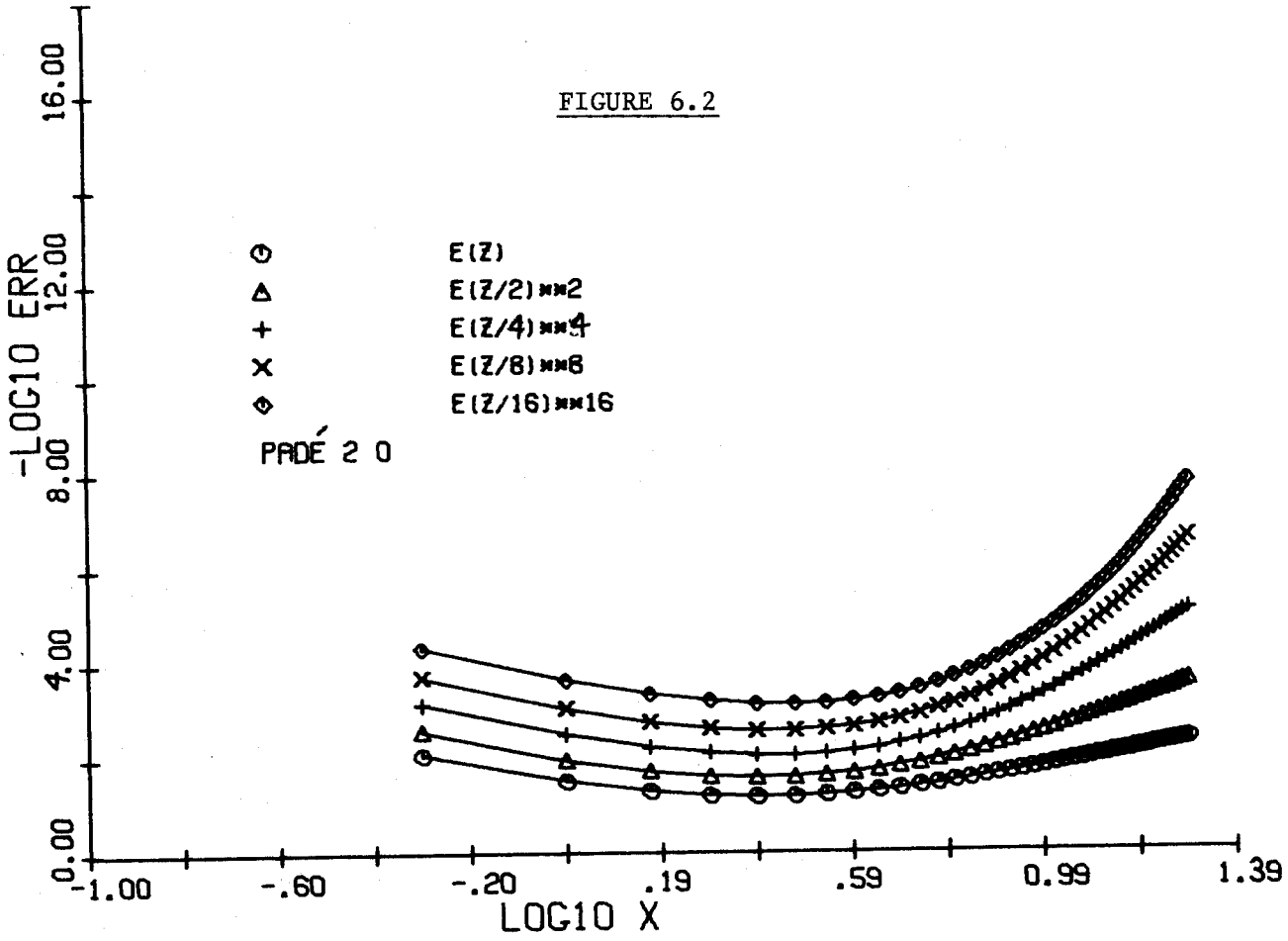


FIGURE 6.2



$A_{11}(x)$ (Section 3.5) all have the property that, for $x >> 1$, the uniform error in the squared approximation, $e_2(x)$, is roughly the square of the error in the original approximation.

Let

$$e_1(x) = R(x) - \exp(-x),$$

then for

$$e_2(x) = R(x/2)^2 - \exp(-x)$$

i.e.

$$e_2(x) = e_1^2(x/2) + 2\exp(-x/2)e_1(x/2)$$

which for large x, implies

$$e_2(x) = e_1^2(x/2).$$

FIGURE 6.3

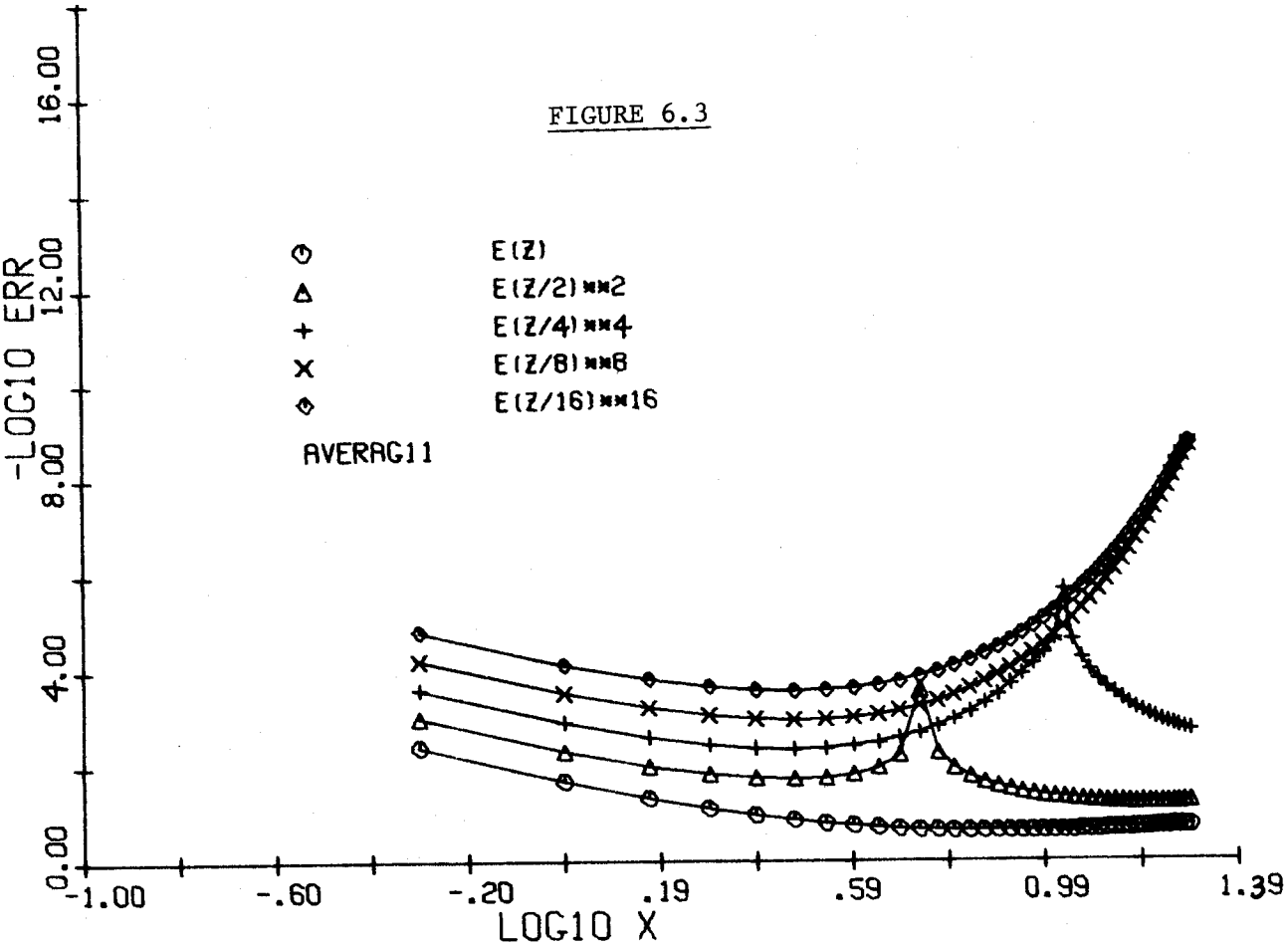


FIGURE 6.4

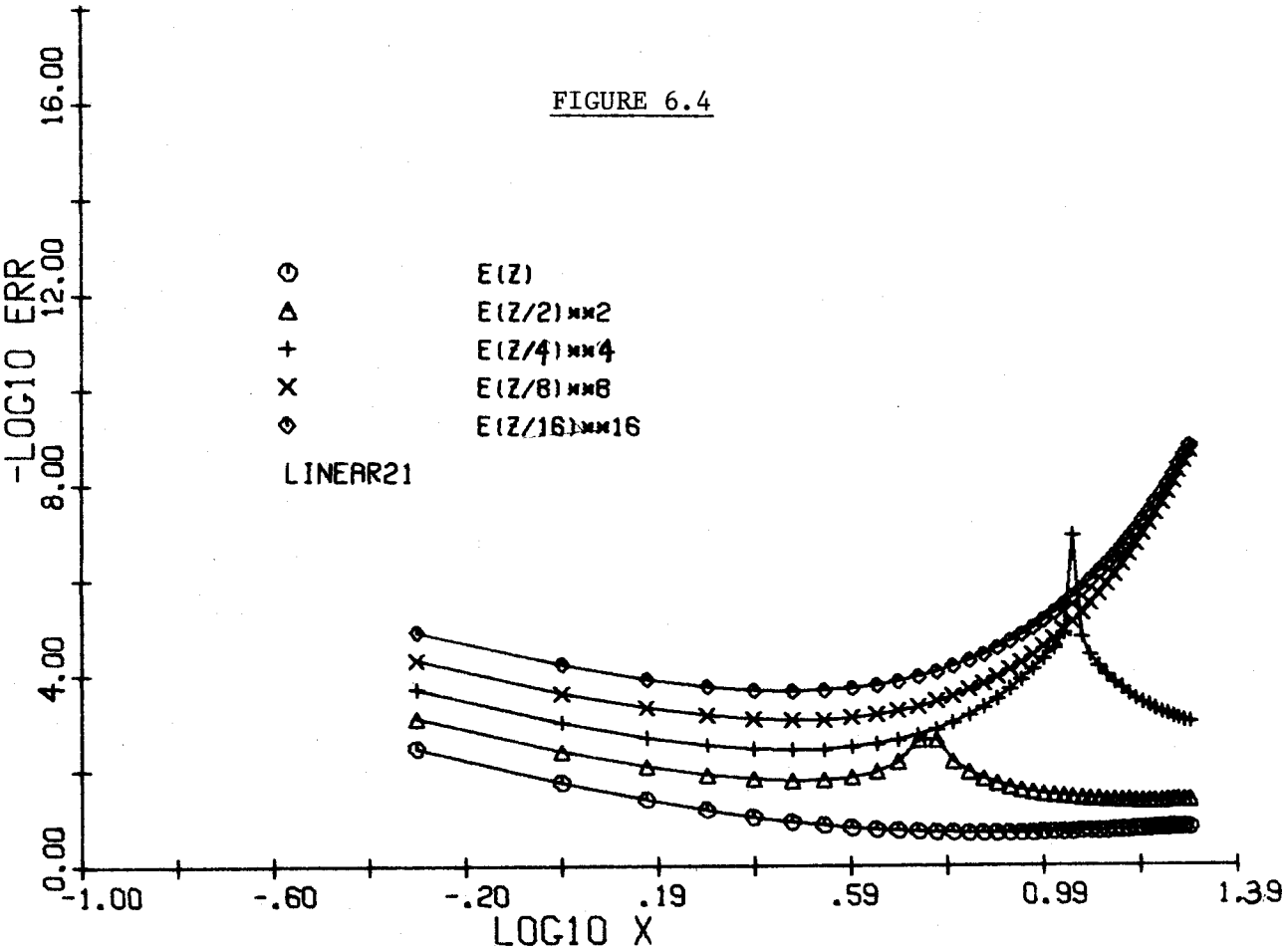
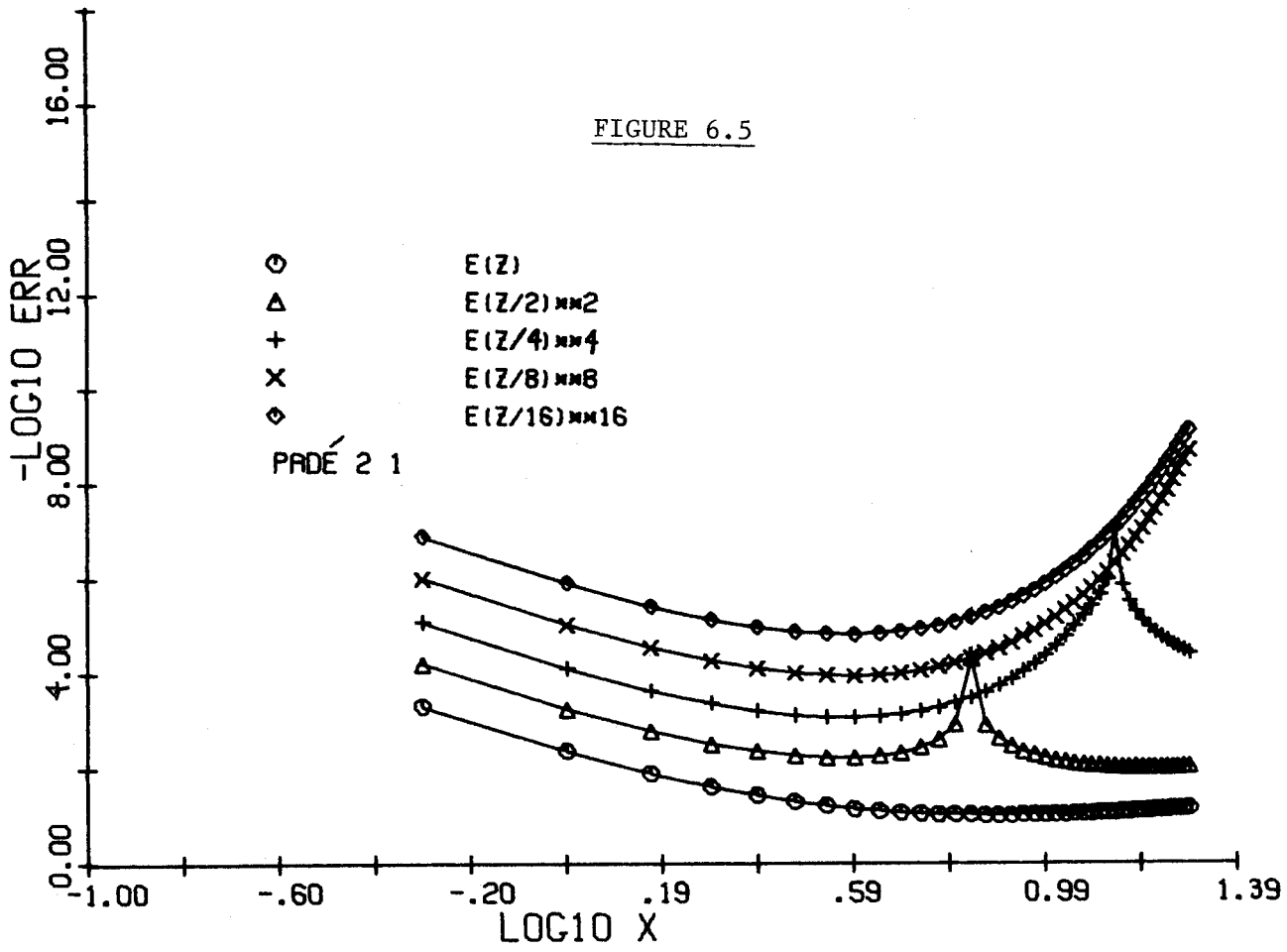


FIGURE 6.5



The graphs labelled Figures 6.1 - 6.5 illustrate this behavior. The order of the approximations in the neighborhood of $x = 0$ provide damping of the error for $x \in [0, 1.5]$ (approximately) where the coefficient c of the error near $z = 0$, $e(z) \doteq c\, z^k$, $k \geq 2$ guarantees a decrease in error $(\frac{1}{2})^{k-1}$ per squaring step. The non-L-acceptable approximations exhibit the same behavior on a finite t-interval, although for stiff systems, the restriction in the time-steps necessary to keep the spectrum of the matrix problem in this time-interval may be unacceptable. (The cusps in the graphs are associated with cross-over points, where the error vanishes.) The close agreement of the E_{11} extrapolated approximation to the

Padé (2,2) is evident for stable systems whose coefficient matrix has a real spectrum (Figs. 6.6-6.8).

FIGURE 6.6

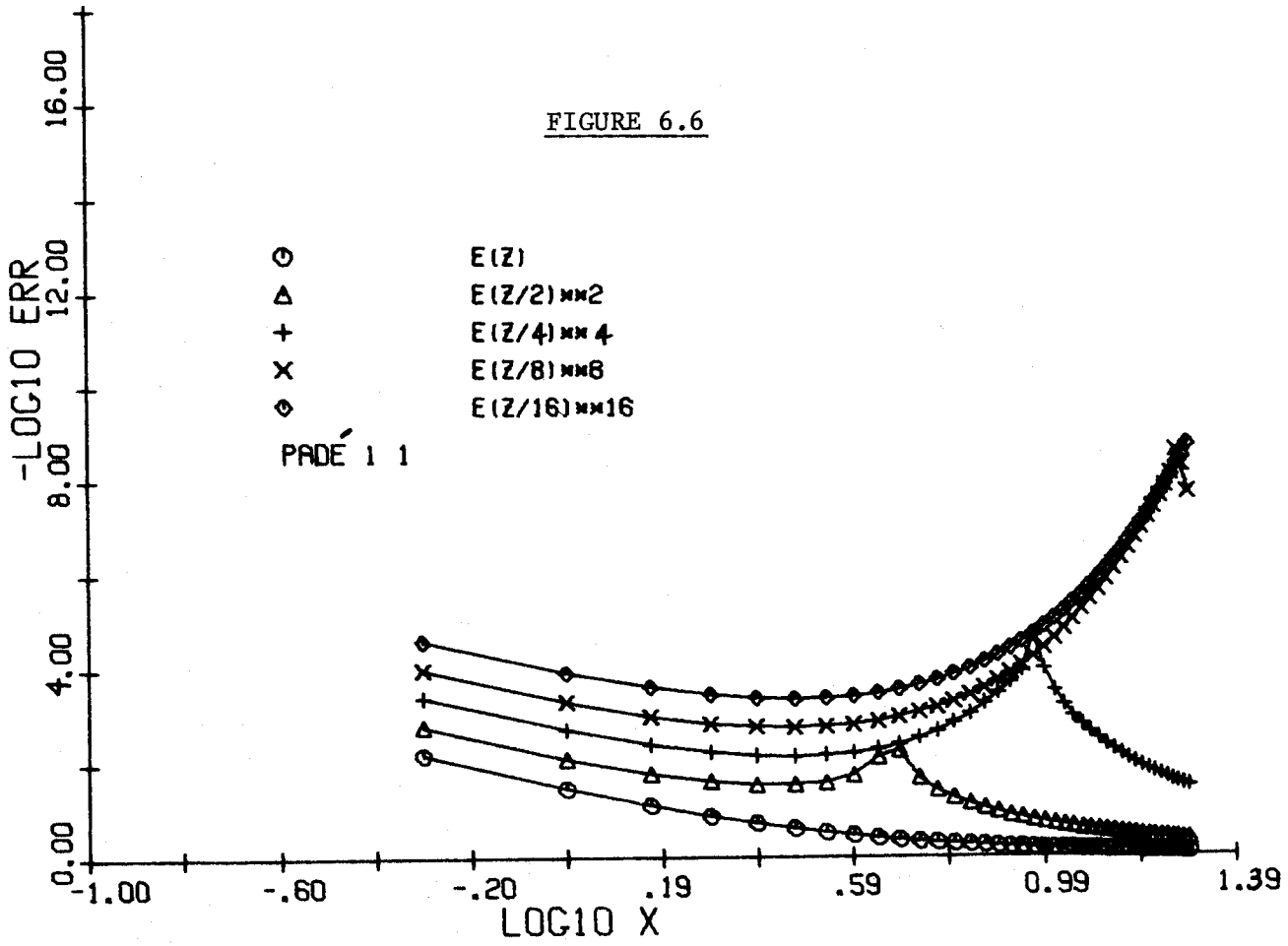


FIGURE 6.7

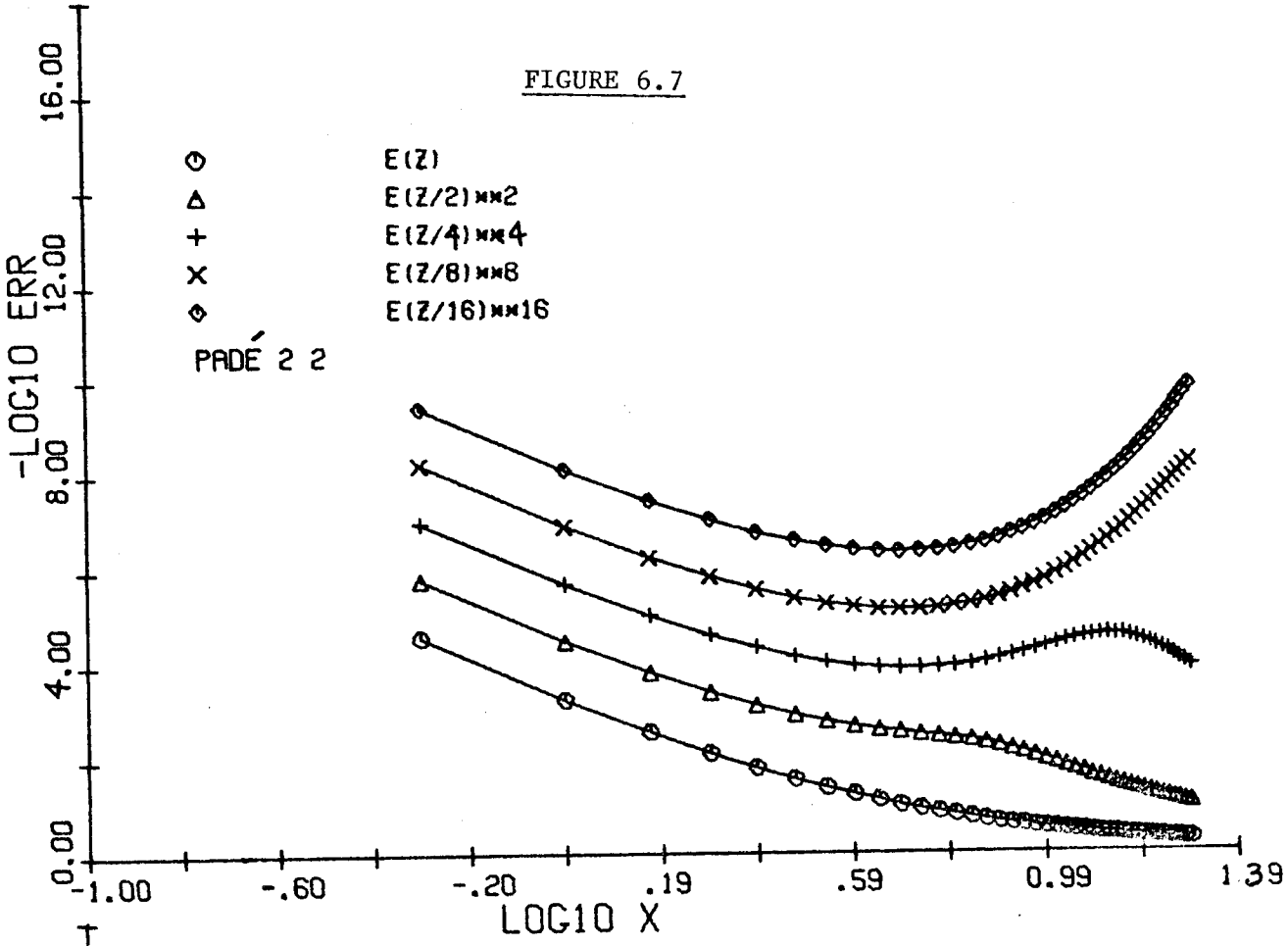
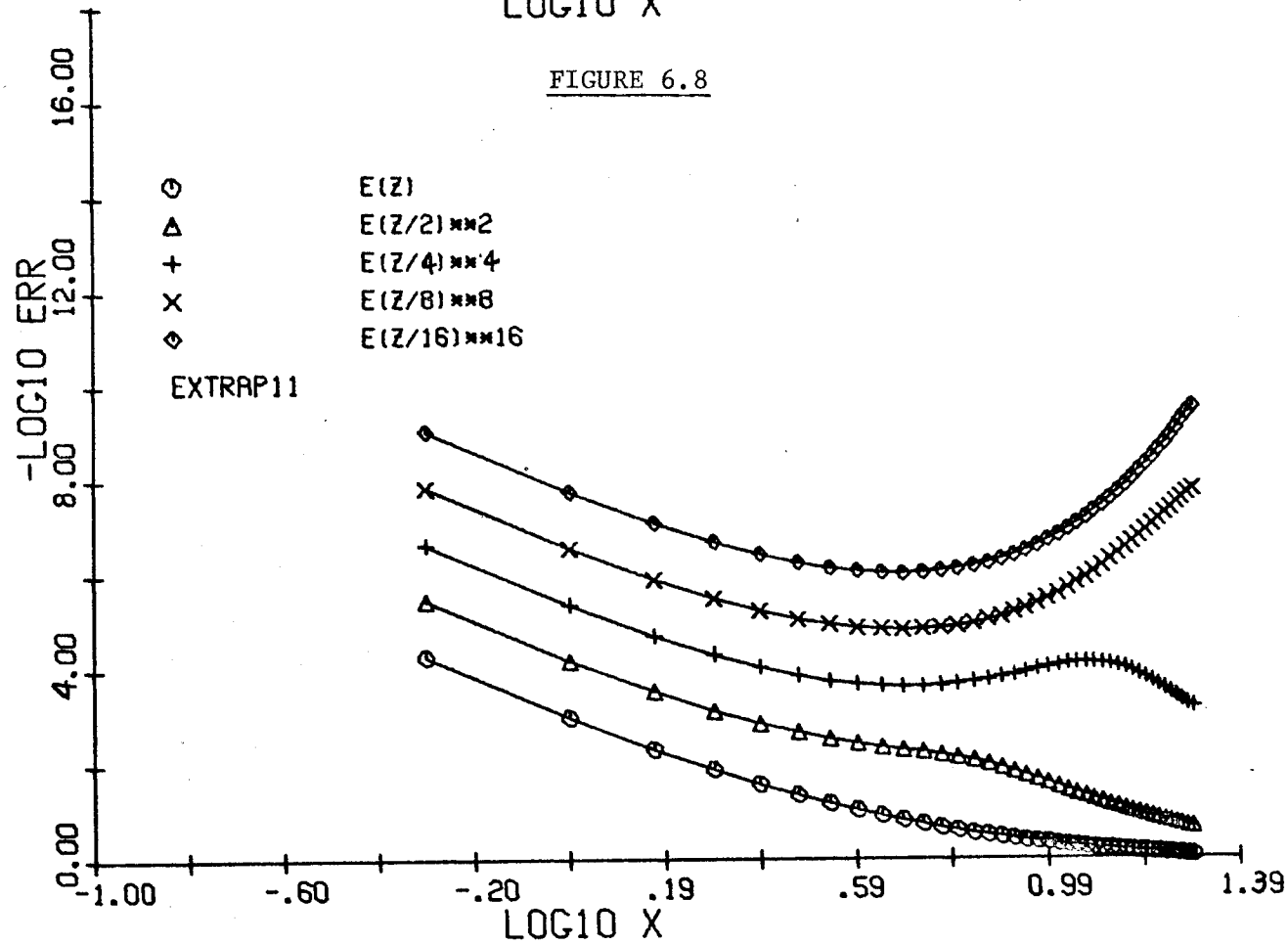


FIGURE 6.8



6.4 Conclusions and Numerical Results

When solving initial-value systems, an increase in stability of a method which results from L-acceptability is more beneficial to the computation than an increase in the order of the method. Implementation of L-acceptable methods which have some computational considerations in their derivation, such as L21, can result in efficient algorithms at modest cost. Whether there exist significantly higher order linear approximations (unlikely), or whether it would be desirable to consider the implementation of linear approximations with greater uniformity, are questions yet to be answered.

The approximations derived without computational considerations may clearly be implemented at reasonable cost with the use of complex elimination. Further investigation of its stability is another major topic to be analysed.

Structural simplifications with respect to the choice of quadrature points (such as in the methods based on L21 and in Chapter five) may exist for a few other methods of modest order. For approximations for which no simplifications exist, a "good" choice of quadrature points must be made. In this respect, one of the negative results of Chapter five is that for nonlinear systems, the Padé approximations do not have stable quadratures to full order.

The following test problems are designed to illustrate two main points which occur in the thesis: the damping properties of L-acceptable methods in the presence of transient solution components; and the ability of such methods to deal with discontinuities in initial-boundary values for the heat equation.

The equation to be solved for t in [0,1] is

$$u_t = u_{xx}, \quad x \in [0,1],$$

with

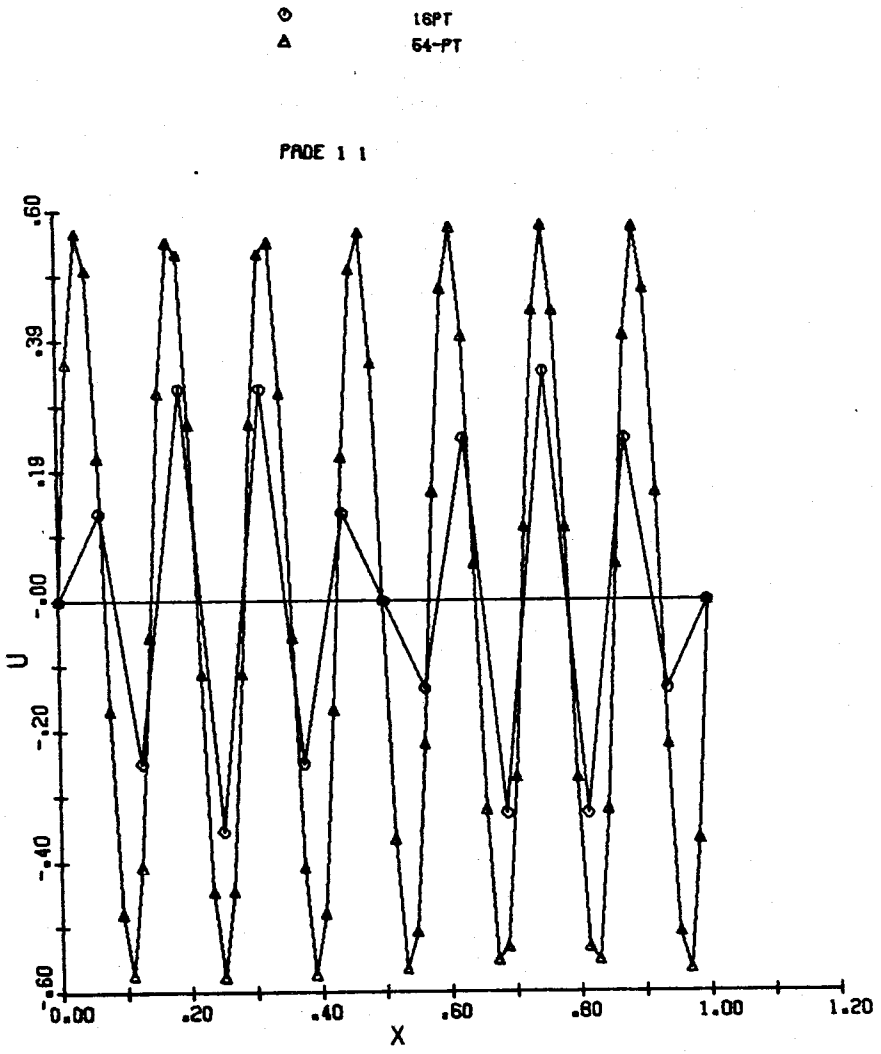
i) $u(x,0) = \sin\pi x + \sin 14\pi x$

$u(0,t) = u(1,t) \equiv 0,$

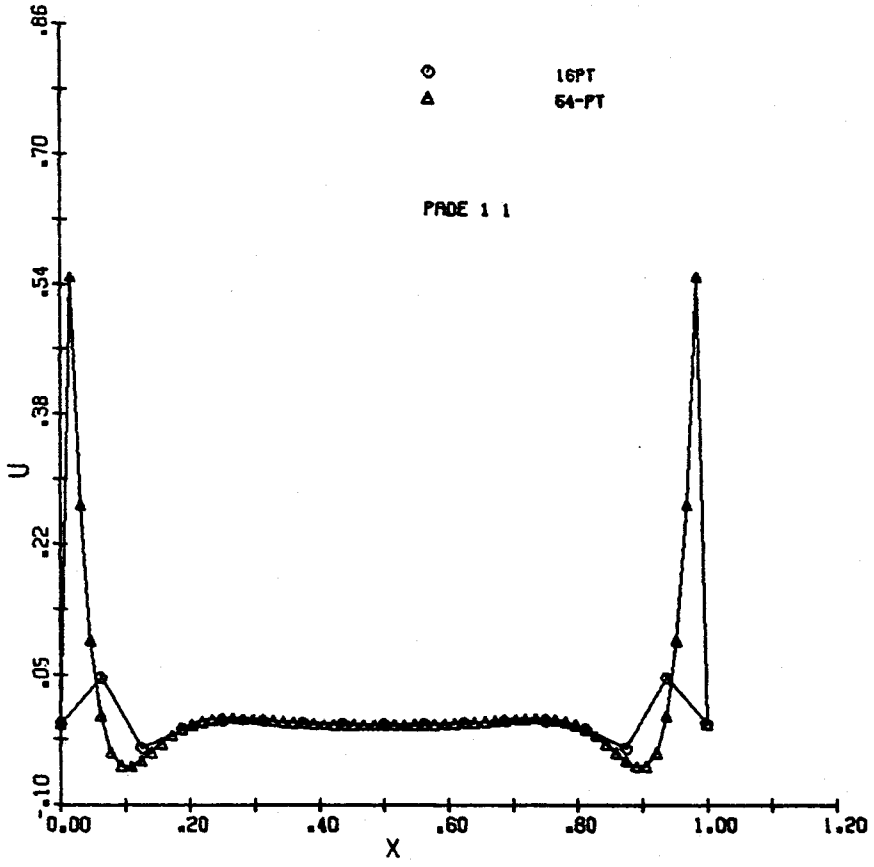
ii) $u(x,0) \equiv 1$

$u(0,t) = u(1,t) \equiv 0,$

using the customary 3-point central difference approximation to the second partial derivative with respect to x. Figures 6.9 - 6.12 are graphs of the finite difference solutions to i and ii respectively at t = 1 for $\Delta x = 0.0625$ and $\Delta x = 0.015625$ for the trapezoidal rule, L21, Padé(2,0), and Padé(2,1) - based algorithms ($\Delta t = 0.0625$ in both cases). The Padé(2,0) and Padé(2,1) algorithms are implemented in simulated complex arithmetic.

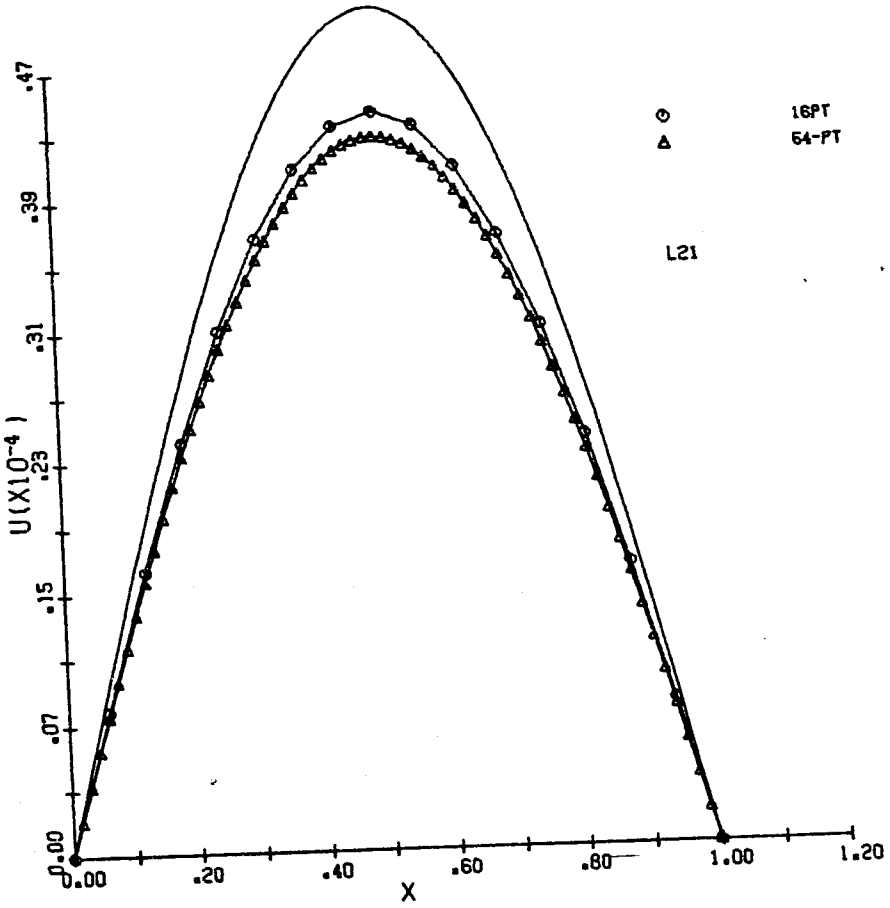


EXAMPLE I

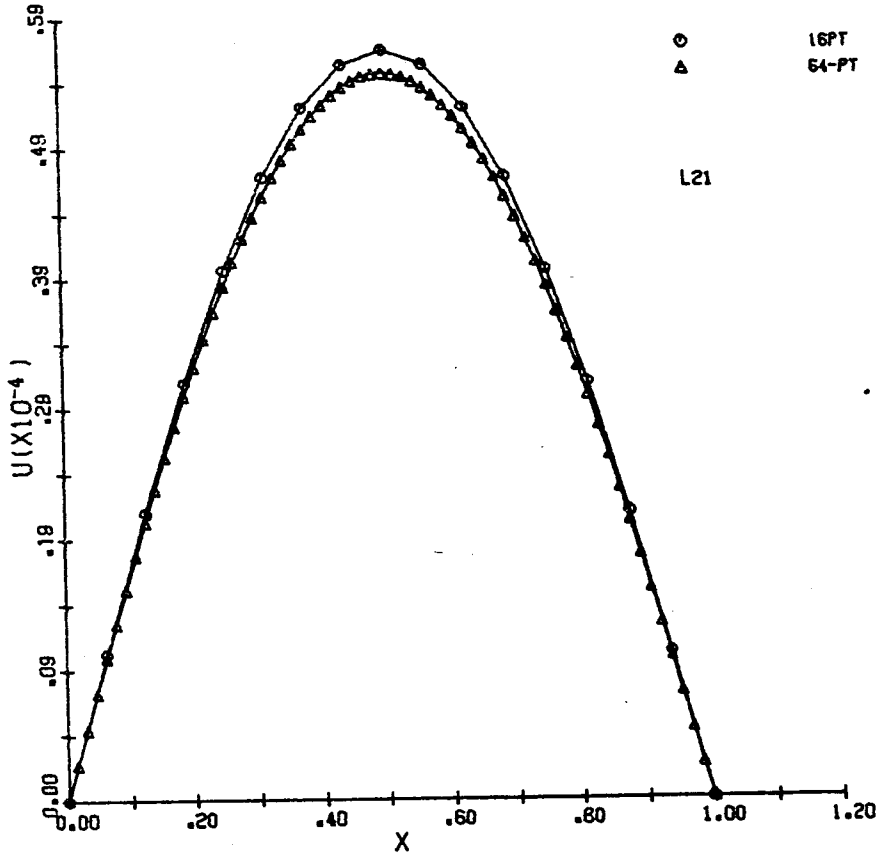


EXAMPLE II

FIGURE 6.9 - Trapezoidal Rule.

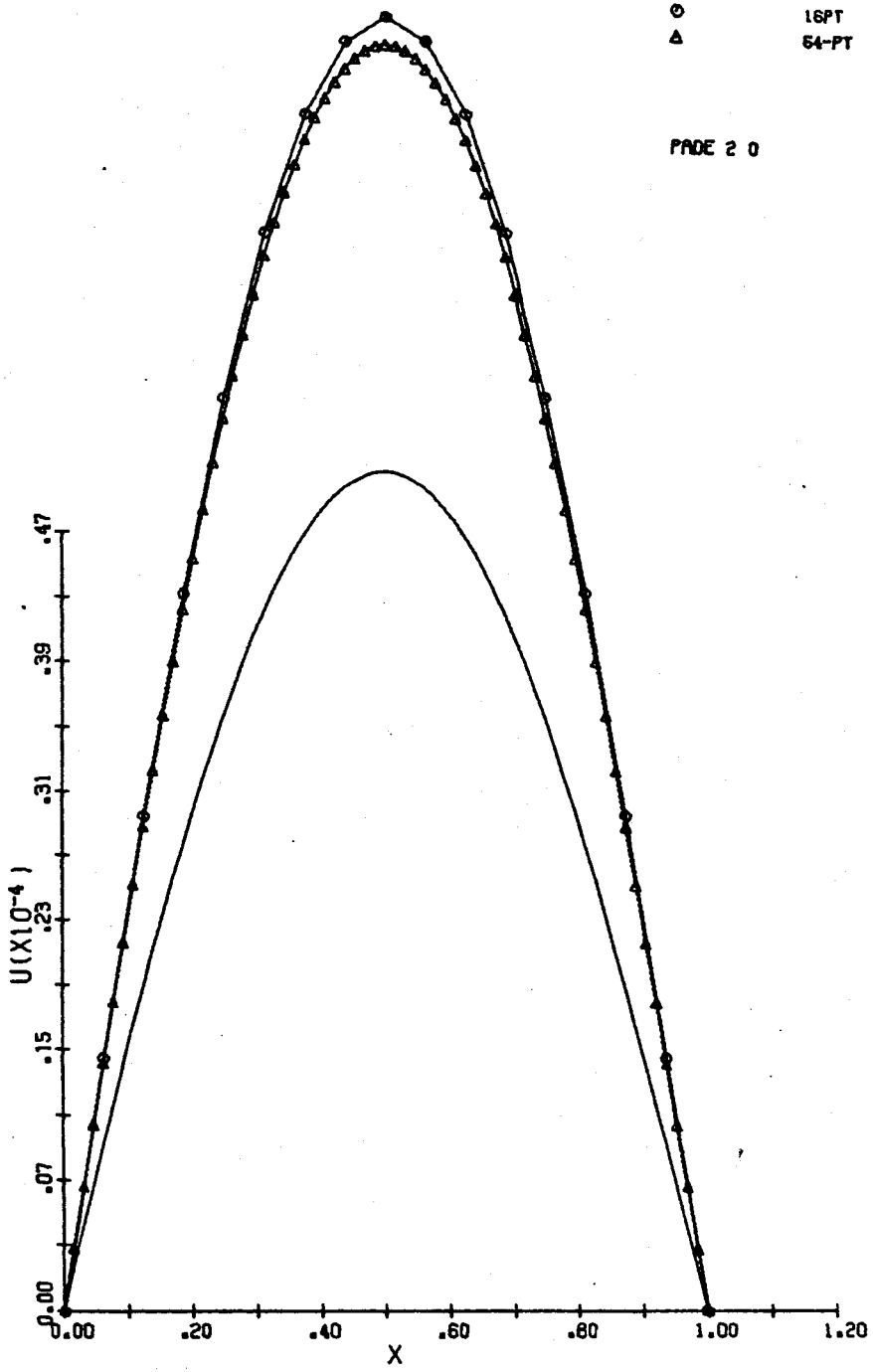


EXAMPLE I

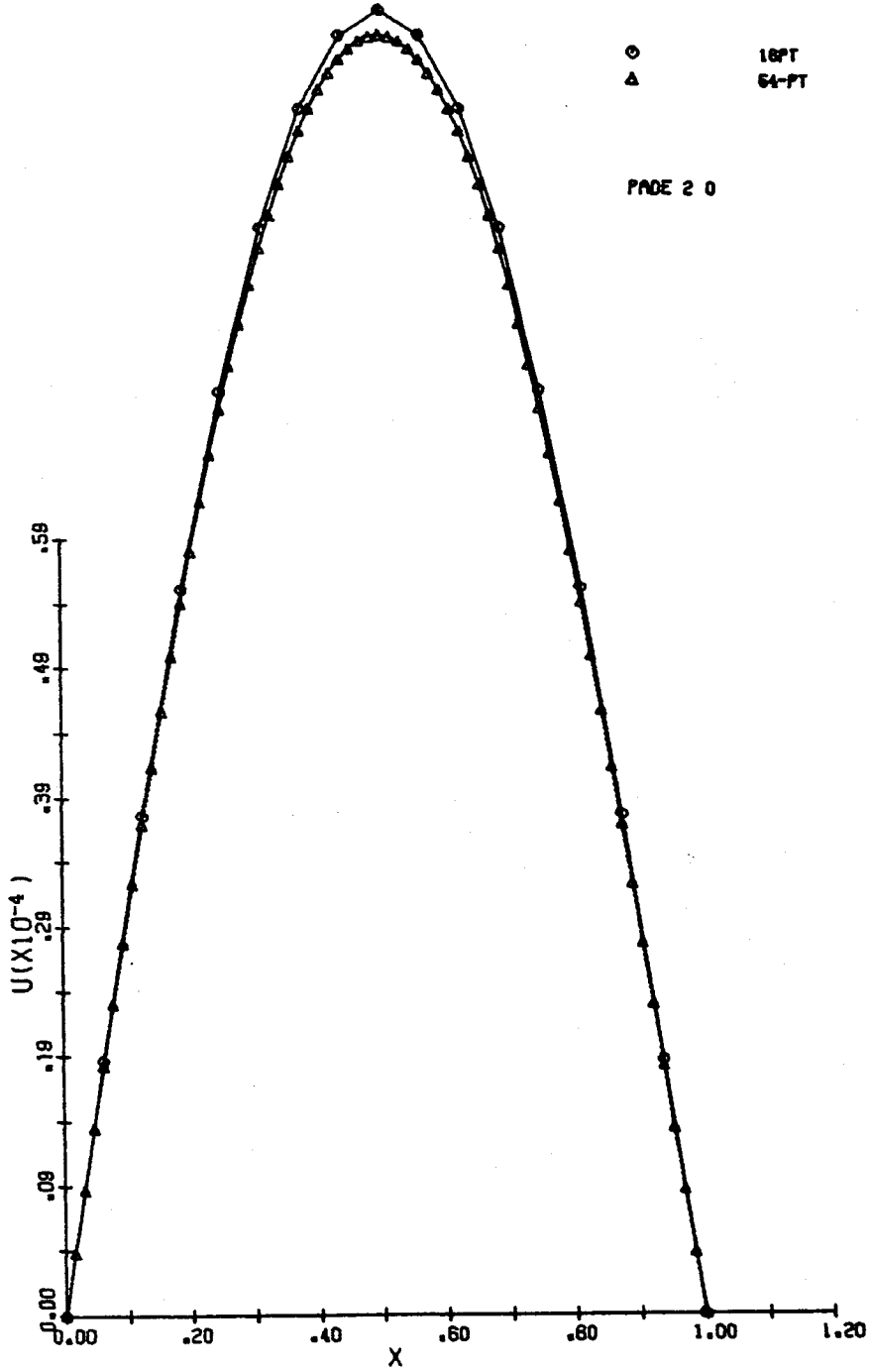


EXAMPLE II

FIGURE 6.10 - L21 Implementation.



EXAMPLE I



EXAMPLE II

FIGURE 6.11 - Padé(2,0) Implementation.

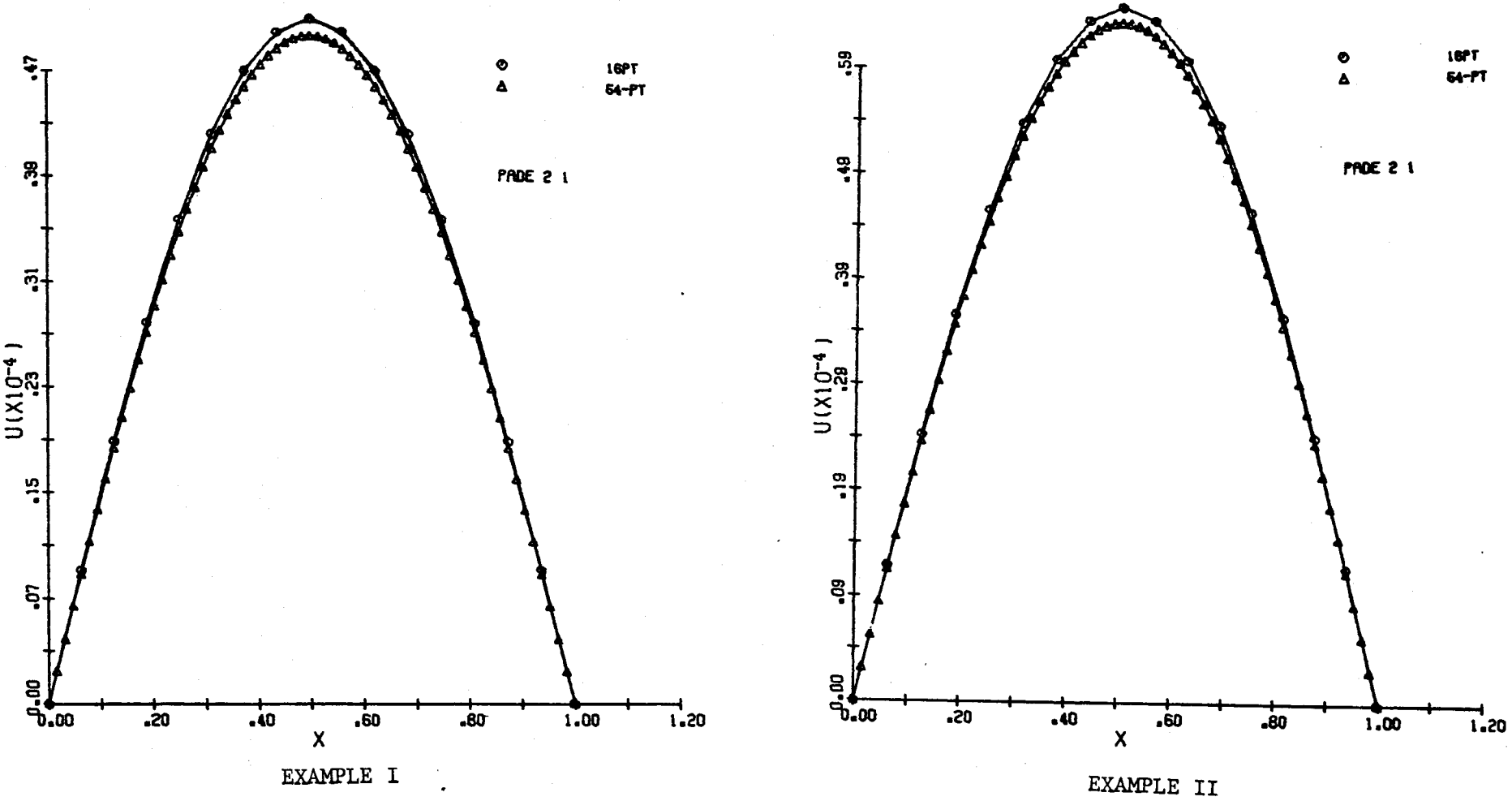


FIGURE 6.12 - Padé(2,1) Implementation.

Such experimentation leads naturally to the conclusion that the spatial discretization for partial differential equations of parabolic type is a problem which is independent of the exponential approximation chosen to simulate time-behavior. Thus, given a particular differential system which is written in the form of an ordinary differential system, with possible algebraic constraints, one may apply the results of Chapters 3 and 4 to particular exponential approximations to yield implementations which have the desired attributes, e.g. attenuation of transient components, to yield a solution to the ordinary differential equation problem which is as accurate as possible, from a numerical standpoint. The interplay of time and space discretization errors normally associated with partial differential equations need not necessarily be part of the discussion, provided the exponential approximation used has the desired stability properties.

BIBLIOGRAPHY

1. Brenner, J.L., "Expanded Matrices from Matrices with Complex Elements", SIAM Review 3 (1961), pp. 165-166.

2. Butcher, J.C., "On the Attainable Order of Runge-Kutta Methods", Math. Comp. 19 (1965), pp. 408-417.

3. Butcher, J.C., "Implicit Runge-Kutta Integration Processes", Math. Comp. 18 (1964), pp. 50-64.

4. Butcher, J.C., "Integration Processes Based on Radau Quadrature Formulas", Math. Comp. 18 (1964), pp. 233-244.

5. Butcher, J.C., "A Modified Multistep Method for the Numerical Integration of Ordinary Differential Equations", J. ACM 12 (1965), pp. 124-135.

6. Butkov, E., Mathematical Physics, Addison-Wesley, Reading, Mass., 1966.

7. Buzbee, W.L., "Application of Fast Poisson Solvers to the Numerical Approximation of Parabolic Problems", Report LA-4950-T, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1972.

8. Cavendish, J.C., Culham, W.E., and Varga, R.S., "A Comparison of Crank-Nicholson and Chebyshev Rational Methods for Numerically Solving Linear Parabolic Problems", J. Comp. Physics 10 (1972), pp. 354-368.

9. Chartres, B.A., and Geuder, J.C., "Computable Error Bounds for Direct Solution of Linear Equations", J. ACM 14 (1967), pp. 63-71.

10. Chipman, F., "Numerical Solution of Initial-Value Problems using A-Stable Runge-Kutta Processes", Dept. of A.A.C.S., University of Waterloo Research Report CSRR2042, 1971.

11. Coddington, E.A., and Levinson, N., Theory of Ordinary Differential Equations, McGraw-Hill, New York, 1955.

12. Cody, W.J., Meinardus, G., and Varga, R.S., "Chebyshev Rational Approximation to e^{-x} in $[0,\infty)$ and Application to Heat-Conduction Problems", J. Approx. Theory 22 (1969), pp. 50-65.

13. Dahlquist, G., "Convergence and Stability in the Numerical Integration of Ordinary Differential Equations", Math. Scand. 4 (1956), pp. 33-53.

14. Dahlquist, G., "A Special Stability Problem for Linear Multistep Methods", BIT 3 (1963), pp. 27-43.

15. Dill, C., and Gear, C.W., "A Graphical Search for Stiffly Stable Methods for Ordinary Differential Equations", J. ACM 18 (1971), pp. 75-79.

16. Ehle, B.L., "On Padé Approximations to the Exponential Function and A-Stable Methods for the Numerical Solution of the Initial-Value Problems", Dept. of A.A.C.S., University of Waterloo Research Report CSRR 2010, 1969.

17. Forsythe, G., and Moler, C., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Englewood Cliffs, N.J., 1967.

18. Gantmakher, F.R., Theory of Matrices, Vols. 1 and 2, Chelsea Publishing Co., New York, 1959.

19. Gear, C.W., "Numerical Integration of Stiff Ordinary Differential Equations", Report No. 221, Dept. of Comp. Sci., University of Illinois, Urbana, Ill., 1967.

20. Gear, C.W., "Rational Approximations by Implicit Runge-Kutta Schemes", BIT 10 (1970), pp. 20-22.

21. George, J.A., "On Block Elimination for Sparse Linear Systems", SIAM J. Numer. Anal. 11 (1974), pp. 585-603.

22. Gourlay, A.R., "A Note on Trapezoidal Methods for the Solution of Initial-Value Problems", Math. Comp. 24 (1970), pp. 629-633.

23. Isaacson, E., and Keller, H.B., Analysis of Numerical Methods, John Wiley and Sons, New York, 1966.

24. Jain, R.K., "Some A-Stable Methods for Stiff Ordinary Differential Equations", Math. Comp. 26 (1972), pp. 71-77.

25. Lambert, J.D., and Sigurdson, S.T., "Multistep Methods with Variable Matrix Coefficients", SIAM J. Numer. Anal. 9 (1972), pp. 715-733.

26. Lawson, J.D., "On the Exactness of Implicit Runge-Kutta Processes for Particular Integrals", BIT 12 (1972), pp.586-588.

27. Lawson, J.D., "Generalized Adams Methods for Stiff Systems of Ordinary Differential Equations", Utilitas Mathematica , to appear.

28. Lawson, J.D., "Generalized Runge-Kutta Processed for Stable Systems with Large Lipschitz Constants", SIAM J. Numer. Anal. 4 (1967) pp. 620-625.

29. Lawson, J.D., "Some Numerical Methods for Stiff Ordinary and Partial Differential Equations", Proc. Second Manitoba Conference on Numerical Math., 1972, pp. 27-34.

30. Lawson, J.D., "Order-Constrained Chebyshev Rational Approximations", Math. Comp. to appear.

31. Lees, M., "An Extrapolated Crank-Nicholson Difference Scheme for Quasilinear Parabolic Equations". Nonlinear Partial Differential Equations, (W.F. Ames, Ed.) Academic Press (1967), pp. 193-201.

32. Legras, J., "Resolution numérique des grands systèmes differentiels linéaires", Numer. Math. 8 (1966), pp. 14-28.

33. Liniger, W., and Willoughby, R., "Efficient Integration Methods for Stiff Systems of Ordinary Differential Equations", SIAM J. Numer. Anal. 7 (1970), pp. 47-66.

34. Loscalzo, F.R., "An Introduction to the Application of Spline Functions to Initial-Value Problems", Theory and Applications of Spline Functions, (T.N.E. Greville, Ed.), Academic Press, New York, 1969, pp. 37-64.

35. Ratchford, H.H.(Jr.), "Rounding Errors in Parabolic Problems. 1: The One-Space Variable Case", SIAM J. Numer. Anal. 5 (1968), pp. 156-171.

36. Shampine, L.F., "Local Extrapolation in the Solution of Ordinary Differential Equations", Math. Comp. 27 (1973). pp. 91-97.

37. Struble, R.A., Nonlinear Differential Equations, McGraw-Hill, New York, 1962.

38. Taussky, O., Problem #4846, Advanced Problems and Solutions, Amer. Math. Monthly, 66 (1959), p. 427.

39. Van der Sluis, A., "Stability of Solutions of Linear Algebraic Systems", Num. Math. 14 (1970), pp. 246-251.

40. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Oxford University Press, London, 1965.

41. Willoughby, R.A., Private Communication.