EVALUATION OF A FAMILY OF RUNGE-KUTTA ORIENTED PARALLEL METHODS FOR THE SOLUTION OF ODE'S

by

PING CHENG

A thesis submitted to the School of Graduate Studies
and Research of the University of Ottawa
in partial fulfillment of the requirements of
Master of Science in System Science

Ottawa, Ontario, 1995

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-15708-3

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ACKNOWLEDGMENTS

I would like to express my immense sense of gratitude to my supervisor, Dr. Louis G. Birta, for his advice, valuable guidance, encouragement to this research work and great effort in helping me to prepare this thesis.

I would like to thank Dr. Daniel Lane, director of the Systems Science Programme, for his assistance and encouragement throughout the course of my study.

Abstract

The numerical solution of ordinary differential equations (ODE's) can be a computationally intensive task. It is becoming widely believed that the only feasible means for solving such computationally intensive problems in science and engineering is to use parallel computers efficiently. As a result, there is an increasing interest in the development of parallel methods for the numerical solution of ODE's. This research is, for the most part, still in its preliminary stages.

Our goal in this thesis is to contribute to the evolving knowledge about parallel methods for the solution of ODE's. In this context, we examine in detail one particular class of methods. This class is Runge-Kutta oriented in the sense that the underlying computational process is based on Runge-Kutta formulas. However, from a broader perspective, the methods in this family also have an essential predictor-corrector feature.

Our study examines stability and performance aspects of this class of methods as originally proposed. In addition, a modification to the approach is suggested and similarly evaluated. Performance is examined in the context of a suite of test problems and results are compared to previously obtained results with two families of parallel Predictor-Corrector oriented methods.

Table of Contents

1. Introduction

1.1 Motivation and Overview of Past Work....................1
1.2 Notation and Background Concepts....................4
1.3 Contributions of the Thesis....................11
1.4 Organization of the Thesis....................12

2. Method Development and Implementation

2.1 Comparison of RK-Oriented and PC-Oriented Methods....................13
2.2 The Parallel Iterated Runge-Kutta (PIRK) Family of Methods......14
2.3 The Block PIRK (BPIRK) Family of Methods....................16
  2.3.1 Formulation of the BPIRK Approach....................16
  2.3.2 Generation of the Matrices V_i and W_i....................18
2.4 Implementation of the BPIRK Family of Methods....................25
2.5 Formulation of the BPIRK$^+$ Family of Methods....................31

3. Absolute Stability Bounds for the BPIRK and BPIRK$^+$ Families

3.1 Stability Matrices for the BPIRK and BPIRK$^+$ Families....................38
3.2 Stability Bounds for the BPIRK and BPIRK$^+$ Families....................41

4. Performance Evaluation

4.1 Scope of the Experiments....................58
4.2 Comparison Criteria....................59

4.3 Flowchart of the Experimental Program..60
4.4 Test Results..63
4.4.1 Validation of the Implementation..63
4.4.2 Evaluation of Test Results..65

5. Concluding Remarks

5.1 Conclusions..85
5.2 Future Work....... 86

Appendix I

Test Problems Used in the Thesis...87

Appendix II

Implicit RK Formulas Used in the BPIRK and BPIRK$^+$ Processes...........91

Appendix III

Structure of the Program...93

Bibliography...**95**

List of Figures

Figure 1 A Highlevel View of the BPIRK Solution Process........................25
Figure 2(a) Initialization Procedure...26
Figure 2(b) Solution Procedure...27
Figure 2(c) Solution Procedure for $U_i^{(m)}$...29
Figure 2(d) Termination Procedure...30
Figure 3 Configuration for BPIRK Family of Methods..............................31
Figure 4 Configuration for $BPIRK^+$ Family of Methods.............................32
Figure 5 Stability Bounds for BPIRK Methods of Order 3..........................46
Figure 6 Stability bounds for $BPIRK^+$ Methods of Order 3.........................47
Figure 7 Stability Bounds for BPC Methods of Order 3.............................48
Figure 8 Stability Bounds for PBPC Methods of Order 3...........................49
Figure 9 Stability Bounds for BPIRK Method of Order 5...........................50
Figure 10 Stability Bounds for $BPIRK^+$ Methods of Order 5.......................51
Figure 11 Stability Bounds for BPC Methods of Order 5.............................52
Figure 12 Stability Bounds for PBPC Methods of Order 5...........................53
Figure 13 Stability Bounds for BPIRK Methods of Order 7..........................54
Figure 14 Stability Bounds for $BPIRK^+$ Methods of Order 7........................55
Figure 15 Stability Bounds for BPC Methods of Order 7.............................56
Figure 16 Stability Bounds for PBPC Methods of Order 7...........................57
Figure 17 A Highlevel View of Experimentation..60
Figure 18(a) Initialization Procedure for Experimentation................................61
Figure 18(b) Termination Procedure for Experimentation................................62

List of Tables

Table 1 Stability Bounds for BPIRK Family with Lagrange Predictor.........45

Table 2 Stability Bounds for BPIRK Family with Hermite Predictor...........45

Table 3 accuracy_order Obtained in [12]..63

Table 4 accuracy_order Obtained in Our Implementation............................64

Table 5 Test Results for TP1-TP5 Using BPIRK with the Lagrange Predictor...68

Table 6 Test Results for TP1-TP5 Using BPIRK$^+$ with the Lagrange Predictor...70

Table 7 Test Results for TP1-TP5 for BPC Family.......................................72

Table 8 Test Results for TP1-TP5 for PBPC Family.....................................73

Table 9 Test Results for TP1 Using BPIRK with the Lagrange Predictor....74

Table 10 Test Results for TP2 Using BPIRK with the Lagrange Predictor....75

Table 11 Test Results for TP3 Using BPIRK with the Lagrange Predictor....76

Table 12 Test Results for TP4 Using BPIRK with the Lagrange Predictor....77

Table 13 Test Results for TP5 Using BPIRK with the Lagrange Predictor....78

Table 14 Test Results for TP1 Using BPIRK with the Hermite Predictor......79

Table 15 Test Results for TP2 Using BPIRK with the Hermite Predictor......80

Table 16 Test Results for TP3 Using BPIRK with the Hermite Predictor......81

Table 17 Test Results for TP4 Using BPIRK with the Hermite Predictor......82

Table 18 Test Results for TP5 Using BPIRK with the Hermite Predictor......83

Table 19 Test Results for TP5 Using BPIRK with the Hermite Predictor......84

Chapter 1 Introduction

1.1 Motivation and Overview of Past Work

In a continuous system simulation study, the system is typically described by a set of first-order ordinary differential equations (ODE's); i.e., a set of equations of the form:

$$y' = \frac{d}{dt} y = f(t, y(t)) \tag{1.1.1}$$

where, y and f are vectors of dimension L.

In this thesis, our concern is with the solution of (1.1.1) over the time interval $[t_0, t_f]$ with $y(t_0)=y_0$ where t_0, t_f and y_0 are given. This is usually referred to as the initial value problem (IVP). The discussion in this thesis is concerned with nonstiff problems. Discussion of stiff IVPs can be found in [2], [3], [4], [8], [10], [17], and [18].

The fundamental step in the simulation activity consists of solving these equations numerically. The numerical solution of ODE's can require substantial computation. As a result, a variety of parallel methods for carrying out this numerical task have been proposed in the literature.

Gear [13], [14] classifies the approaches for achieving parallelism in solving a set of ODE's into two broad categories:

(i) parallelism across the method or equivalently parallelism across time;

(ii) parallelism across the system or equivalently parallelism across space.

Included in class (i) are algorithms that exploit several concurrent function evaluations within a step; e.g., the Runge-Kutta (RK) oriented methods discussed in the next chapter, as well as techniques that solve for many steps simultaneously;

e.g., the Predictor-Corrector (PC) oriented methods [1], [5], [7], [13], [14]. The techniques that exploit parallelism in the evaluation of f as well as in performing vector operations and in solving linear and nonlinear equations that may arise at each step of the solution process fall into the second class, class (ii), of methods. Waveform relaxation [27], [29] is one of the currently active area of this class.

Parallelism across a method implies that parts of the integration procedure itself (rather than parts of the problem to be solved) are distributed over the set of available processors. A highly desirable feature in this distribution is the achievement of a balanced workload. An extension of the classical predictor-corrector approach to a block (i.e., multi-point) context provides a relatively straightforward means for achieving parallelism with a balanced workload. In this PC-oriented approach, each of the available processors carries out a computation that relates to a distinct point on the solution axis. The computation corresponds either to a computation of a predicted or a corrected value or to a derivative function evaluation. All processors carry out a sequence of computationally identical tasks; hence, a balanced workload is achieved.

The Block Predictor-Corrector (BPC) and the Parallel Block Predictor-Corrector (PBPC) methods are two particular subfamilies of PC-oriented parallel methods. The BPC family has its origins in the work of Shampine and Watts [35] who suggested an extension of predictor-corrector methods into a block context using the earlier work of Rosser [34]. Birta and Abou-Rabia [5] extended this work and, in particular, suggested a way to enlarge the stability interval. Tam [36] has constructed an interesting subfamily of BPC methods which has the feature that the stability region of its members does not decrease as order increases.

Miranker and Lininger [28] established the basis for the PBPC family. Their work was later refined by Katz et al. [26]. A comparison of some methods from both these families with respect to stability properties and performance using a suite of test problems, can be found in Abou-Rabia et al. [1].

An alternate and potentially attractive approach has recently evolved from an extension of Runge-Kutta methods [19], [20], [21], [22], [23], [24], [25] and [30]. In this approach, an RK formula (explicit or implicit) is employed as a base method. Parallelism is achieved by exploiting the multiple function evaluations required within each step of a standard one-step RK formula. In the case where the base method is implicit, further opportunities for parallelism arise in solving the associated nonlinear equations. Miranker and Lininger [28] derived a class of RK-oriented methods, but their approach is ineffective because of poor stability.

An s-stage RK method for the numerical solution of (1.1.1) is defined by:

$$g_i = f(t+c_i h,\ y(t)+h\sum_{j=1}^{s} a_{ij}g_j),\quad i=1,2,\ldots,s \tag{1.1.2}$$

and

$$y_{t+h} = y(t) + h\sum_{i=1}^{s} b_i g_i \tag{1.1.3}$$

where y_{t+h} provides an approximation to y(t+h) developed from the known solution value y(t). The scalar h represents the integration stepsize.

In an explicit process, g_i depends only on g_j for $j<i$; i.e.,

$$a_{ij}=0 \quad j\geq i \tag{1.1.4}$$

An implicit process, on the other hand, is one for which (1.1.4) is not satisfied. A comprehensive discussion of implicit formulas can be found in [9], [11], [17], and [18].

The use of explicit RK formulas is not attractive in developing parallel methods because the order of the resulting method is limited and the stability region is small [23].

A recently proposed RK-oriented approach is a family of iterated RK methods. These have their origin in the work of van der Houwen and Sommeijer [19] who formulated the Parallel Iterated RK (PIRK) methods. These were extended by van der Houwen and Cong [20] into Block PIRK(BPIRK) methods. The underlying (i.e., base) RK method is implicit. Both PIRK and BPIRK methods are implemented as a predictor-corrector scheme. In the predictor phase, an initial value is produced for the internal stage values g_i in (1.1.2). The corrector phase refines these values via some iterative scheme. Finally the resultant solution value is produced using (1.1.3). A variety of approaches is available to produce the initial (i.e., predicted) values for g_i. A change in the predictor formula used, in the underlying RK formula, in the iteration process, or even in the number of iterations of the corrector formula gives a different member of the family. Jackson and Nørsett [36] refer to these iterated RK methods as RK-PC methods.

Parallel methods in Gear's first category are, for the most part, PC-oriented or RK-oriented. Comprehensive numerical evaluations of methods from both these broad families remain to be carried out in order to obtain better insight into their strengths and weaknesses.

1.2 Notation and Background Concepts

We summarize below some notation and concepts that will be used throughout the thesis. The presentation is separated into two categories. The first reviews aspects which are relatively widely used while the second is distinctive to the particular needs of the work outlined in the thesis; its relevance begins in section 2.3.

Category 1:

a) Order

Suppose we have a numerical process for the solution of (1.1.1) which generates a solution value y_{n+1} at $t_{n+1}=t_n+h$. The method is said to be of order p (or to be of p^{th}-order) if

$$y(t_n+h) - y_{n+1} = O(h^{p+1})$$

Here $y(t_n+h)$ is the solution to (1.1.1) which passes through $y(t_n)$ (and through any other points that are used by the numerical process in producing y_{n+1}).

b) Runge-Kutta methods

From (1.1.2) and (1.1.3), it can be observed that an s-stage Runge-Kutta method can be readily characterized by the three arrays :

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \cdot & \cdot & \cdots & \cdot \\ a_{s1} & a_{s2} & \cdots & a_{ss} \end{bmatrix}$$

$$b=(b_1,b_2,\ldots,b_s)^T$$

and

$$c=(c_1,c_2,\ldots,c_s)^T$$

A convention which we also adopt is that:

$$c_i = \sum_{j=1}^{s} a_{ij} \qquad (1.2.1)$$

and

$$\sum_{i=1}^{s} b_i = 1 \qquad (1.2.2)$$

Virtually all the existing methods satisfy (1.2.1) and (1.2.2) [11].

A compact and widely used representation for an s-stage RK formula is provided by the so-called Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

c) Functional iteration

In this section we consider a system of L equations in L unknowns expressed in the form:

$$\begin{aligned} x_1 &= g_1(x_1, x_2, \ldots, x_L) \\ x_2 &= g_2(x_1, x_2, \ldots, x_L) \\ &\bullet \quad \bullet \quad \bullet \quad \bullet \\ x_L &= g_L(x_1, x_2, \ldots, x_L) \end{aligned} \tag{1.2.3}$$

Let $X=(x_1, x_2, \ldots, x_L)^T$ and $G=(g_1, g_2, \ldots, g_L)^T$ then (1.2.3) is be written as

$$X = G(X) \tag{1.2.4}$$

Functional iteration is one of several possible numerical procedures for finding the solution X^* of (1.2.4). Let $X^0=(x_1^0, x_2^0, \ldots, x_L^0)^T$ be an initial approximation to X^*. Consider the sequence $\{X^j\}$ which is recursively generated by the relation :

$$X^j = G(X^{j-1}), \quad j=1,2,\ldots$$

Under appropriate conditions, X^j will converge to X^* as $j \to \infty$. The following lemma [33] gives a set of conditions which assures this convergence.

Lemma 1.1: Suppose that the system (1.2.4) has a solution X^*. Further, suppose that for some $\delta > 0$ there exists a neighborhood, Δ, of radius δ about X^* in which

$$\frac{\partial g_i}{\partial x_j}, \quad i=1,2,\ldots,L,\ j=1,2,\ldots,L$$

exist for each point of Δ. Finally, suppose that there exists a constant $0<\varepsilon<1$ such that either

$$\max_{1\leq i\leq L}\sum_{j=1}^{L}\left|\frac{\partial g_i}{\partial x_j}\right|<\varepsilon$$

or

$$\max_{1\leq j\leq L}\sum_{i=1}^{L}\left|\frac{\partial g_i}{\partial x_j}\right|<\varepsilon$$

is satisfied for all the points in Δ. If $\|X^0-X^*\|<\delta$ and

$$X^j = G(X^{j-1}),\quad j=1,2,\ldots \qquad (1.2.5)$$

then

$$\lim_{j\to\infty} X^j = X^*$$

Other sufficient conditions which ensure that the functional iteration process converges to X^* are discussed by Ortega [31].

d) Stability boundary

The ODE's of (1.1.1) have certain stability properties. The aim of stability analysis, in general, is to determine under what conditions any particular discrete replacement (i.e., the numerical method) preserves such properties. This has a significant bearing on the usefulness and reliability of the numerical method. The most widely used approach is to find the stability region of the numerical method under consideration [3], [18]. We outline this approach in the following discussion.

Consider the scalar equation

$$y' = \lambda y \qquad (1.2.6)$$

where λ is a complex scalar. It is straightforward to demonstrate that if $Re(\lambda) \leq 0$ then any solution y(t) of (1.2.6) remains bounded.

A numerical method (using stepsize h) is said to be absolutely stable for $h\lambda$, if it produces a bounded approximations when applied to problem (1.2.6). The region of stability, S, is the region of the $h\lambda$-plane for which the method is absolutely stable. It is appropriate to talk of an interval of absolute stability, which is simply the intersection of S with the real axis. The left most point of this interval is said to be the stability boundary.

Both PC and RK methods can be uniformly presented in the following form:

$$Y_{n+1} = \Phi(h, f(t_1,y_1), f(t_2,y_2), \ldots, f(t_k,y_k))Y_n \tag{1.2.7}$$

Here Y_n is a k-vector of known (previously computed) solution values and Y_{n+1} is a vector of new values to be generated (possibly in parallel).

The substitution of $f(t,y_i) = \lambda y_i$, $i=1,2,\ldots,k$, in (1.2.7) yields

$$Y_{n+1} = \Phi(h, \lambda y_1, \lambda y_2, \ldots, \lambda y_k)\, Y_n = R(\lambda h)Y_n \tag{1.2.8}$$

i.e.

$$Y_{n+1} = R^n(\lambda h)Y_0$$

It is clear that if the norm of matrix $R(\lambda h)$ is less than one, for a fixed h, the value of Y_{n+1} will be bounded as $n \rightarrow \infty$. We define the matrix $R(\lambda h)$ as the stability matrix for method (1.2.7). The following lemma [32] gives a necessary and sufficient condition for stability.

Lemma 1.2 The numerical method (1.2.7) for solving (1.1.1) is stable if and only if the maximum magnitude of the eigenvalues of matrix $R(\lambda h)$ is less than one, i.e., $\rho(R(\lambda h))<1$.

Category 2:

e:	a vector whose entries are all unity (the length of e is generally clear from the context)
e_i:	a vector whose entries are all zero except for the i^{th} entry which is one (i.e., e_i is the i^{th} column of an identity matrix of compatible size)
E_i:	a matrix with zero entries except for those in the i^{th} column which are all unity

In the interests of notation convenience (but without loss of generality) we will take L in (1.1.1) to be unity; i.e., f and y are taken to be scalars. Often, however, it is convenient to write f with vector arguments; e.g., f(T,Y) where T and Y are vectors of equal length, k. In such a case f(T,Y) represents a vector of length k whose i^{th} entry is $f(T_i,Y_i)$, i=1,...,k.

Let μ, ν and τ be vectors of length k; i.e., $\mu=(\mu_1, \mu_2,\ldots, \mu_k)^T$, $\nu=(\nu_1,\nu_2,\ldots,\nu_k)^T$ and $\tau=(\tau_1, \tau_2,\ldots, \tau_k)^T$, then we define:

$$\mu^j = (\mu_1^j, \mu_2^j,\ldots, \mu_k^j)^T$$

and

$$\mu .\!* \nu = (\mu_1\nu_1, \mu_2\nu_2, \ldots, \mu_k\nu_k)^T$$

Also $\tau\in[\mu, \nu]$ means $\tau_i\in[\mu_i, \nu_i]$ for each i=1,2,...,k.

From a Taylor series expansion, we can write:

$$y(te+hx) = y(t)e + hx\,\frac{d}{dt}y(t) + \ldots + \frac{(hx)^j}{j!}\,\frac{d^j}{dt^j}y(t) + \ldots + \frac{(hx)^{q+1}}{(q+1)!}\odot\frac{d^{q+1}}{dt^{q+1}}y(t^*)$$

$$= \sum_{j=0}^{q}\frac{1}{j!}\{(hx)^j\,\frac{d^j}{dt^j}y(t)\} + \frac{1}{(q+1)!}(hx)^{q+1}\odot\frac{d^{q+1}}{dt^{q+1}}y(t^*)$$

$$t^*\in[te, te+hx]$$

where y, t and h are scalars, e and x are vectors of the same length. For convenience, we let $\Lambda^q[y,t,hx]$ represent the right-hand side of the above expansion; i.e.,

$$\Lambda^q[y,t,hx] = y(t)e + hx\,\frac{d}{dt}y(t) + \ldots + \frac{(hx)^j}{j!}\,\frac{d^j}{dt^j}y(t) + \ldots + \frac{(hx)^{q+1}}{(q+1)!} \odot \frac{d^{q+1}}{dt^{q+1}}y(t^*)$$

$$= \sum_{j=0}^{q}\frac{1}{j!}\{(hx)^j\,\frac{d^j}{dt^j}y(t)\} + \frac{1}{(q+1)!}(hx)^{q+1} \odot \frac{d^{q+1}}{dt^{q+1}}y(t^*)$$

Note also that:

$$\frac{d}{dt}y(te+hx) = \frac{d}{dt}y(t)e + hx\frac{d^2}{dt^2}y(t) + \ldots + \frac{(hx)^{j-1}}{(j-1)!}\,\frac{d^j}{dt^j}y(t) + \ldots + \frac{(hx)^q}{q!} \odot \frac{d^{q+1}}{dt^{q+1}}y(t_1^*)$$

$$= \sum_{j=0}^{q}\frac{1}{j!}\{j(hx)^{j-1}\frac{d^j}{dt^j}y(t_{n-1})\} + \frac{1}{q!}(hx)^q \odot \frac{d^{q+1}}{dt^{q+1}}y(t_1^*), \qquad t_1^* \in [te, te+hx]$$

It is clear that the right-hand side of the above is $\Gamma^{q-1}[\frac{d}{dt}y,t,hx]$; i.e.,

$$\Lambda^{q-1}[\frac{d}{dt}y,t,hx] = \frac{d}{dt}y(t)e + hx\frac{d^2}{dt^2}y(t) + \ldots + \frac{(hx)^{j-1}}{(j-1)!}\,\frac{d^j}{dt^j}y(t) + \ldots + \frac{(hx)^q}{q!} \odot \frac{d^{q+1}}{dt^{q+1}}y(t_1^*)$$

$$= \sum_{j=0}^{q}\frac{1}{j!}\{j(hx)^{j-1}\frac{d^j}{dt^j}y(t_{n-1})\} + \frac{1}{q!}(hx)^q \odot \frac{d^{q+1}}{dt^{q+1}}y(t_1^*)$$

That is:

$$y(te+hx) = \Lambda^q[y,t,hx]$$

and

$$\frac{d}{dt} y(te+hx) = \Lambda^{q-1}[\frac{d}{dt} y,t,hx]$$

1.3 Contributions of the Thesis

The central concern of this thesis is with an examination of parallel methods for the solution of (1.1.1). The specific contributions within this context are:

i) Formulation of the BPIRK family of methods and the specification of the essential steps in the solution process; i.e., specification of the associated algorithm required for implementation.

ii) Formulation of a modification to the BPIRK approach which seeks to enhance its performance. We refer to this modified family as the BPIRK$^+$ family.

iii) Derivation of stability matrices for both the BPIRK and BPIRK$^+$ methods and the determination of the stability bounds for these methods for various values of internal parameters.

iv) Examination of the numerical performance of BPIRK methods using a suite of test problems (see Appendix I). This is carried out using a program package written in C. The program executes on a single processor and simulates parallel operations.

v) Comparison of the performance of the RK-oriented BPIRK and BPIRK$^+$ methods with that of the PC-oriented BPC and PBPC methods. For the latter, we use results previously reported in [1], [6], and [15] .

1.4 Organization of the Thesis

We introduce BPIRK methods in Chapter 2. Section 2.1 provides a comparison of RK-oriented methods with PC-oriented methods. A derivation of PIRK methods is given in section 2.2 and an extension of these methods into the BPIRK methods is formulated in section 2.3. In section 2.4, we present a set of flowcharts for a software package for implementing the BPIRK family of methods. Section 2.5 develops the BPIRK$^+$ family.

Chapter 3 discusses the stability of both the BPIRK and BPIRK$^+$ families. Stability matrices for both BPIRK and BPIRK$^+$ families are first derived. Stability bounds of the BPIRK and BPIRK$^+$ families are compared. In addition, these bounds are compared with the stability bounds of the BPC and PBPC families.

Performance evaluation is given in Chapter 4. Section 4.2 gives the comparison criteria. Flowcharts of the experimental program are given in section 4.3. A validation of our implementation of the BPIRK approach is provided by executing one of the test problems used by van der Houwen and Cong [20]. This validation is described in section 4.4. The results of an extensive set of numerical experiments are presented and discussed in sections 4.4 and 4.5.

Chapter 5 summarizes this research work. Some further projects related to this thesis are given in section 5.2.

The test problems used in the numerical experiments are given in Appendix I. The implicit RK formulas used in the implementation of the BPIRK and BPIRK$^+$ methods are provided in Appendix II. A description of the structure for the experimental program of the BPIRK family of methods is given in Appendix III.

Chapter 2 Method Development and Implementation

2.1 Comparison of RK-Oriented and PC-Oriented Methods

As an introduction to a detailed examination of parallel RK-oriented methods, it is helpful to examine the main features of scalar (i.e., single-processor) RK-oriented and PC-oriented methods.

A PC-oriented method, in general, is constructed from a predictor-corrector pair. Such a pair has the following typical form:

$$y_{n+1}^{0} = \alpha^{0} \bullet Y_n + h\beta^{0} \bullet f(T_n, Y_n) \qquad \text{(Predictor)} \quad (2.1.1)$$

$$y_{n+1} = \alpha \bullet Y_{n+1}^{0} + h\beta \bullet f(T_{n+1}, Y_{n+1}^{0}) \qquad \text{(Corrector)} \quad (2.1.2)$$

Here $Y_n = (y_n, y_{n-1}, y_{n-2}, \ldots, y_{n-k+1})^T$ and $Y_{n+1}^{0} = (y_{n+1}^{0}, y_n, \ldots, y_{n-k+3}, y_{n-k+2})^T$ are vectors of length k, while α^0, β^0, α and β are k-dimensional parameter vectors and h is the stepsize. T_n and T_{n+1} are also a k-dimensional vector which provides the time points that correspond to Y_n and Y_{n+1} respectively. The value y_{n+1} is an approximation to $y(t_n+h)$. Furthermore, $y_0 = y(t_0)$ and $t_n = t_0+nh$, where t_0 is the initial point on the time axis.

The PC-pair (2.1.1) and (2.1.2) shows that previously obtained solution values (as contained in Y_n) are first used in the predictor formula (2.1.1) to produce a tentative solution value y_{n+1}^{0}. These are subsequently refined by the corrector formula (2.1.2) to produce the final solution value y_{n+1}.

An s-stage Runge-Kutta method has the form:

$$g_i = f(t_n+c_ih,\ y_n+h\sum_{j=1}^{s} a_{ij}g_j), \quad i=1,2,\ldots,s \qquad (2.1.3)$$

and

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i g_i \tag{2.1.4}$$

In (2.1.3) and (2.1.4), the only previous solution value that is used is the most current value, y_n. To obtain the g_i, function evaluations take place at a set of intermediate points (i.e., t_n+c_ih, $i=1,2,\ldots,s$) and the new solution value y_{n+1} is found from only this information without relying on previous results. However, the many function evaluations necessary can introduce significant round-off error.

The RK-oriented methods are self-starting. The PC-oriented methods, on the other hand, need some other technique to generate the k starting values in Y_0. This makes these methods more complex.

Parallel RK-oriented and PC-oriented methods share most of these same features.

Many RK-oriented and PC-oriented parallel families of methods have been proposed in the past few years [3], [1], [19], [20], [22], [21], [24]. Theoretical and practical results have demonstrated that these methods can be practically useful. Much further work, however, remains to be done in order to obtain deeper insights into the strengths and weaknesses of these approaches.

2.2 The Parallel Iterated Runge-Kutta (PIRK) Family of Methods

The PIRK family of parallel methods for solving (1.1.1) was originally proposed by van der Houwen and Sommeijer [19]. An overview of this approach is presented below.

Consider the s-stage, implicit, one-step Runge-Kutta method given in (2.1.3) and (2.1.4) with the definitions:

$$g = (g_1, \ldots, g_s)^T$$

and

$$A_i = (a_{i1}, \ldots, a_{is})^T$$

Formulas (2.1.3) and (2.1.4) can be written as:

$$g_i=f(t_n+c_ih,\ y_n+hA_i^T g) \qquad i=1,2,\ldots,s \tag{2.2.1}$$

and

$$y_{n+1}=y_n+hb^Tg \tag{2.2.2}$$

Using the notation conventions introduced in section 1.3, (2.2.1) can be written in a compact form as:

$$g=f(t_ne+ch,\ y_ne+hAg) \tag{2.2.3}$$

The implementation of this RK process therefore requires a solution to the implicit equation given in (2.2.3) at each time step. Any method for solving a system of nonlinear equations can be applied here. The simplest is functional iteration (see section 1.3). The recurrence relation associated with this approach is:

$$g^{(j)}=f(t_ne+ch,\ y_ne+h\,Ag^{(j-1)}), \qquad j=1,2,\ldots,m \tag{2.2.4}$$

Here m is a parameter of the solution process and $g^{(m)}$ is used as an approximation for g in (2.2.1) and (2.2.2) thereby achieving a realization of the method. It is important to observe that each of the s components of $g^{(j)}$ can be computed independently by a separate processor which thereby introduces parallelism into the solution procedure.

This process requires an initial value (i.e., a value for $g^{(0)}$) which is frequently taken to be $g^{(0)}=f(t_n,\ y_n)e$. The method used to determine the initial value $g^{(0)}$ is called a predictor for the PIRK method.

2.3 The Block PIRK(BPIRK) Family of Methods

2.3.1 Formulation of the BPIRK Approach

An s-stage PIRK method provides limited parallelism inasmuch as only s separate processors can be utilized. This underlying idea was extended into a block context by van der Houwen and Cong [20]. In their formulation, a block of M solution values is generated. However, only one of these points is used as an output value while the remainder are used only for producing predicted values in the subsequent (adjacent) block. The approach idea is called the Block PIRK (i.e., BPIRK) approach.

The development proceeds by first modifying the relationships in (2.2.2) and (2.2.3). More specifically, premultiply (2.2.3) by hA and then add $y_n e$ to both sides to obtain:

$$y_n e+hAg = y_n e+hAf(t_n e+hc,\ y_n e+hAg) \tag{2.3.1}$$

Let U= $y_n e+hAg$, then (2.3.1) can be written as:

$$U = y_n e+hAf(t_n e+hc,\ U) \tag{2.3.2}$$

and (2.2.2) becomes

$$y_{n+1}= y_n+hb^T f\ (t_n e+hc,\ U) \tag{2.3.3}$$

The relations in (2.3.2) and (2.3.3) are an alternate representation of an s-stage implicit Runge-Kutta process and provide the basis for our subsequent discussion.

Observe that the specifications in (2.3.2) and (2.3.3) can be applied for a set of distinct stepsizes $h_i = d_i h$, i=1,2,...,M; i.e.,

$$U_i = y_n e+h_i Af(t_n e+h_i c,\ U_i), \qquad i= 1,2,...,M \tag{2.3.4}$$

$$y_{n+1,i} = y_n + h_i b^T f\,(t_n e + h_i c,\ U_i)\ , \qquad i = 1,2,\ldots,M \qquad (2.3.5)$$

A convenient choice is $d_1=1$ in which case $h_1=h$ and the value $y_{n+1,1}$ corresponds to $t=t_{n+1}$, in other words, $y_{n+1,1}=y_{n+1}$. If $d_i>1$ for $i=2,3,\ldots,M$, the remaining values $y_{n+1,i}$, will fall to the right of $y_{n+1,1}$. For notational convenience, the values $y_{n+1,i}$ are collected together in the M-vector Y_{n+1}, i.e., $Y_{n+1}=(y_{n+1,1}, y_{n+1,2}, \ldots, y_{n+1,M})^T$ which represents a block of M numerical approximations to the exact solution values $y(t_{n+1,i})$, with $t_{n+1,i}=t_n+h_i= t_n+d_ih$. In a similar way, $Y_n=(y_{n,1}, y_{n,2}, \ldots, y_{n,M})^T$ with $t_{n,i}=t_{n-1}+h_i= t_{n-1}+d_ih$ and $y_{n,1}=y_n$. In the special case where n=0 (i.e., at the first step), we take the M-vector $Y_0=(y_0, \ldots, y_0)^T$, where y_0 is the given initial value.

The generation of the block of solution values Y_{n+1} (via (2.3.5)) requires a solution to the implicit equations given by (2.3.4). Like the PIRK method, this can be achieved via a functional iteration process; i.e.,

$$U_i^{(j)} = y_n e + h_i A f(t_n e + h_i c,\ U_i^{(j-1)}\ ), \qquad j=1,2,\ldots,m \qquad (2.3.6)$$

where m is an integer parameter which corresponds to the number of iterations used during the solution process.

Note that a separate processor can be used to carry out the calculation specified in (2.3.6) for each of the M values of i. Furthermore each component of the s-vector $U_i^{(j)}$ can itself be evaluated by a separate processor. Consequently the computation implied by (2.3.6) can be distributed over M×s processors. It is particularly important to observe that this distribution is evenly balanced inasmuch as processors are uniformly active.

The solution values within the M-vector Y_{n+1} are obtained (from (2.3.5)) as:

$$y_{n+1,i} = y_n + h_i b^T f\,(t_n e + h_i c,\ U_i^{(m)})\ , \qquad i = 1,2,\ldots,M \qquad (2.3.7)$$

This final step in the procedure can be uniformly distributed over M processors.

The functional iteration process specified in (2.3.6) requires a value for $U_i^{(0)}$. Various alternatives can be used to obtain this initial (i.e., predicted) value. Following are three examples suggested by van der Houwen and Cong [20]. Their work utilized only the Lagrange predictor; in our work we consider both Lagrange and Hermite predictors.

Lagrange: $$U_i^{(0)} = V_i\, Y_n \tag{2.3.8}$$

Hermite: $$U_i^{(0)} = V_i\, Y_n + hW_i f(T_n, Y_n) \tag{2.3.9}$$

Adams: $$U_i^{(0)} = E\, Y_n + hW_i f(T_n, Y_n) \tag{2.3.10}$$

A formulation of the specifications for the s×M matrices V_i and W_i will be given in section 2.3.2. The formulas (2.3.6), (2.3.7) together with one of (2.3.8), (2.3.9), and (2.3.10) are called an M-dimensional BPIRK method. It is clear that if M=1, the BPIRK method is reduced to PIRK method discussed in section 2.2.

2.3.2 Generation of the Matrices V_i and W_i

Our concern in this discussion is constrained to Lagrange and Hermite predictors. In particular, we outline an approach for determining the s×M matrices V_i and W_i in Lagrange and Hermite predictors respectively, for any particular i in the range 1,2,..., M.

a) The Lagrange case:

Note that $U_i^{(0)}$ in (2.3.8) is dependent on Y_n; i.e., $U_i^{(0)} = U_i^{(0)}(Y_n)$ where the j^{th} component of Y_n is an approximation to the solution value at the point $(t_{n-1}+hd_j)$. In other words, Y_n is an approximation to the M-vector $y(t_{n-1}e+hd)$ where $d = (d_1, d_2, \ldots, d_M)^T$. Furthermore, $U_i^{(0)}$ itself is intended to serve as an approximation to $y(t_n e + d_i hc) = y(t_{n-1}e + h(d_i c + e))$ (because $t_n = t_{n-1} + h$). If this approximation is to be of order q, then the condition which must be satisfied is:

$$y(t_{n-1}e+h(d_ic+e)) - U_i^{(0)}(\, y(t_{n-1}e+hd)) = O(h^{q+1})$$

i.e.,

$$y(t_{n-1}e+h(d_ic+e)) - V_i\, y\,(t_{n-1}e+hd) = O(h^{q+1}) \qquad (2.3.11)$$

Using the notation in section 1.2, the left-hand side of (2.3.11) can be expanded as following:

$$y(t_{n-1}e+h(d_ic+e)) - V_i\, y\,(t_{n-1}e+hd)$$

$$= \Lambda^q[y,\, t_{n-1}, h(d_i\, c+e)] - V_i\, \Lambda^q[y,\, t_{n-1}, hd]$$

$$= \sum_{j=0}^{q} \frac{1}{j!}\{\, (d_i\, c+e\,)^j h^j \frac{d^j}{dt^j} y(t_{n-1})\} + \frac{1}{(q+1)!}(d_i\, c+e\,)^{q+1} \odot\{\, h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^{*})\,\}$$
$$- V_i\, [\sum_{j=0}^{q} \frac{1}{j!}\{\, d^j\, h^j \frac{d^j}{dt^j} y(t_{n-1})\} - \frac{1}{(q+1)!} d^{q+1} \odot \{h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^{*})\}]$$

$$= \sum_{j=0}^{q} \frac{1}{j!}\{[(d_i\, c+e\,)^j - V_i\, d^j]\, h^j \frac{d^j}{dt^j} y(t_{n-1})\} \;+\; \frac{1}{(q+1)!}(d_i\, c+e\,)^{q+1} \odot\{\, h^{q+1}$$
$$\frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^{*})\,\} - \frac{1}{(q+1)!} V_i d^{q+1} \odot\{h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^{*})\}$$

$$= \sum_{j=0}^{q} \sigma_i^{(j)}\, h^j \frac{d^j}{dt^j} y(t_{n-1}) + \sigma_i^{(q+1)} h^{q+1}$$

where $\sigma_i^{(q+1)} = \frac{1}{(q+1)!}\{\, (d_i\, c+e\,)^{q+1} \odot \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^{*}) - V_i\, d^{q+1} \odot \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^{*})\,\}$

$$\sigma_i^{(j)} = \frac{1}{j!}\; \{(d_i\, c+e\,)^j - V_i\, d^j\,\}, \qquad j=0,1,\ldots,q \qquad (2.3.12)$$

and

t_{1i}^{*} is an s-vector with $t_{1i}^{*} \in [\, t_{n-1}e,\, t_{n-1}e+h\, d_i\, c+he] = [\, t_{n-1}e,\, t_n e+h\, d_i\, c]$,

t_{2i}^{*} is an M-vector with $t_{2i}^{*} \in [\, t_{n-1}e,\, t_{n-1}e+h\, d]$.

Note also that $\sigma_i^{(j)}$ is an s-vector for each j=0,1,...,(q+1).

Using the preceding analysis, (2.3.11) becomes

$$\sum_{j=0}^{q} \sigma_i^{(j)} h^j \frac{d^j}{dt^j} y(t_{n-1}) + \sigma_i^{(q+1)} h^{q+1} = O(h^{q+1}) \qquad (2.3.13)$$

In view of (2.3.13), a necessary condition for the approximation (2.3.8) to be of order q is :

$$\sigma_i^{(j)} = 0, \qquad j=0,1,\ldots,q \qquad (2.3.14)$$

Notice that (2.3.14) provides (q+1)×s linear equations which serve to specify the entries of the coefficient matrix V_i. To facilitate the solution of these equations, we define the s×(q+1) matrix P_i as:

$$P_i = (\, e,\ d_i\, c+e,\ (d_i\, c+e)^2\, ,\ \ldots,\ (d_i\, c+e)^q)$$

and the M×(q+1) matrix Q as:

$$Q = (e, d\, , d^2\, , \ldots, d^q\,)$$

Then, from (2.3.12) and (2.3.14) it follows that:

$$P_i - V_i\, Q = 0 \qquad (2.3.15)$$

We now choose q=M-1 which results in Q being a square matrix. Because of our earlier assumption in section 2.2.1 that the d_i, i=1,2, ...,M, are distinct, it follows that the square matrix Q is nonsingular. Thus (2.3.15) yields:

$$V_i = P_i Q^{-1} \qquad (2.3.16)$$

The above analysis has proved the following theorem:

Theorem 2.1 If the Lagrange predictor (2.3.8) is of order $q=M-1$, then the $s\times M$ matrix V_i satisfies

$$V_i = P_i Q^{-1}$$

b) The Hermite Case:

Following the same reasoning used above, a necessary condition for the Hermite predictor (2.3.9) to be of order q, is :

$$y(t_{n-1}e+h(d_ic+e)) - V_i\, y(t_{n-1}e+hd) - hW_i y'(t_{n-1}e+hd) = O(h^{q+1}) \tag{2.3.17}$$

Using the notation in section 1.3, the left-hand side of (2.3.17) can be expanded as follows :

$$y(t_{n-1}e+h(d_ic+e)) - V_i\, y(t_{n-1}e+hd) - hW_i y'(t_{n-1}e+hd)$$

$$= \Lambda^q[y, t_{n-1}, h(d_i c+e)] - V_i \Lambda^q[y, t_{n-1}, hd] - hW_i \Lambda^{q-1}[\frac{d}{dt} y, t_{n-1}, hd]$$

$$\begin{aligned}
= &\sum_{j=0}^{q} \frac{1}{j!}\{ (d_i c+e)^j h^j \frac{d^j}{dt^j} y(t_{n-1})\} + \frac{1}{(q+1)!}(d_i c+e)^{q+1} \odot\{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^*) \} \\
&- V_i [\sum_{j=0}^{q} \frac{1}{j!}\{d^j h^j \frac{d^j}{dt^j} y(t_{n-1})\} - \frac{1}{(q+1)!} d^{q+1} \odot\{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^*)\}] \\
&- W_i [\sum_{j=0}^{q} \frac{1}{j!}\{j d^{j-1} h^j \frac{d^j}{dt^j} y(t_{n-1})\} - \frac{1}{q!} d^q \odot\{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{3i}^*)\}]
\end{aligned}$$

$$= \sum_{j=0}^{q} \frac{1}{j!}\{[(d_i c+e)^j - V_i d^j + jW_i d^{j-1})] h^j \frac{d^j}{dt^j} y(t_{n-1})\} + \frac{1}{(q+1)!}(d_i c+e)^{q+1}$$

$$\odot\{\, h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{1i}^{*})\,\} - \frac{1}{(q+1)!}V_i d^{q+1}\odot\{h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{2i}^{*})\} - \frac{1}{q!}W_i d^{q}\odot\{\, h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{3i}^{*})\}$$

$$= \sum_{j=0}^{q}\sigma_i^{(j)} h^j\frac{d^j}{dt^j}y(t_{n-1}) + \sigma_i^{(q+1)}h^{q+1}$$

where $\sigma_i^{(q+1)} = \frac{1}{(q+1)!}\{\, (d_i\, c+e\,)^{q+1}\odot\frac{d^{q+1}}{dt^{q+1}}y(t_{1i}^{*}) - V_i\, d^{q+1}\odot\frac{d^{q+1}}{dt^{q+1}}y(t_{2i}^{*}) - (q+1)\, W_i\, d^{q}\odot\frac{d^{q+1}}{dt^{q+1}}y(t_{3i}^{*})\}$

$$\sigma_i^{(j)} = \frac{1}{j!}\,\{(d_i\, c+e\,)^j - V_i\, d^j - jW_i\, d^{j-1}\}, \qquad j=0,1,\ldots,q \tag{2.3.18}$$

and

t_{1i}^{*} is an s-vector with $t_{1i}^{*}\in[\, t_{n-1}e,\, t_{n-1}e+hd_ic+he] = [\, t_{n-1}e,\, t_{n}e+hd_ic]$,

t_{2i}^{*} and t_{3i}^{*} are M-vectors with $t_{2i}^{*}, t_{3i}^{*}\in[\, t_{n-1}e,\, t_{n-1}e+h\, d]$.

In order that (2.3.17) be satisfied, it is necessary that

$$\sigma_i^{(j)} = 0, \qquad j=0,1,\ldots,q \tag{2.3.19}$$

We now choose q=2M-1. To facilitate the analysis, we define
i) the s×M matrix P_i as:

$$P_i = (\, e,\, d_i\, c+e,\, (d_i\, c+e)^2\, ,\, \ldots,\, (d_i\, c+e)^{M-1})$$

ii) the M×M matrices Q and R as:

$$Q = (e,\, d\, ,\, d^2\, ,\, \ldots,\, d^{M-1}\,)$$

$$R = (0,\, e,\, 2d,\, 3d^2,\, \ldots,\, (M-1)d^{M-2})$$

iii) the s×(q-M+1) matrix $\overline{P}_i$ as:

$$\overline{P}_i = ((d_i\, c+e)^M, (d_i\, c+e)^{M+1}, \ldots, (d_i c+e)^q)$$

iv) the M×(q-M+1) matrices $\overline{Q}$ and $\overline{R}$ as:

$$\overline{Q} = (d^M, d^{M+1}, \ldots, d^q)$$

$$\overline{R} = (Md^{M-1}, (M+1)d^M, \ldots, qd^{q-1})$$

Then, from (2.3.18) and (2.3.19) it follows that:

$$P_i - V_iQ - W_iR = 0 \tag{2.3.20}$$

and

$$\overline{P}_i - V_i\, Q - W_i\overline{R} = 0 \tag{2.3.21}$$

Because of our earlier assumption in section 2.2.1 that the d_i, i=1,2,...,M, are distinct, Q is nonsingular. Thus (2.3.20) and (2.3.21) become:

$$V_i = [P_i - W_i\, R]\, Q^{-1} \tag{2.3.22}$$

and

$$W_i\, [RQ^{-1}Q - \overline{R}] = P_iQ^{-1}\, Q - \overline{P}_i \tag{2.3.23}$$

The above analysis has proved the following theorem:

Theorem 2.2 If the Hermite predictor (2.3.9) is of order q≥M, then V_i and W_i satisfy:

$$V_i = [P_i - W_i\, R]\, Q^{-1}$$

and

$$W_i\, [RQ^{-1}\overline{Q} - \overline{R}] = P_iQ^{-1}\overline{Q} - \overline{P}_i$$

The following theorem (see Jackson et al. [25]) gives an important result regarding the order of RK-PC methods. It applies to both Lagrange and Hermite predictors.

Theorem 2.3 If the conditions (2.3.14) and (2.3.15) are satisfied (i.e., the order of the predictor is q) and if the base RK formula (2.3.7) is of order p, then the order of the $U_i^{(m)}$ (regarded as an approximation to the true solutions at $t_n e+d_i hc$), $i=1,2,\dots,M$, is $p_{iter} = q+m+1$ and the order of the BPIRK method (2.3.6), (2.3.7), together with (2.3.8) or (2.3.9) is $r = \min(p, p_{iter})$.

The proof of this theorem needs the concept of Butcher Series which is a concept similar to the Taylor series expansion for real function (see [16] for a comprehensive introduction to Butcher Series).

Note finally that we take the same approach as van der Houwen and Cong [20] for the specification of the parameters d_i , $i=1,\dots, M$, in our experiment. We choose (suppose that $c_i > c_j > 0$ ($i>j$)) :

$$d_1=1, \qquad d_i=1+c_{i-1}, \qquad i=2,\dots,s+1,$$

and

$$\text{if } c_s \neq 1, \quad \text{then} \quad d_i=\frac{s+i}{s+1}, \qquad i=s+2,\dots,M,$$

$$\text{else} \quad d_i=\frac{s+i-1}{s}, \qquad i=s+2,\dots,M.$$

2.4 Implementation of the BPIRK Family of Methods

In this section we outline an implementation of the BPIRK family of methods. The presentation is provided as a sequence of hierarchically organized flowcharts.

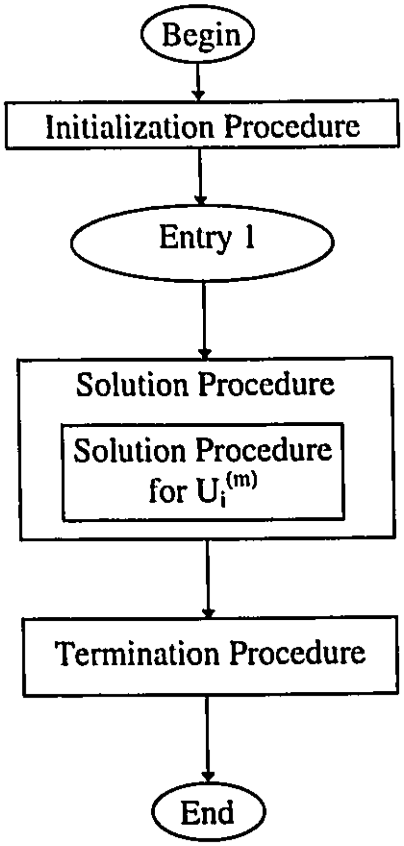


Figure 1 Highlevel View of the BPIRK Solution Process

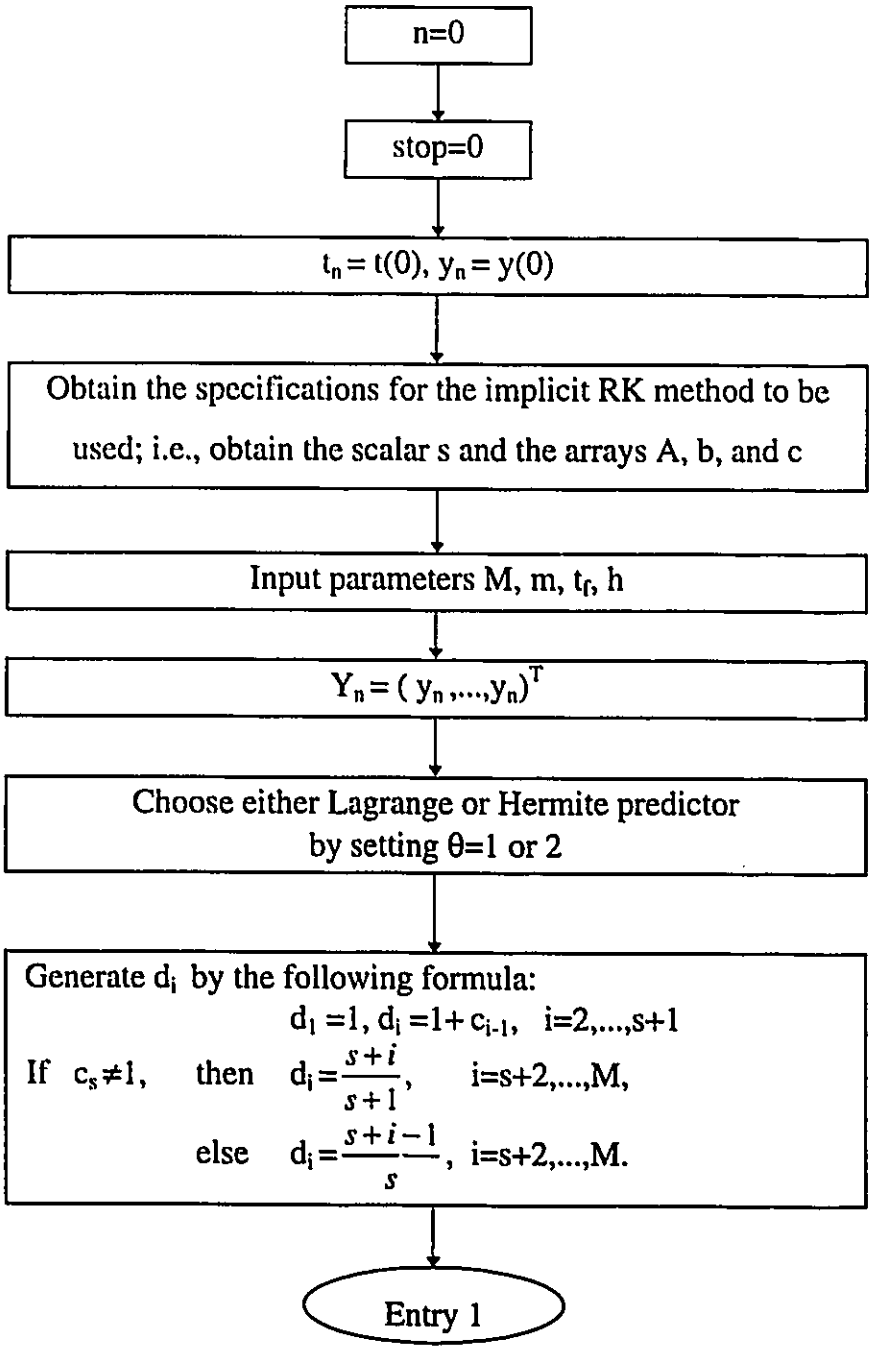


Figure 2(a) Initialization Procedure

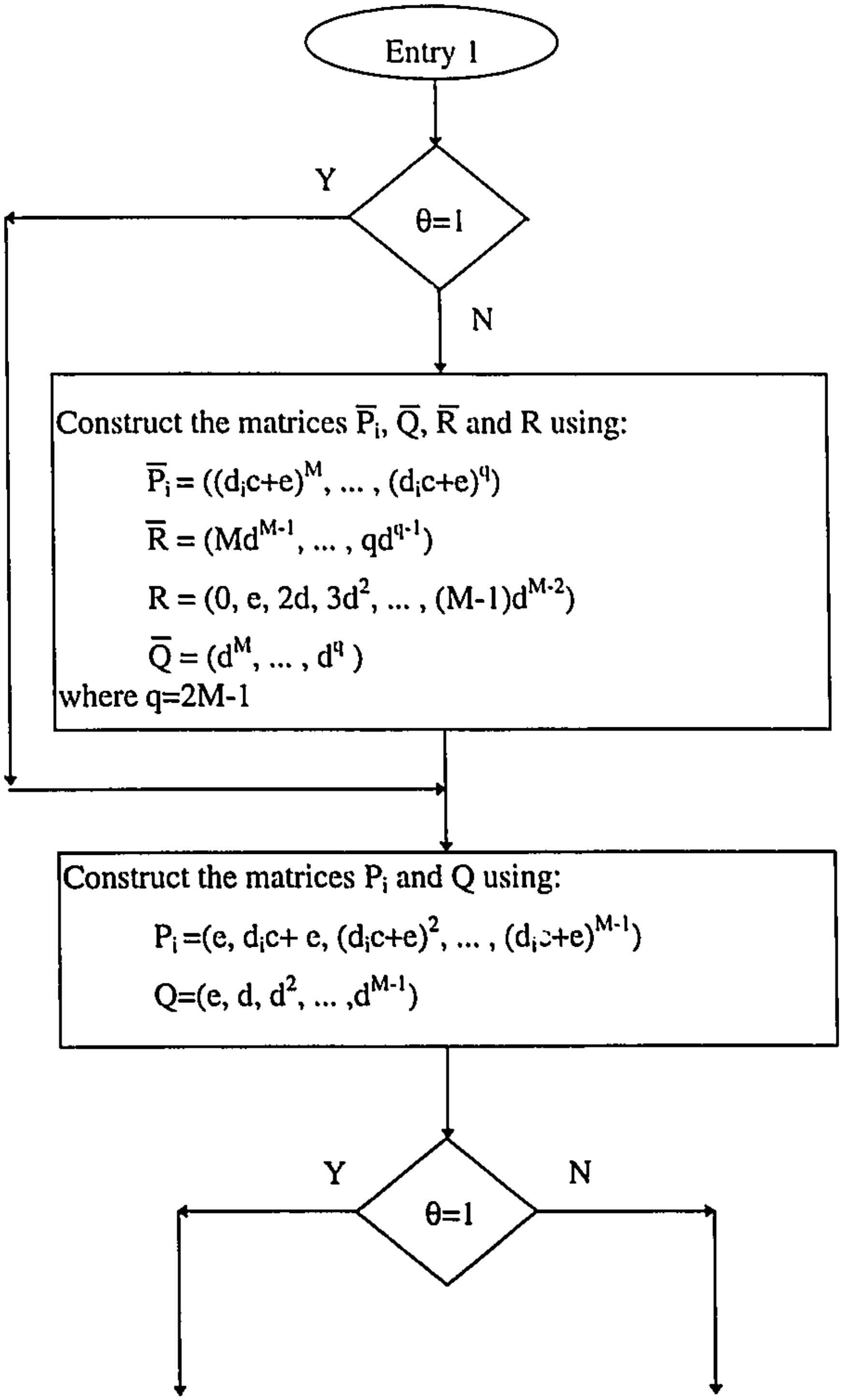


Figure 2(b) Solution Procedure

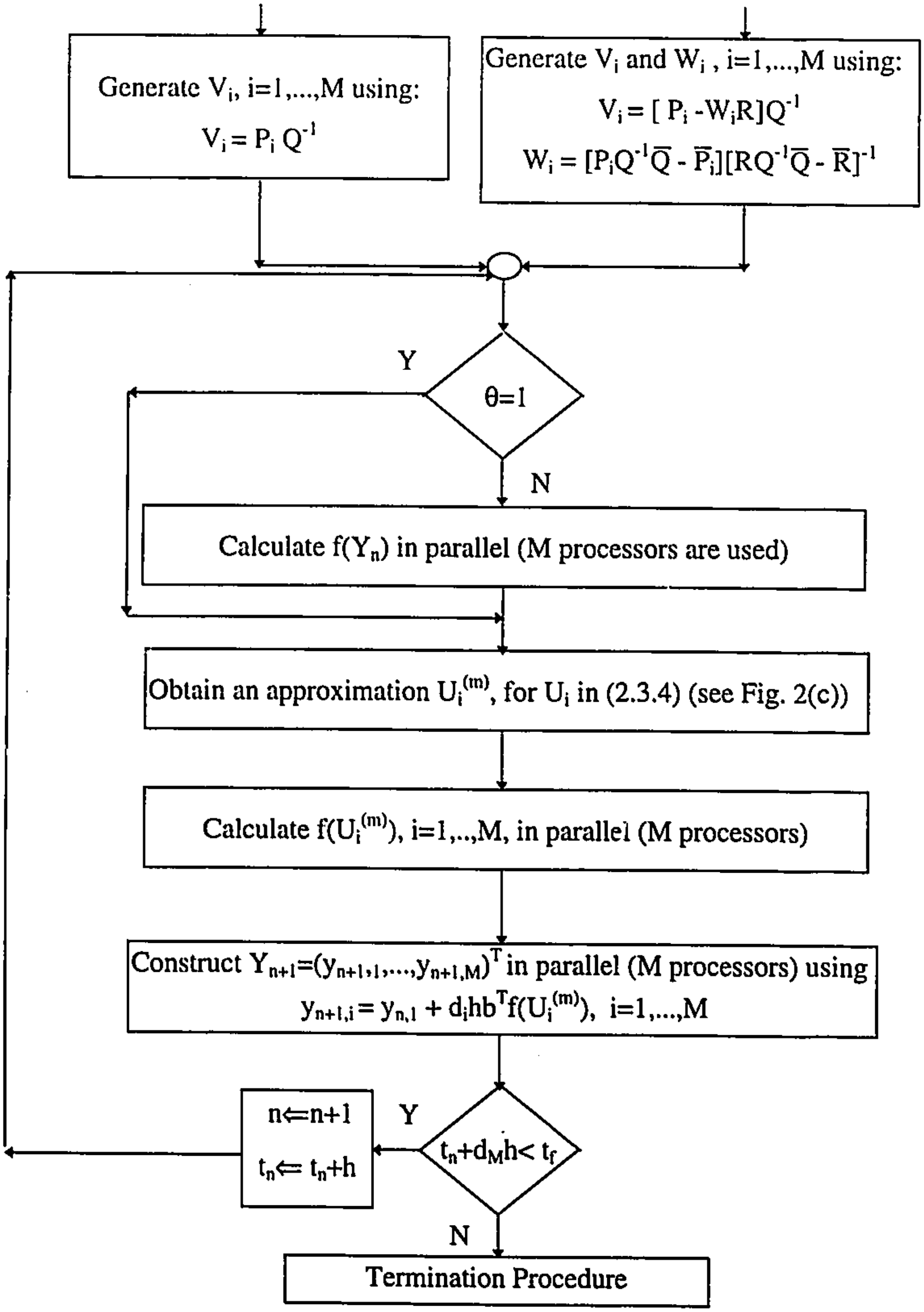


Figure 2(b) (Cont.) Solution Procedure

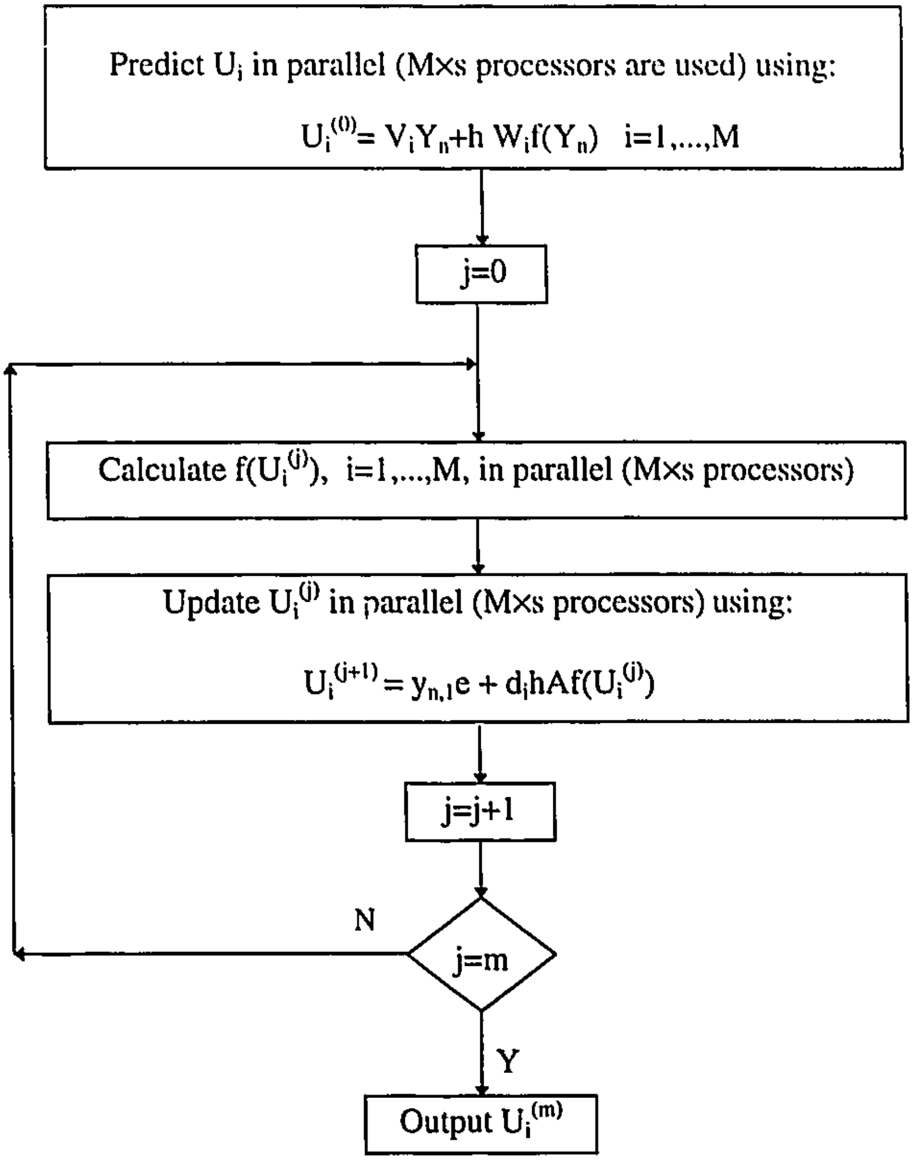


Figure 2(c) Solution Procedure for $U_i^{(m)}$

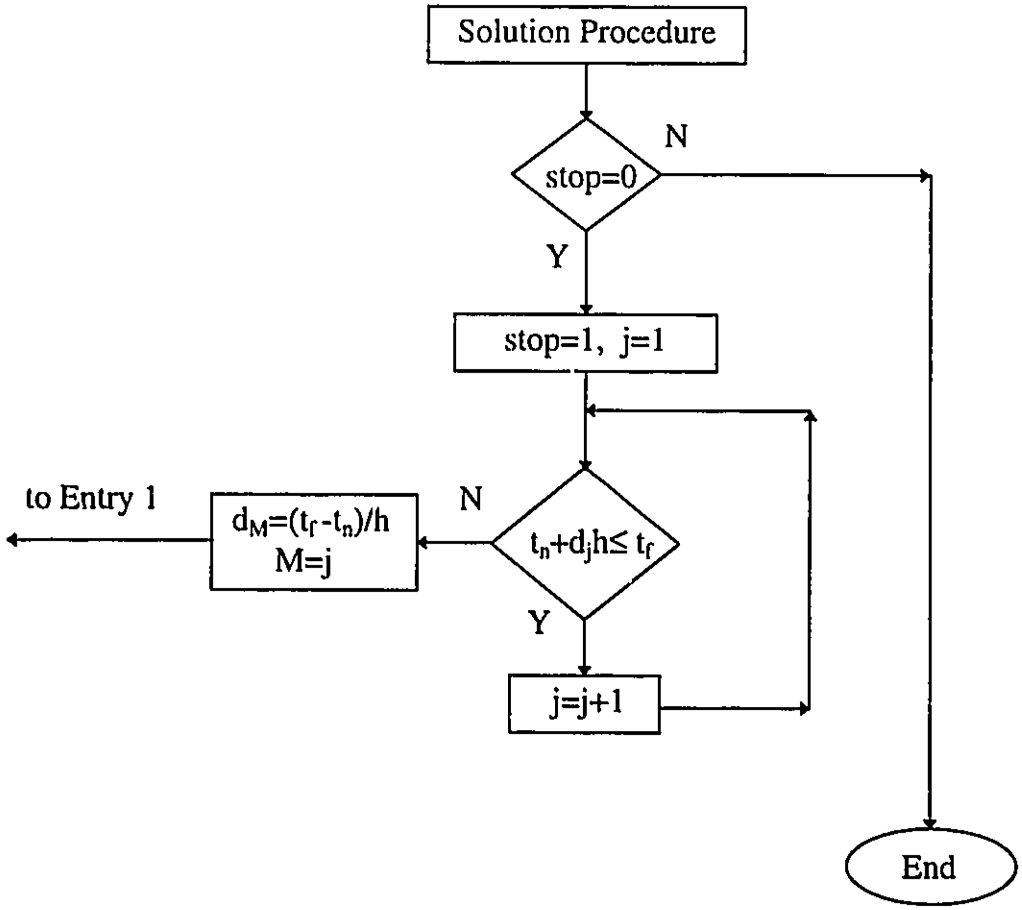
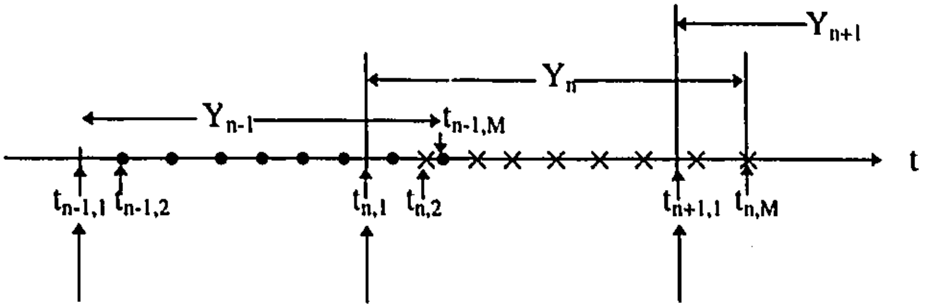


Figure 2(d) Termination Procedure

2.5 Formulation of the BPIRK$'$ Family of Methods

In the BPIRK family of methods as outlined in section 2.3.1, we know that only the point $y_{n+1,1}$ is taken as an output point in the block of points Y_{n+1}. All the other points in Y_{n+1} are treated as intermediate points. They are used only to obtain the predictor in the subsequent step. In this section, we consider making the intermediate points into output points. This provides a new approach to the parallel solution of (1.1.1). To facilitate the understanding of the two families of methods, we give the following configuration in Figures 3 and 4.



base point for the
block of points Y_n and
output point for Y_{n-1}

base point for the
block of points Y_{n+1} and
output point for Y_n

base point for the
block of points Y_{n+2} and
output point for Y_{n+1}

Figure 3 Configuration for BPIRK Family of Methods
where • and × indicate intermediate points for blocks Y_{n-1} and Y_n respectively

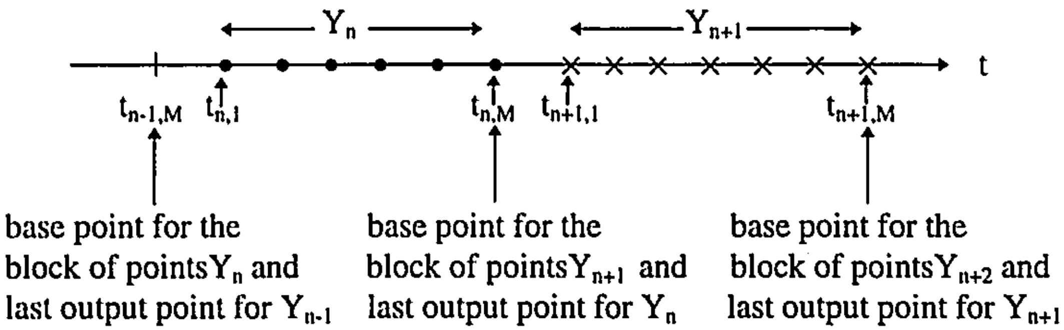


Figure 4 Configuration for BPIRK$^+$ Family of Methods
where • and × indicate output points for blocks Y_n and Y_{n+1} respectively

For the BPIRK$^+$ family, we still use y_n indicating the base point for block Y_{n+1}. Here, $y_n=y_{n,M}$ and the corresponding time point is $t_n=t_{n,M}$.

Now, we formulate the approach to make the intermediate points, $y_{n+1,i}$, $i=2,\dots,M$, into output points. The formulation proceeds by first modifying the relationships in (2.3.4), (2.3.6) and (2.2.7). More specifically, letting $y_{n,M}$ and $t_{n,M}$ replace y_n and t_n respectively, we have

$$U_i = y_{n,M}e+h_iAf(t_{n,M}e+h_ic,\ U_i\ ) \tag{2.5.1}$$

$$U_i^{(j)} = y_{n,M}e+h_iAf(t_{n,M}e+h_ic,\ U_i^{(j-1)}\ ),\ j=1,2,\dots,m \tag{2.5.2}$$

and

$$y_{n+1,i} = y_{n,M}+h_i b^Tf\,(t_{n,M}e+h_ic,\ U_i^{(m)})\,,\quad i= 1,2,\dots,M \tag{2.5.3}$$

The formulas (2.5.2) and (2.5.3) together with a predictor for $U_i^{(0)}$ are called an M-dimensional BPIRK$^+$ family of methods.

Here we take the same predictor formulas (2.3.8) and (2.3.9) given in section 2.3; i.e.,

$$\text{Lagrange:}\quad U_i^{(0)} = V_i^+ Y_n \tag{2.5.4}$$

$$\text{Hermite:}\quad U_i^{(0)} = V_i^+ Y_n + hW_i^+ f(T_n, Y_n) \tag{2.5.5}$$

However, we use V_i^+ and W_i^+ instead of V_i and W_i for the BPIRK family. Now, we derive the matrices V_i^+ and W_i^+ for BPIRK$^+$ family.

a) The Lagrange case:

Here $U_i^{(0)}$ is intended to serve as an approximation to $y(t_n e + d_i hc) = y(t_{n-1}c + h(d_i c + ed_M))$ (because $t_n = t_{n-1} + h_M = t_{n-1} + hd_M$). If (2.5.4) is to be of order q, then the condition which must be satisfied is:

$$y(t_{n-1}e + h(d_i c + ed_M)) - V_i^+ y(t_{n-1}e + hd) = O(h^{q+1}) \tag{2.5.6}$$

Using the notation in section 1.2, the left-hand side of (2.5.6) can be expanded as following:

$$y(t_{n-1}e + h(d_i c + ed_M)) - V_i^+ y(t_{n-1}e + hd)$$

$$= \Lambda^q[y, t_{n-1}, h(d_i c + ed_M)] - V_i^+ \Lambda^q[y, t_{n-1}, hd]$$

$$= \sum_{j=0}^{q} \frac{1}{j!}\{ (d_i c + ed_M)^j h^j \frac{d^j}{dt^j} y(t_{n-1})\} + \frac{1}{(q+1)!}(d_i c + e)^{q+1} \odot \{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^*)\}$$
$$- V_i^+ [\sum_{j=0}^{q} \frac{1}{j!}\{ d^j h^j \frac{d^j}{dt^j} y(t_{n-1})\} - \frac{1}{(q+1)!} d^{q+1} \odot \{h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^*)\}]$$

$$= \sum_{j=0}^{q} \frac{1}{j!} \{[(d_i c+ed_M)^j - V_i d^j]\, h^j \frac{d^j}{dt^j} y(t_{n-1})\} + \frac{1}{(q+1)!} (d_i c+e)^{q+1} \odot \{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^{*}) \} - \frac{1}{(q+1)!} V_i^{+} d^{q+1} \odot \{ h^{q+1} \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^{*}) \}$$

$$= \sum_{j=0}^{q} \sigma_i^{(j)} h^j \frac{d^j}{dt^j} y(t_{n-1}) + \sigma_i^{(q+1)} h^{q+1}$$

where $\sigma_i^{(q+1)} = \frac{1}{(q+1)!} \{ (d_i c+ed_M)^{q+1} \odot \frac{d^{q+1}}{dt^{q+1}} y(t_{1i}^{*}) - V_i^{+} d^{q+1} \odot \frac{d^{q+1}}{dt^{q+1}} y(t_{2i}^{*}) \}$

$$\sigma_i^{(j)} = \frac{1}{j!} \{ (d_i c+ed_M)^j - V_i^{+} d^j \}, \qquad j=0,1,\ldots,q$$

and

t_{1i}^{*} is an s-vector with $t_{1i}^{*} \in [\, t_{n-1}e,\ t_{n-1}e+hd_i c+hed_M] = [\, t_{n-1}e,\ t_n e+hd_i c]$,

t_{2i}^{*} is an M-vector with $t_{2i}^{*} \in [\, t_{n-1}e,\ t_{n-1}e+hd]$.

Since the necessary condition for the approximation (2.5.4) to be of order q is:

$$\sigma_i^{(j)} = 0, \qquad j=0,1,\ldots,q$$

that is

$$\frac{1}{j!} \{ (d_i c+ed_M)^j - V_i^{+} d^j \} = 0, \qquad j=0,1,\ldots,q \tag{2.5.7}$$

To facilitate the solution of (2.5.7), we define the s×(q+1) matrix P_i^{+} as:

$$P_i^{+} = (\, e,\ d_i c+ed_M,\ (d_i c+ed_M)^2,\ \ldots,\ (d_i c+ed_M)^q)$$

and the M×(q+1) matrix Q is defined the same as in section 2.3, i.e.,

$$Q = (e, d, d^2, \ldots, d^q)$$

Let q=M-1. Then the necessary condition (2.5.7) leads to :

$$V_i^+ = P_i Q^{-1}$$

b) The Hermite case:

Here $U_i^{(0)}$ is intended to serve as an approximation to $y(t_n c+d_i hc) = y(t_{n-1}c+h(d_i c+ed_M))$ (because $t_n=t_{n-1}+h_M=t_{n-1}+hd_M$). By the same reason as for (2.3.11) in section 2.3.2, if we substitute $U_i^{(0)}$ by (2.5.5), for each i, i=1,2,...,M, then we get:

$$y(t_{n-1}e+h(d_i c+ed_M)) - V_i^+ y(t_{n-1}e+hd) - hW_i^+ y'(t_{n-1}e+hd) = O(h^{q+1}) \quad (2.5.8)$$

Applying the same procedure to (2.5.8) as for (2.3.17) yields

$$y(t_{n-1}e+h(d_i c+ed_M)) - V_i^+ y(t_{n-1}e+hd) - hW_i^+ y'(t_{n-1}e+hd)$$

$$= \Lambda[y, t_{n-1}, h(d_i c+ed_M)]^q - V_i^+\Lambda[y, t_{n-1}, hd]^q - hW_i^+\frac{d}{dt}\Lambda[y, t_{n-1}, hd]^q$$

$$= \sum_{j=0}^{q}\frac{1}{j!}\{ [(d_i c+ed_M)^j - (V_i^+ d^j + jW_i^+ d^{j-1})]h^j\frac{d^j}{dt^j}y(t_{n-1})\}$$
$$\frac{1}{(q+1)!}(d_i c+ed_M)^{q+1}\odot\{ h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{1i}^*)\} - \frac{1}{(q+1)!}V_i^+d^{q+1}\odot\{$$
$$h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{2i}^*)\} - \frac{1}{q!}W_i^+d^q\odot\{ h^{q+1}\frac{d^{q+1}}{dt^{q+1}}y(t_{3i}^*)\}$$

$$= \sum_{j=0}^{q}\sigma_i^{(j)} h^j\frac{d^j}{dt^j}y(t_{n-1}) + \sigma_i^{(q+1)}h^{q+1}$$

where $\sigma_i^{(q+1)} = \frac{1}{(q+1)!}\{ (d_i c+ed_M)^{q+1}\odot\frac{d^{q+1}}{dt^{q+1}}y(t_{1i}^*) - V_i^+ d^{q+1}\odot\frac{d^{q+1}}{dt^{q+1}}y(t_{2i}^*) -$

$$(q+1)\ W_i^+\ d^q \odot \frac{d^{q+1}}{dt^{q+1}} y(t_{3i}^*)\}$$

$$\sigma_i^{(j)} = \frac{1}{j!}\ \{(d_i\ c+ed_M\ )^j - V_i^+\ d^j\ -jW_i^+\ d^{j-1}\}, \qquad j=0,1,\ldots,q,$$

and

t_{1i}^* is an s-vector with $t_{1i}^* \in [\ t_{n-1}e,\ t_{n-1}e+h\ d_i\ c+hed_M] = [\ t_{n-1}e,\ t_ne+hd_ic]$,

t_{2i}^* and t_{3i}^* are M-vectors with $t_{2i}^*,\ t_{3i}^* \in [\ t_{n-1}e,\ t_{n-1}e+h\ d]$

The necessary condition for the approximation (2.5.8) to be of order q is:

$$\sigma_i^{(j)} = 0, j=0,1,\ldots,q$$

that is

$$\frac{1}{j!}\ \{(d_ic+ed_M\ )^j - V_i^+d^j\ \}=0, \qquad j=0,1,\ldots,q \tag{2.5.9}$$

To facilitate the solution of (2.5.9), we define :

i) the s×M matrix P_i^+ as:

$$P_i^+ = (\ e,\ d_ic+d_Me,\ (d_ic+d_Me)^2\ ,\ \ldots,\ (d_ic+d_Me)^{M-1})$$

ii) the M×M matrices Q and R as:

$$Q = (e,\ d\ ,\ d^2,\ \ldots,\ d^{M-1}\ )$$

$$R = (0,\ e,\ 2d,\ 3d^2,\ \ldots,\ (M-1)d^{M-2})$$

iii) the s×(q-M+1) matrix $\bar{P}_i^+$ as:

$$\overline{P}_i^+ = ((d_i\, c+d_M e)^M, (d_i\, c+d_M e)^{M+1}, \ldots, (d_i c+d_M e)^q)$$

iv) the M×(q-M+1) matrices $\overline{Q}$ and $\overline{R}$ as:

$$\overline{Q} = (d^M, d^{M+1}, \ldots, d^q)$$

$$\overline{R} = (Md^{M-1}, (M+1)d^M, \ldots, qd^{q-1})$$

The necessary condition (2.5.9) leads to :

$$P_i^+ - V_i^+Q - W_i^+R = 0$$

$$\overline{P}_i^+ - V_i^+\overline{Q} - W_i^+\overline{R} = 0$$

Let q=2M-1. Then the matrices V_i^+ and W_i^+ can be generated by the following equations:

$$V_i^+ = [P_i - W_i\, R]\, Q^{-1}$$

$$W_i^+[RQ^{-1}\overline{Q} - \overline{R}] = P_iQ^{-1}\overline{Q} - \overline{P}_i$$

Note that the definitions of matrices Q, R, $\overline{Q}$, $\overline{R}$ are the same as in section 2.3.2. Differences do occur, however, in the definitions of P_i and P_i^+.

Theorems 2.3 of section 2.3.2 apply equally to the BPIRK$^+$ family of methods.

Chapter 3
Absolute Stability Bounds for the BPIRK and BPIRK$^+$ Families

3.1 Stability Matrices for the BPIRK and BPIRK$^+$ Families

Stability analysis is based on the special case of (1.1.1) where $y'=\lambda y$ (where y is a scalar, see notation in section 1.2). Since the Lagrange predictor (i.e., $W_i=0$, $q=M-1$) is a special case of the Hermite predictor, our analysis in this section assumes the use of the Hermite predictor.

i) Stability Matrix for the BPIRK Family

Using the definitions for E_l and e_l given in section 1.2, equations (2.3.4), (2.3.6) and (2.3.7) (for any particular i in the range 1 through M) can be written as:

$$U_i = E_l Y_n + h_i A f(t_n e + h_i c, U_i) \tag{3.1.1}$$

$$U_i^{(j)} = E_l Y_n + h_i A f(t_n e + h_i c, U_i^{(j-1)}), \quad j=1,2,\ldots,m \tag{3.1.2}$$

$$y_{n+1,i} = e_l^T Y_n + h_i b^T f(t_n e + h_i c, U_i^{(m)}) \tag{3.1.3}$$

For the particular case of interest; namely, $f(t, y)= \lambda y$, (3.1.1) becomes:

$$U_i = E_l Y_n + h_i \lambda A U_i \tag{3.1.4}$$

i.e.

$$U_i = [I-h_i\lambda A]^{-1} E_l Y_n \tag{3.1.5}$$

where I is the s×s identity matrix. Similarly, (3.1.2), (3.1.3) and (2.3.9) become:

$$U_i^{(j)} = E_1 Y_n + h_i \lambda A U_i^{(j-1)}, \qquad j=1,2,\ldots,m \tag{3.1.6}$$

$$y_{n+1,i} = e_1^T Y_n + h_i \lambda b^T U_i^{(m)} \tag{3.1.7}$$

and

$$U_i^{(0)} = (V_i + h\lambda W_i) Y_n \tag{3.1.8}$$

Recall that $h_i = d_i h$, and let $z = \lambda h$. Subtracting (3.1.6) from (3.1.4), yields

$$U_i^{(j)} - U_i = d_i z A (U_i^{(j-1)} - U_i) \qquad j=1,2,\ldots,m$$

i.e.

$$U_i^{(m)} - U_i = (d_i z A)^m (U_i^{(0)} - U_i) \tag{3.1.9}$$

Also (3.1.5) can be written as:

$$U_i = [I - d_i z A]^{-1} E_1 Y_n \tag{3.1.10}$$

Then from (3.1.7), we have

$$y_{n+1,i} = e_1^T Y_n + d_i z b^T U_i^{(m)}$$

$$= e_1^T Y_n + d_i z b^T U_i + d_i z b^T (U_i^{(m)} - U_i)$$

$$= (e_1^T + d_i z b^T [I - d_i z A]^{-1} E_1) Y_n + d_i z b^T (d_i z A)^m (V_i + h\lambda W_i - [I - d_i z A]^{-1} E_1) Y_n$$

$$= (e_1^T + d_i z b^T [I - d_i z A]^{-1} E_1 + d_i z b^T (d_i z A)^m (V_i + h\lambda W_i - [I - d_i z A]^{1} E_1)) Y_n \tag{3.1.11}$$

Define the M×M matrix R(z) as:

$$R(z) = \begin{bmatrix} R_1^T(z) \\ \vdots \\ R_i^T(z) \\ \vdots \\ R_M^T(z) \end{bmatrix}$$

where $R_i^T(z) = (e_1^T + d_i z b^T[I-d_i zA]^{-1}E_1 + d_i z b^T(V_i+h\lambda W_i - [I-d_i zA]^{-1}E_1))$. Then

$$y_{n+1,i} = R_i^T(z)Y_n$$

and furthermore

$$Y_{n+1} = R(z)Y_n$$

Note, in particular, the dependence of R(z) on the parameter m.

On the basis of the discussion in section 1.2, R(z) is the stability matrix for the BPIRK family of methods.

ii) Stability Matrix for BPIRK$^+$ Family

Using the definitions for E_M and e_M in section 1.2, equations (2.5.1), (2.5.2) and (2.5.3) (for any particular i in the range 1 through M) can be written as:

$$U_i = E_M Y_n + h_i Af(t_{n,M}e+h_i c,\ U_i) \tag{3.1.12}$$

$$U_i^{(j)} = E_M Y_n + h_i Af(t_{n,M}e+h_i c,\ U_i^{(j-1)}), \qquad j=1,2,\ldots,m \tag{3.1.13}$$

$$y_{n+1,i} = e_M^T Y_n + h_i b^T f(t_{n,M}e+h_i c,\ U_i^{(m)}) \tag{3.1.14}$$

Algebraic manipulations similar to those used above, yield the following counterpart to equation (3.1.11):

$$y_{n+1,i} = (e_M{}^T + d_i z b^T [I - d_i z A]^{-1} E_M + d_i z b^T (d_i z A)^m (V_i{}^+ + h\lambda W_i{}^+ - [I - d_i z A]^1 E_M)) Y_n \tag{3.1.15}$$

We define the M×M matrix $R^+(z)$ as:

$$R^+(z) = \begin{bmatrix} (R_1{}^+(z))^T \\ \vdots \\ (R_i{}^+(z))^T \\ \vdots \\ (R_M{}^+(z))^T \end{bmatrix}$$

where $(R_i{}^+(z))^T = e_M{}^T + d_i z b^T [I - d_i z A]^{-1} E_M + d_i z b^T (d_i z A)^m (V_i{}^+ + h\lambda W_i{}^+ - [I - d_i z A]^{-1} E_M)$. Then

$$y_{n+1,i} = (R_i{}^+(z))^T Y_n$$

and furthermore

$$Y_{n+1} = R^+(z) Y_n$$

where $R^+(z)$ is the stability matrix for the BPIRK$^+$ family of methods.

3.2 Stability Bounds for the BPIRK and BPIRK$^+$ Families

The definition of stability bound is given in section 1.2. In this section we present the results of an investigation of stability bounds for the BPIRK and BPIRK$^+$ families of methods.

Of special interest in this evaluation is a comparison of the stability bounds for the BPIRK and BPIRK$^+$ approaches with those of the BPC and PBPC parallel methods. Some results for the latter are provided in Figures 7, 8, 11, 12, 15, and 16 (these have been taken from Table 1 in [1]). Throughout this discussion, we use B to denote the stability bound and N to denote the number of processors used in a particular method.

Figures 5, 6, 9, 10, 13, and 14 display the stability bounds for the BPIRK and BPIRK$^+$ families of methods. These figures focus on the parameter r, the order of the methods.

Theorem 2.3 provides the basis for establishing the order of the BPIRK and BPIRK$^+$ families of methods; i.e., $r=\min(p, p_{iter})$ where p is the order of the underlying RK (implicit) formula that is used and $p_{iter}=q+m+1$ where q is the order of the predictor and m is the number of iterations used in solving (2.3.4) (see (2.3.6)). In the case of the Lagrange predictor, q is set at M-1. Note also that the underlying RK formulas that have been used are of orders p=4, 6, 8, and 10 and for each case the stage values s=p/2 (see Appendix I). The order relationship can therefore be written as :

$$r = \min(2s, M+m)$$

To achieve a third-order BPIRK or BPIRK$^+$ method (i.e., r=3), it must be that either :

a) $2s=3$ and $M+m\geq 3$

or

b) $2s\geq 3$ and $M+m=3$

For each of the RK formulas used, $2s>3$ and consequently case b) holds; i.e.,

$M+m=3$ for s=2, 3, 4 and 5

To achieve a fifth-order BPIRK or BPIRK$^+$ method (i.e., r=5), it must be that either :

a) $2s=5$ and $M+m\geq 5$

or

b) $2s \geq 5$ and $M+m=5$

The choice $s=2$ is not admissible and case b) holds for $s=3$, 4 and 5. In other words, for the RK formulas used, $r=5$ corresponds to :

$M+m=5$ for $s=3$, 4 and 5

From similar reasoning it follow that seventh-order BPIRK and BPIRK$^+$ methods are possible only for the $s=4$ and 5 RK formulas that were used and the underlying relationship is :

$M+m=7$ for $s=4$ and 5

Table 1 and 2 provide stability bounds for the BPIRK approach for the cases of the Lagrange and Hermite predictors respectively. Results for various values of s, m and M are given. A number of observations can be made:

i) With certain exceptions, for corresponding values of s, m, and M, the stability bound for the Lagrange case is larger than for the Hermite case. This difference can be substantial; e.g., 131% for the case $s=5$, $m=10$, $M=4$. The exceptional cases all occur for $M=1$ (the PIRK method) where for most (but not all) values of s and m, the stability bound for the Lagrange case is smaller than for the Hermite case;

ii) Generally, the stability bound decreases monotonously as M increases for any combination of values for s and M. Some exceptions do occur in the Hermite case for $M=3$ (e.g., $s=3$, $m=2$; $s=4$, $m=2$ and $m=4$; $s=5$, $m=2$ and $m=4$);

iii) Generally, the stability bound increases monotonously as m increases, for any particular combination of values for s and M. Again there are exceptions and these occur in both the Lagrange and Hermite cases. In both cases they occur for $s=2$ ($M=1$ and 2 for the Lagrange case and $M=1$, 3 and 4 for the Hermite case).

From Figure 5 through Figure 16, it is clear that the stability bounds of the BPIRK and BPIRK$^+$ families are much larger than those of the BPC and PBPC families. Also, Figures 5, 6, 9, 10, 13, and 14 show that the stability bounds of the BPIRK$^+$ family are larger than those of the BPIRK family; in other words the BPIRK$^+$ family of methods is more stable.

We note also (from Figures 5, 6, 9, 10, 13, and 14) that

i) For any particular order, r, the stability bound remains relatively constant as s changes;

ii) For any particular order, r, the stability bound decreases as M increases (with one exception, Figure 6 with s=3);

iii) Larger values of the order, r, generally provide larger stability bounds.

In Figures 5 through 16, we use the following notation:

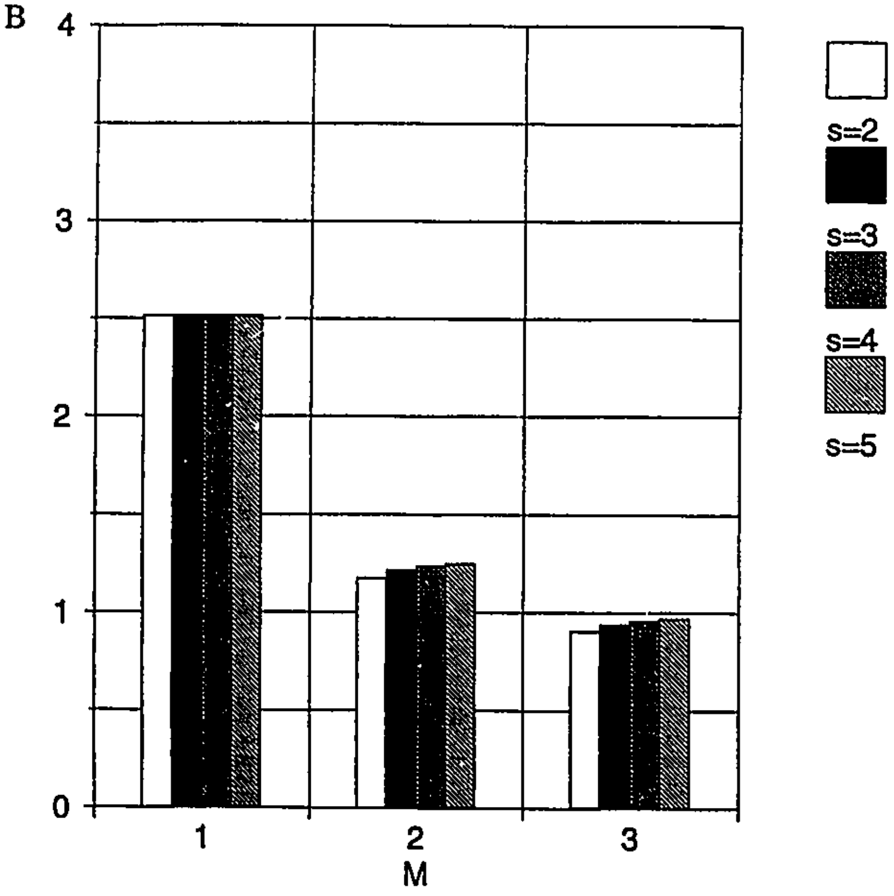
B : Stability bound
N : Number of processors used (in BPIRK and BPIRK$^+$ cases, N=sM)
M : Number of points in one block
m : Number of iterations in one step
s : Stage value of RK method

s	m	M=1	M=2	M=3	M=4	M=5	M=6
2	2	2.51278	1.54001	1.32986	1.26756	1.17529	1.09568
	4	3.54865	3.5508	1.52995	1.18742	1.01556	0.902626
	10	3.46412	3.47268	1.70857	1.37832	1.20547	1.05949
3	2	2.51277	1.596	1.32461	1.22625	1.19859	1.1845
	4	3.21717	2.36828	2.03835	1.80669	1.63462	1.48284
	10	3.89936	3.10719	2.44658	1.99924	1.77818	1.57821
4	2	2.51271	1.62058	1.35851	1.24561	1.19585	1.17113
	4	3.21694	2.4085	2.11775	1.99073	1.94569	1.9137
	10	5.5034	4.30062	3.36842	2.68856	2.24455	1.97832
5	2	2.51276	1.6334	1.37843	1.26141	1.20518	1.1755
	4	3.21698	2.42925	2.15432	2.01919	1.9538	1.92086
	10	5.43419	4.67941	4.25544	3.72819	3.08761	2.59236

Table 1 Stability Bounds for BPIRK Family with Lagrange Predictor

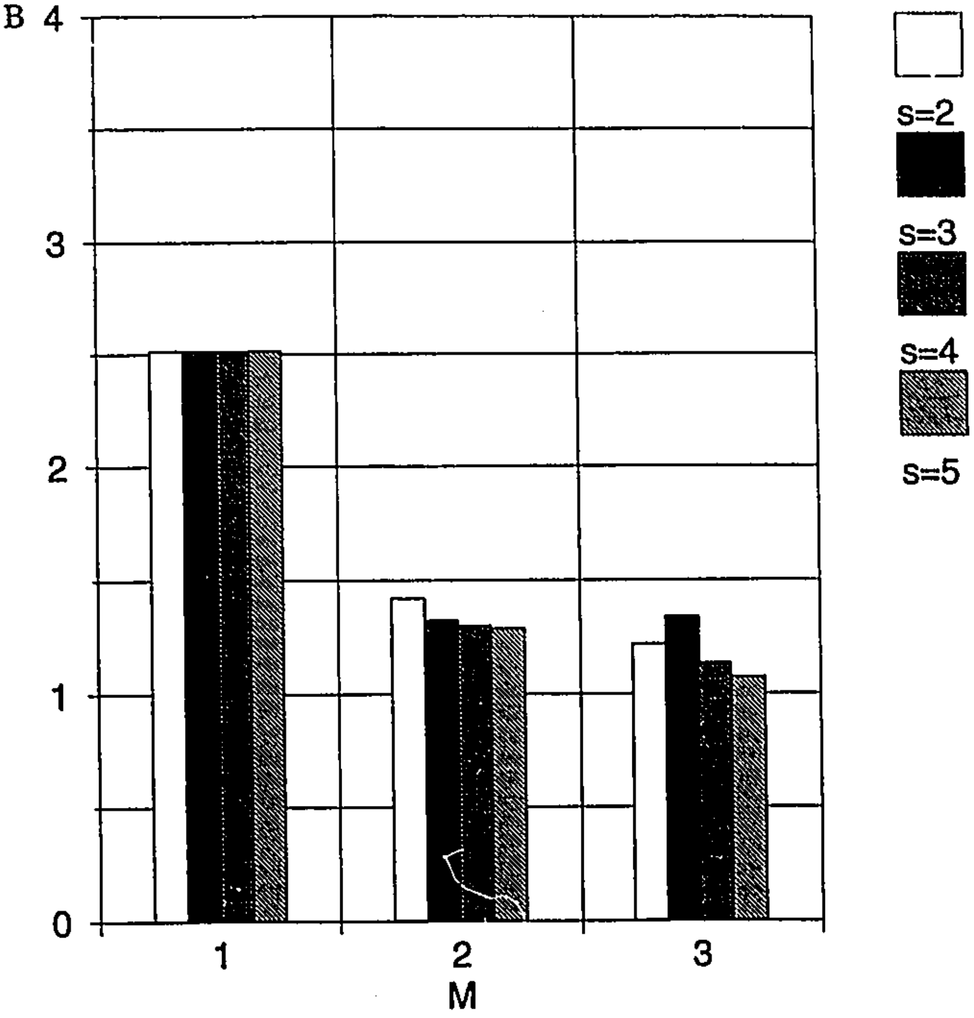
s	m	M=1	M=2	M=3	M=4	M=5	M=6
2	2	2.78536	1.18635	1.15615	0.80761	0.786751	0.0357437
	4	3.54787	1.51881	1.51881	0.789909	0.79644	0.122185
	10	3.46366	2.1146	1.48415	1.14874	0.985085	0.440544
3	2	2.78534	0.940413	1.2013	0.936852	0.896553	0.50666
	4	3.55366	1.84229	1.72424	1.31777	1.01554	0.772087
	10	3.93066	2.67907	2.1585	1.67902	1.64558	1.32145
4	2	2.78526	0.832481	1.16819	0.813225	0.876396	0.50602
	4	3.55334	1.73376	2.09032	1.70225	1.50797	1.39686
	10	4.98839	3.36915	2.84367	2.11878	1.64558	1.5111
5	2	2.78526	0.749306	0.85807	0.681557	0.897118	0.724144
	4	3.5532	1.62954	2.05944	1.60509	1.50011	1.30556
	10	5.74903	4.04093	3.90209	2.82995	2.14703	1.67122

Table 2 Stability Bounds for BPIRK Family with Hermite Predictor



M	m	s=2	s=3	s=4	s=5
1	2	2.51278	2.51277	2.51271	2.51276
2	1	1.16973	1.21418	1.23443	1.24517
3	0	0.90287	0.936155	0.955925	0.967461

Figure 5 Stability Bounds for BPIRK Family with Lagrange Predictor and Order 3 (N=sM, m=3-M)



M	m	s=2	s=3	s=4	s=5
1	2	2.51278	2.51277	2.51271	2.51276
2	1	1.417	1.32469	1.29861	1.28733
3	0	1.21784	1.33817	1.13456	1.07136

Figure 6 Stability Bounds for BPIRK$^+$ Family with Lagrange Predictor and Order 3 (N=sM, m=3-M)

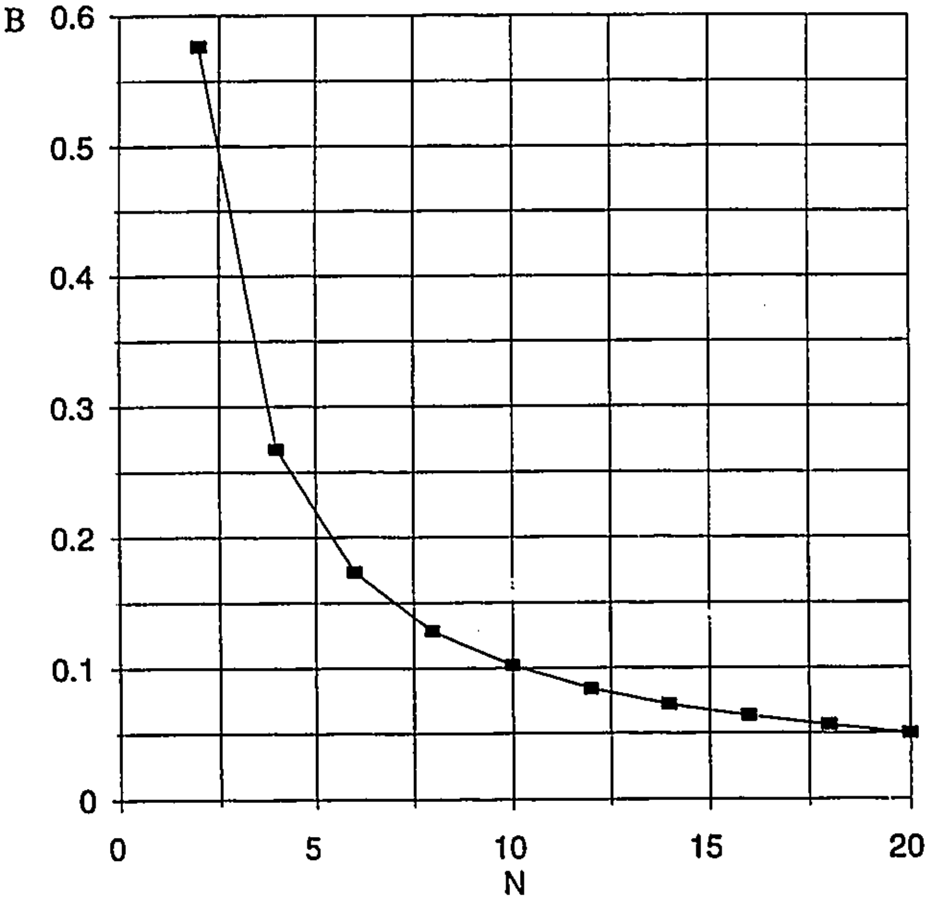


Figure 7 Stability Bounds for BPC Methods of Order 3

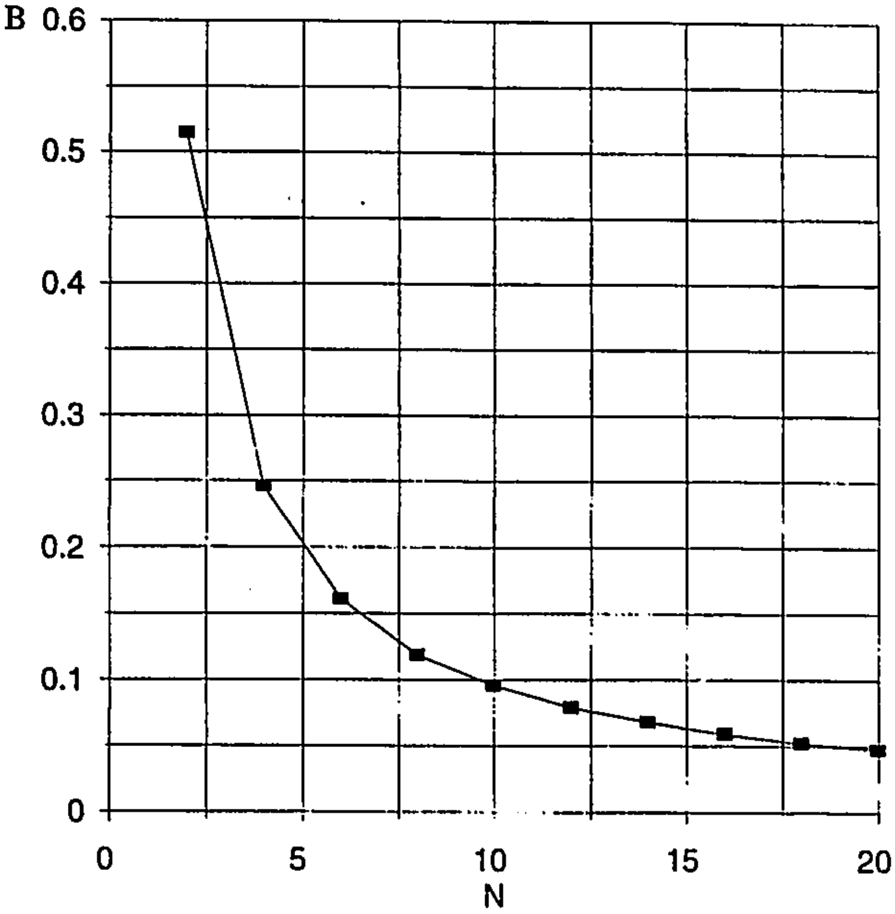
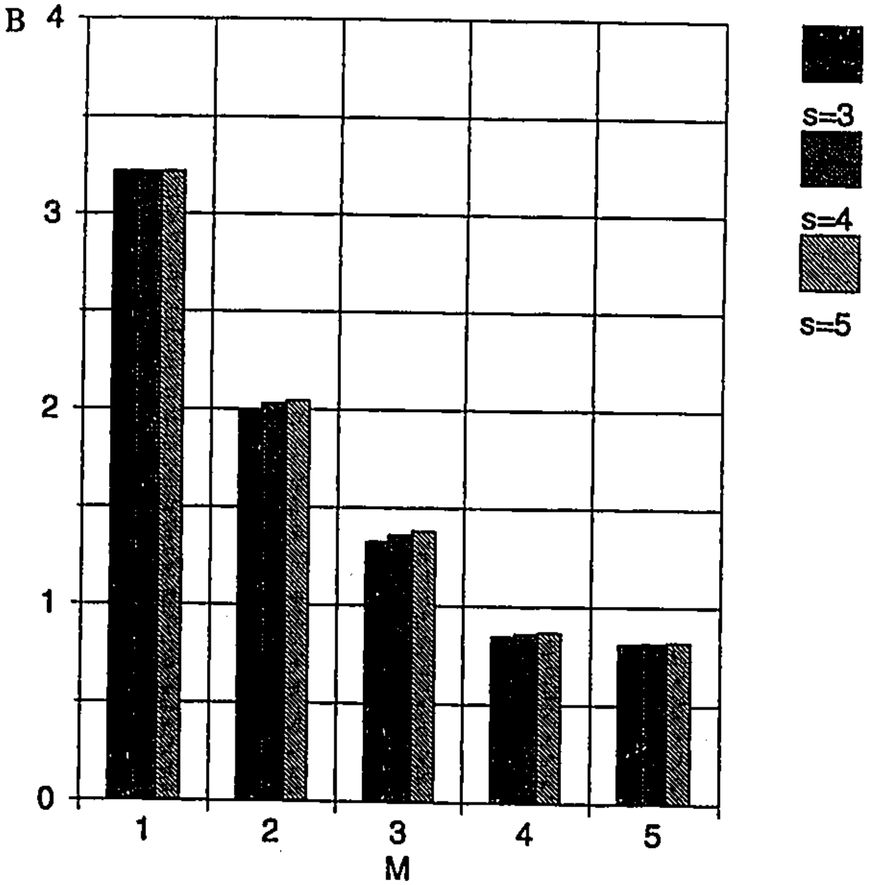
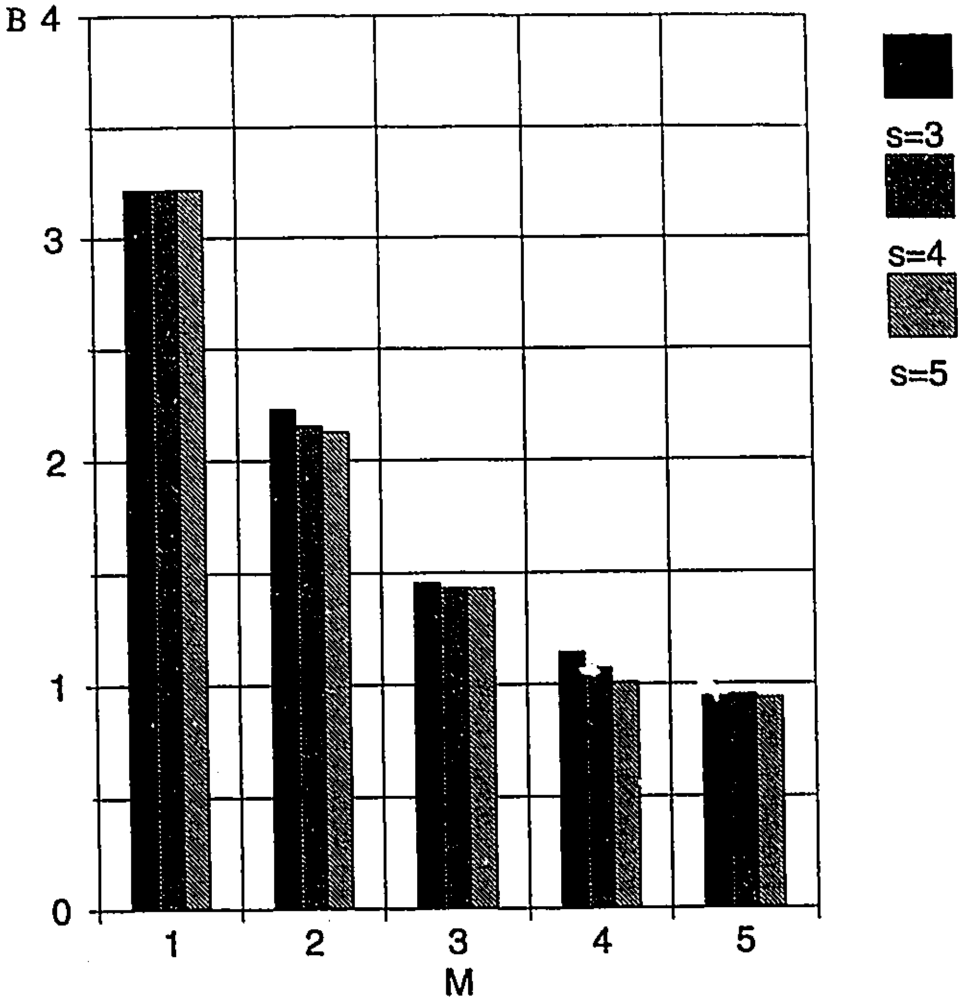


Figure 8 Stability Bounds for PBPC Methods of Order 3



M	m	s=3	s=4	s=5
1	4	3.21717	3.21694	3.21698
2	3	1.99375	2.0279	2.04575
3	2	1.32461	1.35851	1.37843
4	1	0.847054	0.85923	0.868462
5	0	0.810175	0.815178	0.820808

Figure 9 Stability Bounds for BPIRK Family with Lagrange Predictor and Order 5 (N=sM, m=3-M)



M	m	s=3	s=4	s=5
1	4	3.21717	3.21694	3.21698
2	3	2.225	2.15333	2.12594
3	2	1.4548	1.43299	1.42672
4	1	1.14214	1.06794	1.00439
5	0	0.940132	0.949074	0.932976

Figure 10 Stability Bounds for BPIRK$^+$ Family with Lagrange Predictor and Order 5 (N=sM, m=3-M)

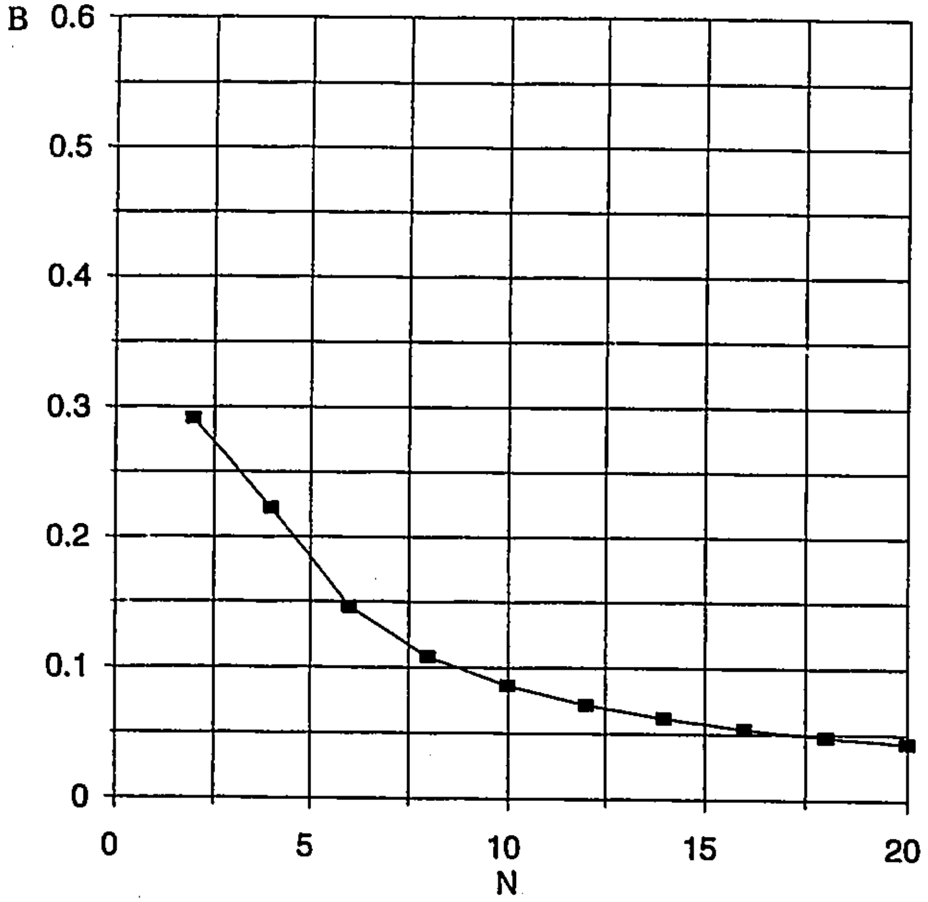


Figure 11 Stability Bounds for BPC Methods of Order 5

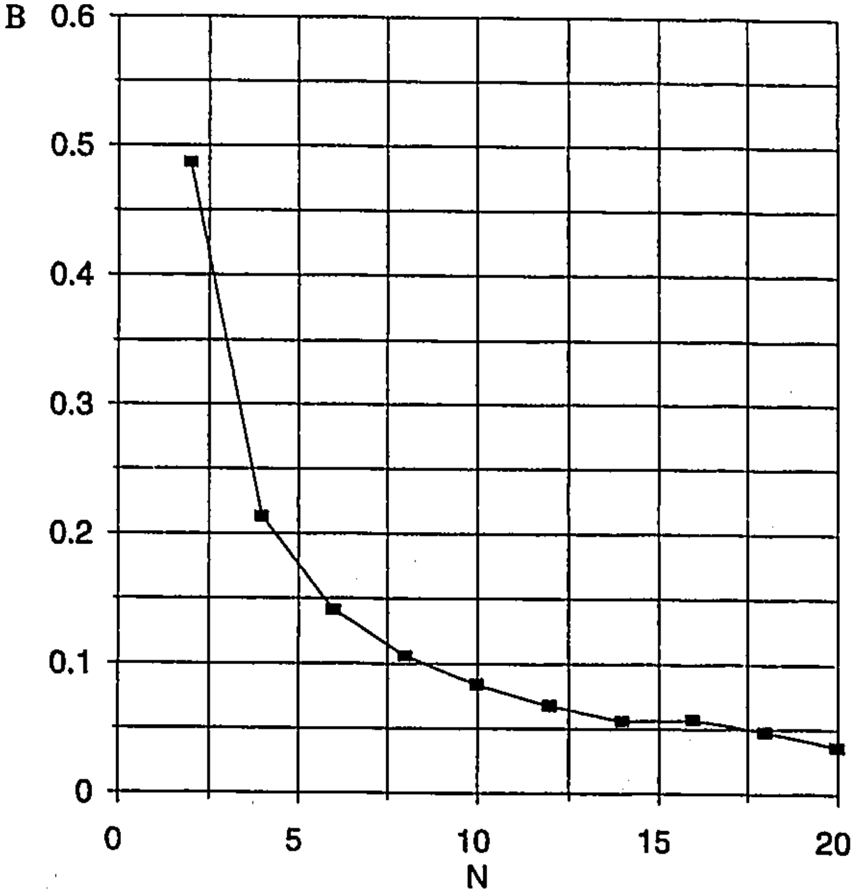
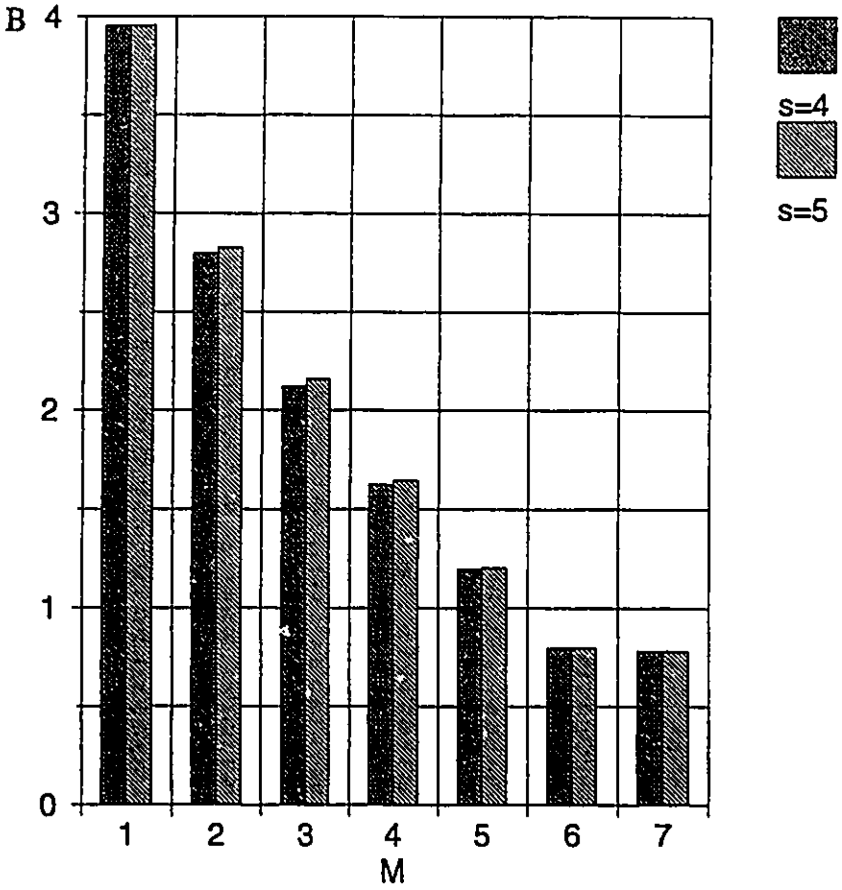
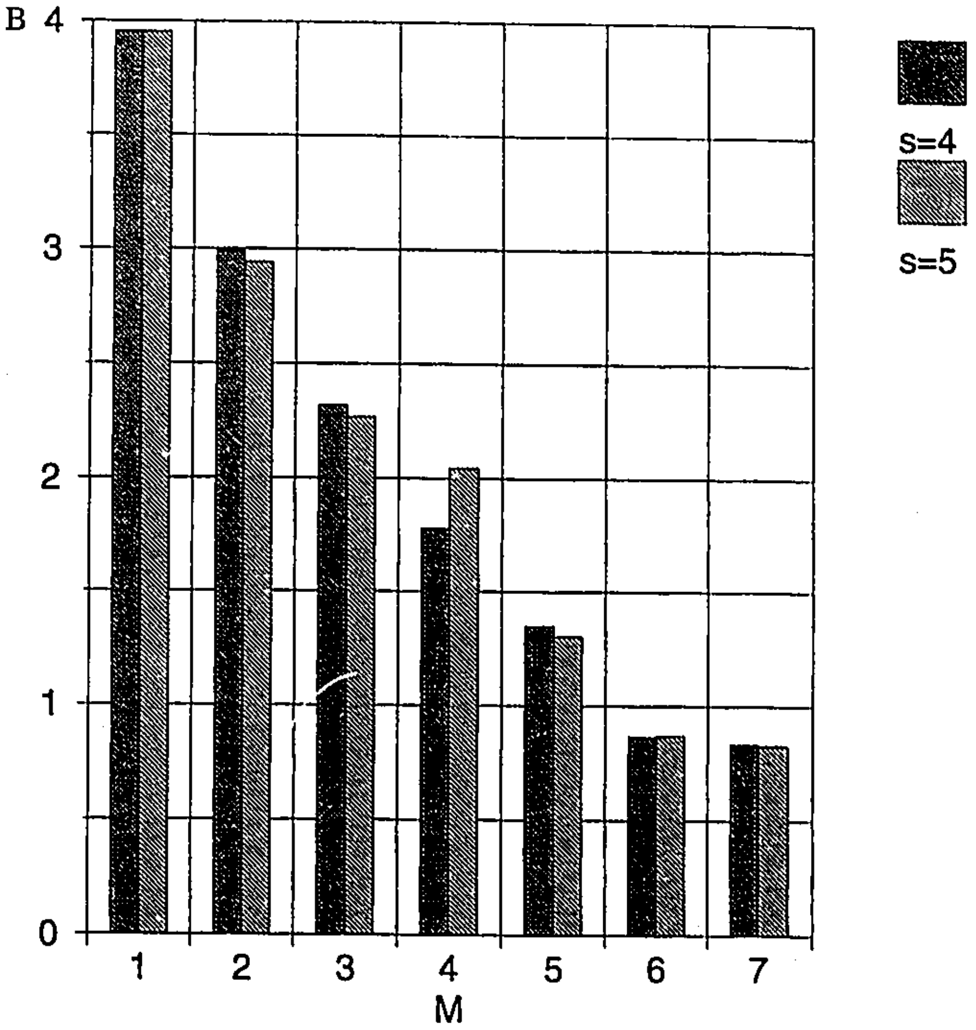


Figure 12 Stability Bounds for PBPC Methods of Order 5



M	m	s=4	s=5
1	6	3.95399	3.95364
2	5	2.79481	2.82003
3	4	2.11775	2.15432
4	3	1.62148	1.64395
5	2	1.19585	1.20518
6	1	0.793037	0.795372
7	0	0.779716	0.779778

Figure 13 Stability Bounds for BPIRK Family with Lagrange Predictor and Order 7 (N=sM, m=3-M)



M	m	s=4	s=5
1	6	3.95399	3.95364
2	5	2.98966	2.94013
3	4	2.31641	2.26653
4	3	1.77465	2.04244
5	2	1.34814	1.30213
6	1	0.865486	0.870338
7	0	0.833153	0.828056

Figure 14 Stability Bounds for BPIRK$^+$ Family with Lagrange Predictor and Order 7 (N=sM, m=3-M)

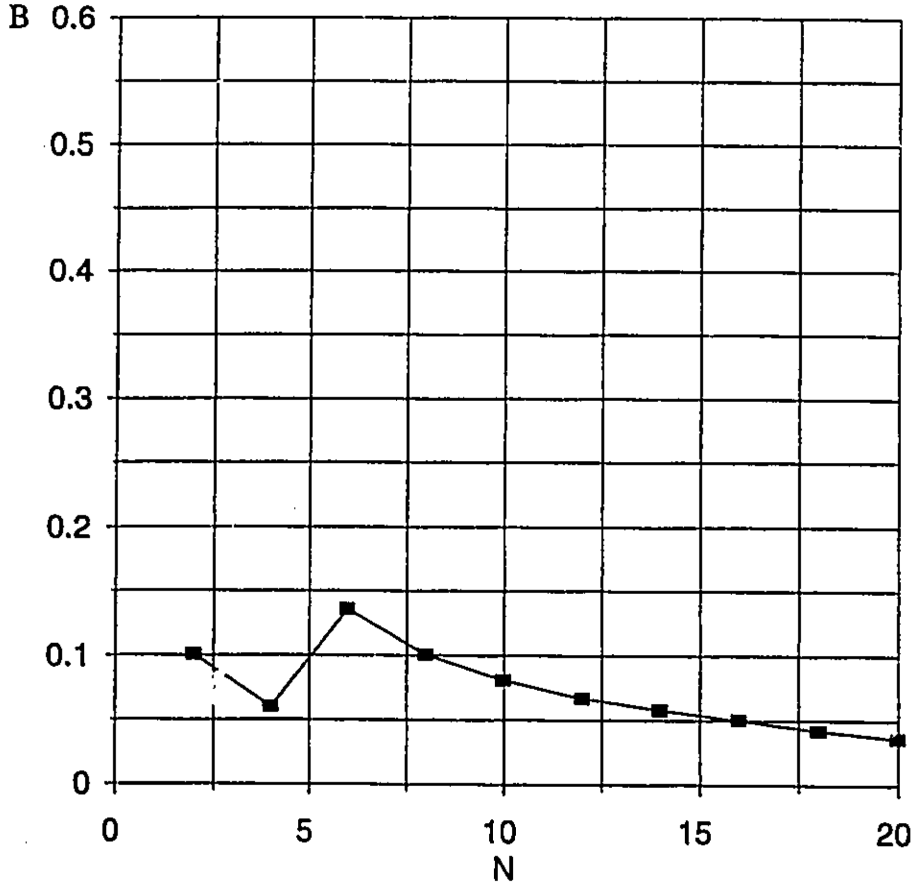


Figure 15 Stability Bounds for BPC Methods of Order 7

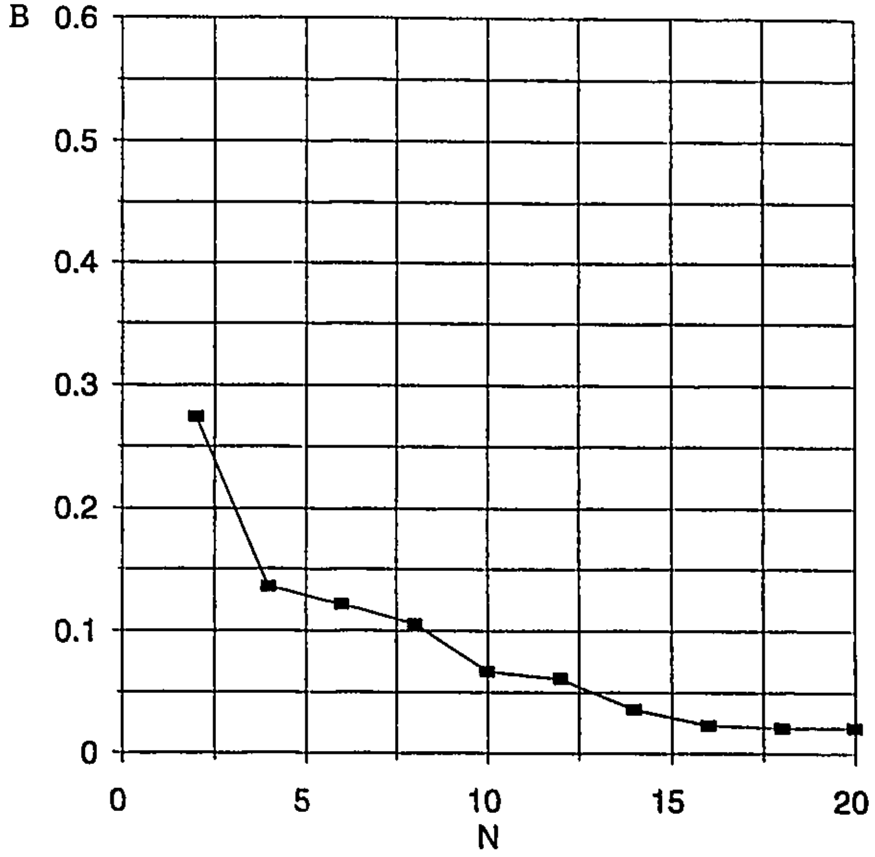


Figure 16 Stability Bounds for PBPC Methods of Order 7

Chapter 4 Performance Evaluation

4.1 Scope of the Experiments

An extensive collection of numerical experiments was carried out to evaluate the performance of the BPIRK and BPIRK$^+$ families. The goals here were both to obtain some insight into the relative performance of these two families and to obtain insight into their performance relative to the BPC and PBPC families. The experiments examine the relative behavior of the Lagrange and Hermite predictors (with various values of the parameter m) as well as the effect of various block sizes, M. A variety of solution orders, r, and solution accuracies are also investigated.

The experiments were carried out on a collection of five test problems which are listed in Appendix I. These problems share the feature of having an explicit closed form solution which provided the basis for determining global solution error in the experiments. These test problems are taken from Abou-Rabia et al. [1]. A sixth problem (TP6) is also given in Appendix I. This problem was used as a basis for validating the implementation (see section 4.4.1).

The experiments were carried out on a single processor workstation (486 PC) using 14 digit (approximate) arithmetic precision. The program code was written in C.

4.2 Comparison Criteria

Two main issues were of concern in the evaluation activity; namely, solution accuracy and speed. Solution accuracy was formulated as follows:

$$(\text{divisor})_i = \max(1, |y_i(t_n)|)$$

and

$$E(n) = \max_{i} |(y_n)_i - y_i(t_n)| / (\text{divisor})_i$$

where $(y_n)_i$ is the i^{th} component of the numerical solution vector at point $t=t_n$ and $y_i(t_n)$ is the corresponding true (exact) solution at t_n. The accuracy of a solution is given by

$$Ac = \max_{n} E(n)$$

We say a solution accuracy of G has been achieved if

$$| Ac - G | < \varepsilon$$

where ε is a small value (e.g., 10^{-9}).

The measure of solution speed used in the experiments was taken to be the number of derivative function evaluations (dfe's) per processor over the prescribed solution interval. This approach offers several advantages which include implementation simplicity and repeatability of results over different computer hardware. We note also that the approach does tend to become increasingly more realistic when the complexity of the derivative functions increases and/or the size of the model (as measured by the number of ode's) increases.

The experiments were organized to examine performance at three different solution accuracies corresponding roughly to low, moderate and high. The corresponding values used for G were : 10^{-3}, 10^{-5} and 10^{-7}.

4.3 Flowchart of the Experimentation Program

In this section we summarize experimentation procedure used in generating the test results. This summary is given in the form of a flowchart. Some parts of this flowchart refer to the underlying BPIRK procedure summarized in Figures 1 and 2 (see section 2.4). A highlevel view is given below.

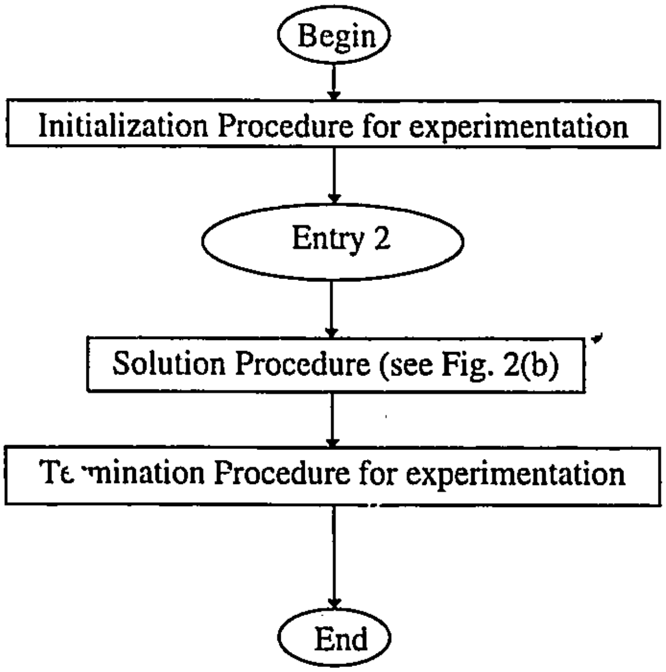


Figure 17 Highlevel View of Experimentation

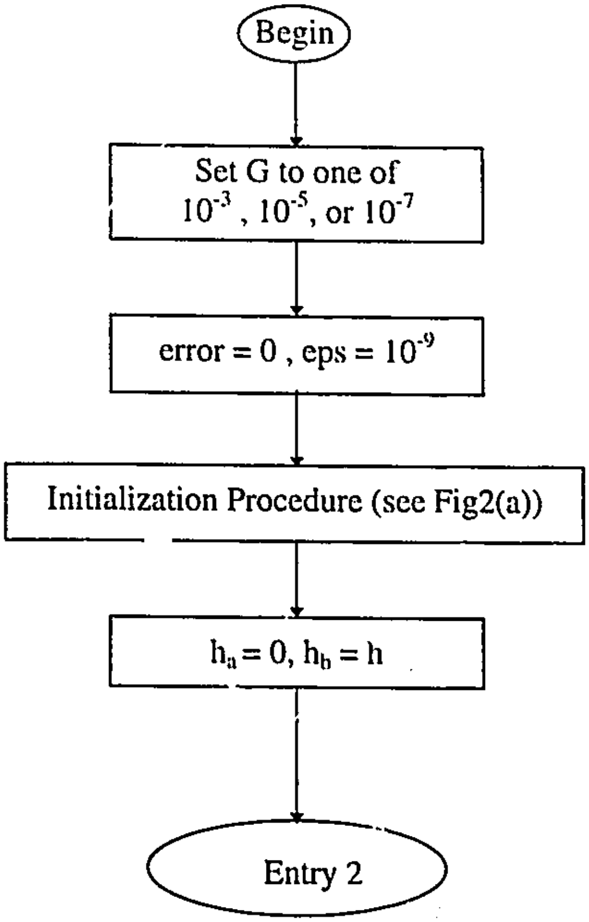


Figure 18(a) Initialization Procedure for Experimentation

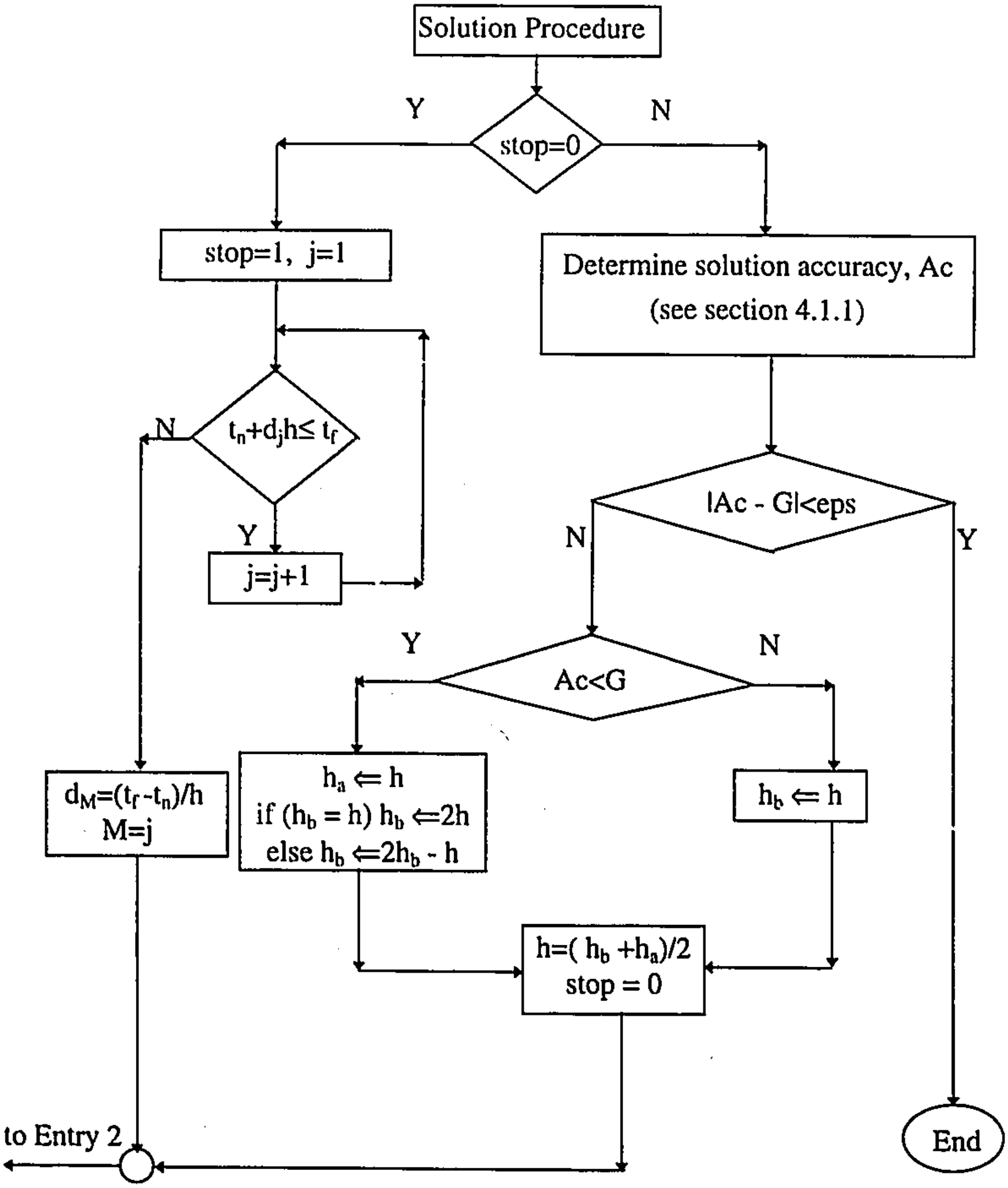


Figure 18(b) Termination Procedure for Experimentation

4.4 Test Results

4.4.1 Validation of the Implementation

We discuss in this section a validation step that was undertaken to establish confidence in the solution code that was developed for the BPIRK approach. The objective here was to duplicate results presented in [20] for Test Problem TP6 (see Appendix I). These results are reproduced below in Table 3 from Table 7 in [20]. We report in Table 4 the test results for TP6 to validate the implementation of BPIRK method.

N_{dfe}	M=1,m=3	M=4,m=0	M=4,m=1	M=4,m=2
240	1.2	3.5	3.5	2.4
480	2.7	5.1	4.8	3.7
960	3.9	6.7	6.0	4.9
1920	5.1	8.2	7.2	6.1

Table 3 Values of accuracy_order Obtained in [20] for the BPIRK Method with s=2 and the Lagrange Predictor

The entries in this Table are values for an accuracy measure called "accuracy_order" which is defined in [20] as follows:

$$\text{accuracy\_order} = \log_{10} | y_{Nf} - y(t_f) |$$

where y_{Nf} is the computed solution value at $t=t_f$ and $|\bullet|$ denotes the Euclidean norm. The column heading N_{dfe} represents number of derivative function evaluations per processor over the course of the solution interval $[0, t_f]$. The underlying relationship between integration stepsize, h, number of steps over the course of the solution interval $[0, t_f]$, N_{step} and t_f is:

$$h \times N_{step} = t_f$$

Furthermore, from the specifications for the BPIRK approach (see (2.3.6) (2.3.7) together with either (2.3.8) (Lagrange predictor) or (2.3.9) (Hermite predictor)), we have that:

$$N_{step} = \begin{cases} (m+1)N_{dfc} & \text{(Lagrange predictor)} \\ (m+2)N_{dfc} & \text{(Hermite predictor)} \end{cases}$$

The specification for stepsize, for any particular experiment, is therefore:

$$h = \begin{cases} (m+1)t_f/N_{dfc} & \text{(Lagrange predictor)} \\ (m+2)t_f/N_{dfc} & \text{(Hermite predictor)} \end{cases}$$

The results obtained with our implementation which correspond to those shown in Table 3, are given below:

N_{dfc}	M=1,m=3	M=4,m=0	M=4,m=1	M=4,m=2
240	1.634	3.577	3.537	2.440
480	2.718	5.152	5.273	3.873
960	4.711	6.688	6.185	5.243
1920	5.610	8.220	7.508	6.560

Table 4 Values of accuracy_order Obtained in Our Implementation for the BPIRK Method with s=2 and the Lagrange Predictor

The differences which exist between the accuracy_order values in Tables 3 and 4 are relatively small. Nevertheless, there appears to be an anomaly here because our results (with a 14 digit machine) tend to be more accurate than those in [20] which uses a 28 digit machine. This counterintuitive result may be due to implementation differences.

4.4.2 Evaluation of Test Results

The results obtained from an extensive set of numerical experiments are summarized in Tables 5 and 6 and Tables 9 through 18. The test problems used in these experiments are specified in Appendix I. The entries in these Tables are values for N_{dfc}; i.e., the number of derivative function evaluations per processor, required to provide the solution over the solution interval $[0, t_f]$. We take this to be a reasonable indication of solution speed.

The data in Tables 5 and 6 is organized to facilitate an examination of solution speed for corresponding values of order of the solution process (i.e., r). Results are given for all five test problems in Table 5 for the BPIRK approach using the Lagrange predictor. Analogous results are provided in Table 6 for the BPIRK$^+$ case. The organization of Tables 5 and 6 enables a comparison with solution speed results for the BPC and PBPC approaches as presented in Tables 7 and 8. The results in these latter two tables has been taken from [1].

The data in Tables 9 through 18 show solution speed results for the five test problems for the BPIRK approach, using both the Lagrange and Hermite predictors. The results in these tables provide insight into the effect of varying the parameters s, m, and M.

From Tables 5 and 6 it can be observed that there is little difference in the solution speed of the BPIRK and BPIRK$^+$ methods (Lagrange predictor) within the context of the five test problems that were treated. In many of the cases both methods have the same solution speed but when differences do occur, the BPIRK methods is usually (but not always) faster. It is also of interest to observe that the case m=0 consistently shows significant deterioration (this corresponds to the case where the (Lagrange) predictor for U in equation (2.3.2) is accepted as the solution for this implicit equation). Two other general trends can be noted:

a) as order (r) increases, solution speed tends to decrease

and

b) for any particular order, there is very little sensitivity to stage value (s) of the underlying RK method, but the effect of m can be substantial. These observations apply equally to all five test problems.

The data in Tables 6 and 7 enables a comparison of the BPIRK and BPIRK$^+$ results with those of the BPC and PBPC methods. The data presentation is intended to facilitate comparison on the basis of methods having the same solution order. The influence of the various parameters (in particular, N, the number of processors used) is a complicating factor in this comparison. It is, nevertheless, clear that the solution speed for the BPIRK and BPIRK$^+$ families is significantly less than for the BPC and PBPC families (which are quite comparable). There are, for example, many cases where the BPC/PBPC results are 3 times (and even more) faster than the BPIRK/BPIRK$^+$ results.

Tables 9 through 18 provide, in particular, a basis for evaluating the relative advantages of the Lagrange and Hermite predictors. The results, however, do not support a clear conclusion. There are significant numbers of cases where each approach provides superior solution speed relative to the other. This observation is, furthermore, somewhat problem dependent; e.g., the TP3 results are heavily biased in favor of the Hermite predictor.

It can also be noted that solution speed generally decreases with increasing order of the solution method. There is also clear evidence that the M=1 case (which corresponds to the PIRK approach) is inferior.

Solution speed behavior as a function of the number of processors used is an aspect of the results that is of special interest. The desired behavior here is a monotonic increase in solution speed as the number of processors increases. Interpreting the results in Tables 9 through 18 from this perspective, needs to be done with care. In particular, it should be noted that for fixed s (i.e., a particular underlying RK method), increasing N is associated with increasing M (because

N=sM). However, from the order relation r=min(2s, M+m) (Lagrange predictor) or r=min(2s, 2M+m) (Hermite predictor) (see section 3.2), the order of the solution method can be altered as M increases.

Data relevant to this issue is presented in Table 19. In this Table, the fastest solution speed (least N_{dfe}) for several values of N is given for each of the solution accuracies considered, for each of the problems. The Table entries have been selected without regard to the values of the associated parameters m, M and s. Each N_{dfe} values was selected because it is the least value for the given problem and the selected values of N and G.

We observe in Table 19 that there the desired monotonic increase in solution speed does occur as N increase from 2 to 5. However, at N=6, there is generally a decrease in solution speed which is then followed by some erratic behavior. In many cases the solution speed attained at N=5 is not improved upon.

In Tables 5 through 19, we use the following notation:

N : Number of processors used (in BPIRK and $BPIRK^+$ cases N=sM)
M : Number of points in one block
s : Stage value RK method
m : Number of iterations in one step

					TP1			TP2			TP3			TP4			TP5		
r	s	m	M	N	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
3	2	2	1	2	411	1902	8832	450	2499	12441	2454	11382	52821	1941	8742	40264	813	3756	17409
		1	2	4	276	1298	6038	594	2690	12402	1362	6346	29472	1094	4932	22722	532	2484	11534
		0	3	6	454	4479	44729	921	8975	89474	2909	29080	290786	519	1644	7496	1098	11008	110111
	3	2	1	3	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	6	292	1370	6376	594	2646	12154	1502	6992	32464	1206	5434	25062	534	2494	11558
		0	3	9	454	4479	44729	919	8972	89471	2909	29080	290786	565	2094	10697	1098	11008	110111
	4	2	1	4	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	8	298	1400	5612	492	2626	12044	1556	7240	33618	1248	5638	25958	534	2494	11558
		0	3	12	454	4479	44729	917	8971	89469	2909	29080	290786	600	2335	12401	1098	11008	110111
	5	2	1	5	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	10	302	1414	6582	592	2616	11984	1586	7362	34188	1248	5738	25958	534	2492	11576
		0	3	15	454	4479	44729	916	8969	89468	2909	29080	290786	602	2337	12401	1098	11008	110111
5	3	4	1	3	155	390	980	260	555	1170	650	1640	4110	685	1635	4015	355	880	2205
		3	2	6	128	332	860	240	580	1424	464	1196	3024	492	1204	2984	304	760	1908
		2	3	9	138	471	1512	207	663	2139	519	1638	5166	225	567	1392	366	1170	3708
		1	4	12	212	1012	4734	608	2820	13084	472	1610	7470	158	372	810	528	2476	11498
		0	5	15	454	4479	44728	924	8975	89474	2910	28080	290800	168	695	6735	1098	11008	110111
	4	4	1	4	155	390	980	260	550	1155	650	1640	4110	685	1635	4015	355	880	2205
		3	2	8	128	336	864	240	580	1424	484	1240	3132	512	1272	3092	304	760	1908
		2	3	12	138	471	1512	204	663	2139	519	1632	5166	327	792	1947	366	1170	3708
		1	4	16	212	1012	4734	608	2820	13084	472	1610	7470	158	372	810	528	2476	11498

Table 5 Test Results for TP1-TP5 Using BPIRK with the Lagrange Predictor

					TP1			TP2			TP3			TP4			TP5		
r	s	m	M	N	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
5	4	0	5	20	454	4479	44729	924	8975	89474	2910	28080	290800	167	694	6736	1098	11008	110111
	5	4	1	5	155	390	980	260	550	1155	650	1640	4110	685	1635	4015	355	880	2205
		3	2	10	128	340	868	240	580	1424	496	1264	3188	536	1288	3144	304	760	1908
		2	3	15	138	471	1512	201	663	2139	519	1632	5166	363	873	2142	366	1170	3708
		1	4	20	212	1012	4734	608	2820	13084	472	1610	7470	206	368	1112	528	2476	11498
		0	5	25	454	4479	44729	924	8975	89474	2454	28082	290800	164	693	6736	1098	11008	110111
7	4	6	1	4	105	210	399	161	287	518	378	728	1400	455	847	1575	259	490	938
		5	2	8	78	180	366	150	288	558	252	570	1128	366	678	1302	234	462	888
		4	3	12	65	205	475	150	275	605	320	640	1350	250	460	865	255	550	1185
		3	4	16	100	292	760	224	564	1420	356	744	1552	184	292	456	304	760	1908
		2	5	20	138	471	1512	207	663	2139	525	1638	5169	126	213	336	366	1170	3708
		1	6	24	212	1012	4734	608	2820	13084	472	1610	7470	106	176	752	528	2476	11498
		0	7	28	454	4479	44729	924	8975	89480	2910	28080	290800	111	696	6723	1098	11008	110111
	5	6	1	5	105	203	399	168	287	518	378	728	1400	455	847	1575	259	490	938
		5	2	10	78	180	366	156	294	558	246	582	1152	372	690	1320	234	462	888
		4	3	15	65	205	475	150	275	605	320	640	1345	285	525	985	255	550	1185
		3	4	20	100	292	760	224	564	1420	356	744	1552	188	348	636	304	760	1908
		2	5	25	138	471	1512	207	663	2139	525	1638	5169	123	231	393	366	1170	3708
		1	6	30	212	1012	4734	608	2820	13084	472	1610	7470	118	182	752	528	2476	11498
		0	7	35	454	4479	44732	924	8975	89458	2909	29080	290730	113	696	6785	1098	11008	110111

Table 5(Cont.) Test Results for TP1-TP5 Using BPIRK with the Lagrange Predictor

r	s	m	M	N	TP1			TP2			TP3			TP4			TP5		
					1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
3	2	2	1	2	411	1902	8832	450	2499	12441	2454	11382	52821	1941	8742	40264	813	3756	17409
		1	2	4	240	1588	7384	590	2478	11162	2060	9572	44434	1504	6780	31244	606	2816	13066
		0	3	6	454	4480	44729	906	8959	89458	2817	28992	290697	767	3764	18373	1098	11009	110111
	3	2	1	3	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	6	334	1562	7210	590	2502	11308	1832	8504	39476	1468	6626	30526	606	2816	13066
		0	3	9	454	4480	44729	908	8960	89459	2852	29024	290729	745	3635	17832	1098	11009	110111
	4	2	1	4	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	8	330	1544	7202	590	2518	12012	1982	9004	35202	1446	6512	30040	596	2774	12870
		0	3	12	454	4480	44729	908	8961	89453	2817	29001	290004	734	3542	17440	1098	11009	110111
	5	2	1	5	411	1902	8832	459	2493	12435	2454	11382	52821	1941	8742	40254	813	3756	17409
		1	2	10	328	1530	7108	590	2518	12012	2062	9572	36182	1424	6444	29676	588	2742	12726
		0	3	15	454	4480	44729	909	8962	89448	2817	28992	290006	726	3471	17146	1098	11009	110111
5	3	4	1	3	155	390	980	260	555	1170	650	1640	4110	685	1635	4015	355	880	2205
		3	2	6	140	368	932	256	600	1432	568	1436	3608	596	1460	3568	312	780	1952
		2	3	9	138	471	1515	192	663	2139	501	1614	5148	492	1185	2910	369	1170	3711
		1	4	12	212	1014	4734	610	2822	13084	478	1610	7470	330	836	2060	530	2478	11498
		0	5	15	455	4480	44729	924	8976	89474	2910	29081	290802	222	692	6736	1099	11009	110111
	4	4	1	4	155	390	980	260	555	1170	650	1640	4110	685	1635	4015	355	880	2205
		3	2	8	140	364	928	256	596	1424	520	1532	3204	600	1440	3516	308	772	1932
		2	3	12	138	471	1515	192	663	2139	525	1610	4935	483	1155	2847	369	1170	3711
		1	4	16	212	1014	4734	608	2822	13084	462	1590	7448	314	814	2008	530	2478	11498

Table 6 Test Results for TP1-TP5 Using BPIRK$^+$ with the Lagrange Predictor

					TP1			TP2			TP3			TP4			TP5		
r	s	m	M	N	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
5	4	0	5	20	455	4480	44729	924	8976	89474	2909	29080	290840	220	692	6736	1099	11009	110111
	5	4	1	5	155	390	980	260	550	1162	650	1640	4110	685	1635	4015	355	880	2205
		3	2	10	136	364	920	252	592	1426	624	1572	3192	596	1420	3476	308	764	1912
		2	3	15	138	471	1515	192	663	2139	477	1596	4932	474	1143	2802	369	1170	3711
		1	4	20	212	1014	4734	608	2822	13084	482	1592	7452	310	798	1972	530	2478	11498
		0	5	25	455	4480	44729	924	8976	89474	2909	29080	290886	218	692	6739	1099	11009	110111
7	4	6	1	4	105	210	399	161	287	518	378	728	1400	455	847	1575	259	490	938
		5	2	8	84	192	390	162	300	570	372	726	1404	408	774	1470	240	462	888
		4	3	12	65	210	475	140	275	605	305	600	1310	370	695	1315	255	550	1185
		3	4	16	100	292	760	224	568	1420	360	752	1556	312	584	1100	304	760	1908
		2	5	20	138	471	1515	207	666	2139	525	1638	5169	231	459	870	369	1173	3711
		1	6	24	212	1014	4734	610	2822	13084	472	1588	7470	152	250	752	530	2478	11498
		0	7	28	455	4498	44729	925	9024	89470	2909	29080	290800	129	834	32965	1099	11009	110111
	5	6	1	5	105	203	399	168	287	518	378	728	1400	455	847	1575	259	490	938
		5	2	10	84	192	384	162	300	570	372	726	1404	408	774	1452	234	462	888
		4	3	15	90	210	475	140	275	605	295	590	1290	365	685	1295	255	550	1185
		3	4	20	100	292	760	224	568	1420	360	752	1556	308	576	1092	304	760	1908
		2	5	25	138	471	1515	207	666	2139	525	1638	5169	225	450	855	369	1173	3711
		1	6	30	212	1014	4734	610	2822	13084	472	1588	7470	150	248	752	530	2476	11498
		0	7	35	455	4494	44729	925	9015	89455	2909	29081	290800	133	791	22753	1099	11009	110111

Table 6(Cont.) Test Results for TP1-TP5 Using BPIRK$^+$ with the Lagrange Predictor

		TP1			TP2			TP3			TP4			TP5		
r	N	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
3	2	193	501	1667	639	1829	6401	571	1599	4571	961	3071	10241	401	1279	3657
	4	251	1001	5001	535	1281	8531	1601	6401	32000	1281	5121	24001	401	1601	8535
	6	279	1335	5335	475	1707	11379	1779	8535	42669	1281	6829	33001	427	2135	9525
	8	315	1251	6669	459	2135	10241	2001	9145	41669	1537	7681	33001	535	2561	10001
	10	287	1335	6401	467	2561	11705	2135	10241	40001	1537	6859	38401	513	2561	10241
	12	335	1335	6669	475	2135	11379	2135	10669	42669	1709	8001	32001	535	2439	11113
5	2	137	249	553	245	531	1161	221	419	999	381	851	1705	175	399	913
	4	113	193	417	201	401	915	181	363	727	275	641	1397	133	319	711
	6	105	177	371	179	357	777	157	243	889	257	569	1281	119	267	611
	8	105	167	417	161	357	753	139	401	1143	241	549	1397	125	267	641
	10	101	167	445	161	367	733	161	459	1281	257	615	1537	129	285	733
	12	99	187	477	153	357	713	191	535	1335	257	641	1577	135	307	713
7	2	113	205	309	141	262	489	227	255	395	209	423	763	265	275	423
	4	101	177	249	147	199	355	273	287	309	273	307	511	355	365	375
	6	71	129	185	97	177	305	111	167	267	143	233	465	121	133	267
	8	69	125	179	89	161	291	105	155	251	137	213	427	121	133	201
	10	69	119	175	81	143	271	107	147	229	129	221	385	121	135	197
	12	71	121	177	77	143	267	103	149	335	129	215	395	123	135	215

Table 7 Test Results for the BPC Method

r	N	TP1			TP2			TP3			TP4			TP5		
		1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
3	2	127	574	3202	336	2004	6402	1252	5002	26669	962	3431	16002	336	1336	6402
	4	102	288	802	288	802	2669	360	912	2502	482	1502	4802	202	669	2002
	6	85	224	1335	268	668	2668	557	2668	13335	535	2001	8001	224	668	3049
	8	73	335	2001	251	668	4001	835	3335	16001	601	2401	9601	221	835	3848
	10	74	401	2135	230	801	4268	801	4001	16668	601	2401	12801	336	1001	4925
	12	85	446	2135	224	890	5335	835	4446	20835	668	2668	12801	241	1113	5130
5	2	87	170	448	170	367	803	155	559	1432	303	670	1718	157	337	803
	4	65	114	252	127	252	574	159	280	457	202	431	926	120	225	502
	6	58	105	225	105	225	447	154	259	479	202	366	802	119	193	413
	8	58	102	202	86	202	402	141	252	457	169	301	752	120	182	382
	10	55	101	179	81	179	401	144	237	418	173	287	687	119	177	366
	12	57	97	192	80	168	382	140	224	349	163	287	668	119	181	334
7	2	76	129	227	122	215	368	136	190	389	204	357	671	167	200	368
	4	57	86	169	84	136	252	138	345	252	136	233	402	191	197	233
	6	54	86	151	69	105	193	110	154	225	136	202	336	140	145	203
	8	52	69	136	69	102	156	107	150	218	129	202	302	124	128	195
	10	52	68	136	69	102	155	102	145	213	122	194	302	124	128	183
	12	52	68	129	69	105	159	104	141	211	117	193	299	124	128	182

Table 8 Test Results for the PBPC Method

			G=1E-03			G=1E-05			G=1E-07		
S	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	411	125	253	1902	345	759	8832	1080	2376
	2	4	141	115	253	477	345	759	1527	1080	2376
	3	6	138	110	253	471	345	759	1518	1080	2376
	4	8	138	110	253	471	345	759	1518	1080	2376
	5	10	138	110	253	471	345	759	1518	1080	2376
	6	12	138	110	253	471	345	759	1518	1080	2376
3	1	3	411	155	143	1902	390	264	8832	980	572
	2	6	138	75	143	474	210	264	1530	480	572
	3	9	138	70	143	471	210	264	1512	475	572
	4	12	138	65	143	471	210	264	1512	475	572
	5	15	138	65	143	471	210	264	1512	475	572
	6	18	138	65	143	471	210	264	1512	475	572
4	1	4	411	155	110	1902	390	187	8832	980	286
	2	8	138	80	110	474	205	187	1533	475	286
	3	12	138	65	110	471	205	187	1512	475	286
	4	16	138	60	110	471	205	187	1512	475	286
	5	20	138	55	110	471	205	187	1512	475	286
	6	24	138	55	110	471	205	187	1512	475	286
5	1	5	411	155	77	1902	390	121	8832	980	198
	2	10	138	80	77	477	205	121	1533	475	198
	3	15	138	65	77	471	205	121	1512	475	198
	4	20	138	45	77	471	205	121	1512	475	198
	5	25	138	45	77	471	205	121	1512	475	198
	6	30	138	45	77	471	205	121	1512	475	198

Table 9 Test Results for TP1 Using BPIRK with the Lagrange Predictor

			G=1E-03			G=1E-05			G=1E-07		
s	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	450	535	1166	2499	1680	3685	12441	5300	11660
	2	4	354	530	1166	1149	1680	3685	3660	5300	11660
	3	6	312	530	1166	1038	1680	3685	3318	5300	11660
	4	8	312	530	1166	1038	1675	3685	3318	5300	11660
	5	10	312	530	1166	1038	1675	3685	3318	5300	11660
	6	12	312	530	1166	1038	1675	3685	3318	5300	11660
3	1	3	456	260	374	2493	555	803	12435	1170	1738
	2	6	339	160	374	1107	295	803	3534	585	1738
	3	9	207	180	374	663	370	803	2139	775	1738
	4	12	207	180	374	663	370	803	2139	775	1738
	5	15	207	180	374	663	370	803	2139	775	1738
	6	18	207	180	374	663	370	803	2139	775	1738
4	1	4	459	260	198	2493	550	352	12435	1155	616
	2	8	351	150	198	1143	365	352	3645	810	616
	3	12	204	150	198	663	275	352	2139	605	616
	4	16	207	155	198	663	275	352	2139	605	616
	5	20	207	155	198	663	275	352	2139	605	616
	6	24	207	155	198	663	275	352	2139	605	616
5	1	5	459	260	121	2493	550	209	12435	1155	330
	2	10	357	150	132	1161	370	209	3699	825	319
	3	15	201	150	132	663	275	209	2139	605	319
	4	20	207	150	132	663	275	209	2139	605	319
	5	25	207	150	132	663	275	209	2139	605	319
	6	30	207	150	132	663	275	209	2139	605	319

Table 10 Test Results for TP2 Using BPIRK with the Lagrange Predictor

			G=1E-03			G=1E-05			G=1E-07		
s	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	2454	400	770	11382	1060	2420	52821	3470	7645
	2	4	492	345	770	1611	1085	2420	5172	3470	7645
	3	6	519	280	770	1632	1100	2420	5166	3475	7645
	4	8	525	280	770	1638	1100	2420	5166	3475	7645
	5	10	525	280	770	1638	1100	2420	5166	3475	7645
	6	12	525	280	770	1638	1100	2420	5166	3475	7645
3	1	3	2454	650	352	11382	1640	583	52821	4110	1210
	2	6	501	305	319	1617	610	561	5172	1310	1210
	3	9	525	320	319	1638	640	561	5166	1350	1210
	4	12	525	320	319	1638	640	561	5166	1350	1210
	5	15	525	320	319	1638	640	561	5166	1350	1210
	6	18	525	320	319	1638	640	561	5166	1350	1210
4	1	4	2454	650	220	11382	1640	396	52821	4110	572
	2	8	492	305	198	1611	600	352	5172	1300	528
	3	12	519	320	198	1632	640	352	5166	1350	528
	4	16	525	320	187	1638	640	352	5166	1350	528
	5	20	525	320	187	1638	640	352	5166	1350	528
	6	24	525	320	187	1638	640	352	5166	1350	528
5	1	5	2454	650	231	11385	1640	352	52821	4110	539
	2	10	489	300	198	1611	590	319	5166	1290	451
	3	15	519	320	165	1632	640	319	5166	1345	440
	4	20	522	320	165	1638	640	319	5169	1345	440
	5	25	522	320	165	1638	640	319	5169	1350	440
	6	30	522	320	165	1638	640	319	5169	1350	440

Table 11 Test Results for TP3 Using BPIRK with the Lagrange Predictor

			G=1E-03			G=1E-05			G=1E-07		
s	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	1935	945	2046	8742	2935	6457	40254	9265	20383
	2	4	720	945	2046	2280	2935	6457	7188	9265	20383
	3	6	555	930	2046	1761	2935	6457	5550	9265	20383
	4	8	555	930	2046	1761	2935	6457	5550	9265	20383
	5	10	555	930	2046	1761	2935	6457	5550	9265	20383
	6	12	555	930	2046	1761	2935	6457	5550	9265	20383
3	1	3	1941	685	616	8742	1635	1353	40254	4015	2893
	2	6	804	410	627	2529	925	1353	7986	1990	2893
	3	9	225	275	627	567	615	1353	1392	1315	2893
	4	12	153	295	627	288	620	1353	738	1325	2893
	5	15	117	295	627	318	620	1353	738	1325	2893
	6	18	108	295	627	318	620	1353	738	1325	2893
4	1	4	1941	685	330	8742	1635	594	40254	4015	1023
	2	8	831	435	319	2607	960	550	8268	2070	1023
	3	12	321	250	264	792	460	561	1947	865	1023
	4	16	183	185	275	330	260	561	570	455	1023
	5	20	126	170	297	213	240	561	336	455	1023
	6	24	96	160	319	171	240	561	261	455	1023
5	1	5	1941	685	297	8742	1635	440	40254	4015	715
	2	10	843	455	253	2658	960	374	8412	2105	561
	3	15	363	285	242	873	525	363	2142	985	539
	4	20	192	190	242	384	290	319	891	520	517
	5	25	123	170	231	231	225	308	381	290	517
	6	30	102	160	242	177	190	319	258	240	517

Table 12 Test Results for TP4 Using BPIRK with the Lagrange Predictor

S	M	N	G=1E-03			G=1E-05			G=1E-07		
			m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	813	395	825	3756	1205	2640	17409	3790	8338
	2	4	366	395	825	1170	1205	2640	3708	3790	8338
	3	6	366	375	825	1170	1200	2640	3708	3790	8338
	4	8	366	375	825	1170	1200	2640	3708	3790	8338
	5	10	366	375	825	1179	1200	2640	3708	3790	8338
	6	12	366	375	825	1170	1200	2640	3708	3790	8338
3	1	3	813	355	286	3756	880	682	17409	2205	1485
	2	6	366	260	319	1170	550	682	3708	1190	1485
	3	9	366	255	308	1170	550	682	3708	1185	1485
	4	12	366	255	308	1170	555	682	3708	1190	1485
	5	15	366	255	308	1170	555	682	3708	1190	1485
	6	18	366	255	341	1170	555	682	3708	1190	1485
4	1	4	813	355	220	3756	880	341	17409	2205	583
	2	8	366	260	209	1170	550	319	3708	1190	572
	3	12	366	255	198	1170	550	330	3708	1185	583
	4	16	366	255	209	1170	550	330	3708	1185	583
	5	20	366	255	242	1170	550	330	3708	1185	583
	6	24	366	255	275	1170	550	330	3708	1185	583
5	1	5	813	355	176	3756	880	297	17409	2205	451
	2	10	366	260	176	1170	550	264	3708	1190	396
	3	15	366	255	165	1170	550	275	3708	1185	396
	4	20	366	255	165	1170	550	275	3708	1185	396
	5	25	366	255	187	1170	550	275	3708	1185	396
	6	30	366	255	209	1170	550	275	3708	1185	396

Table 13 Test Results for TP5 Using BPIRK with the Lagrange Predictor

S	M	N	G=1E-03			G=1E-05			G=1E-07		
			m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	321	175	264	2212	330	770	4832	1110	2387
	2	4	141	120	242	486	345	770	1566	1080	2376
	3	6	144	120	242	474	350	770	1518	1085	2387
	4	8	144	120	242	474	350	770	1521	1085	2387
	5	10	144	120	242	474	350	770	1521	1085	2387
	6	12	144	115	231	474	350	770	1521	1085	2387
3	1	3	321	185	220	1923	330	286	4832	995	539
	2	6	138	95	132	444	315	286	1239	510	539
	3	9	141	95	132	441	315	286	1230	510	539
	4	12	141	80	110	441	315	286	1230	515	539
	5	15	141	80	110	441	310	275	1230	515	539
	6	18	141	80	110	441	310	275	1230	515	539
4	1	4	321	185	200	1803	330	209	4832	995	220
	2	8	138	100	110	417	295	209	1233	485	220
	3	12	141	90	110	414	295	209	1230	485	220
	4	16	141	70	88	414	295	198	1230	485	220
	5	20	141	85	88	414	295	198	1230	485	220
	6	24	141	85	88	414	295	198	1230	485	220
5	1	5	321	185	132	1803	330	198	4832	995	176
	2	10	141	95	88	384	280	198	1233	485	176
	3	15	138	75	88	381	280	132	1230	485	176
	4	20	138	65	77	381	280	132	1230	485	176
	5	25	141	75	88	381	280	132	1230	485	176
	6	30	141	75	88	381	280	132	1230	485	176

Table 14 Test Results for TP1 Using BPIRK with the Hermite Predictor

s	M	N	G=1E-03			G=1E-05			G1E-07		
			m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	536	535	840	1696	1680	2640	5360	5300	8340
	2	4	372	530	840	1388	1680	2640	4532	5300	8340
	3	6	420	530	840	1384	1675	2640	4424	5300	8340
	4	8	416	530	840	1384	1675	2640	4424	5300	8340
	5	10	420	530	840	1384	1675	2640	4424	5300	8340
	6	12	2224	530	840	2228	1675	2640	4424	5300	8340
3	1	3	496	260	336	1568	555	612	4952	1170	1320
	2	6	416	160	360	1236	295	612	4024	585	1320
	3	9	268	180	348	812	370	612	2636	775	1320
	4	12	276	180	348	884	370	612	2852	775	1320
	5	15	268	180	348	884	370	612	2852	775	1320
	6	18	316	180	348	884	370	612	2852	775	1320
4	1	4	496	260	216	1568	550	384	4952	1155	672
	2	8	472	150	216	1376	365	384	4495	810	672
	3	12	364	150	216	1232	275	384	3676	605	672
	4	16	364	155	216	900	275	384	2924	605	672
	5	20	264	155	216	884	275	384	2852	605	672
	6	24	308	155	216	884	275	384	2852	605	672
5	1	5	496	260	216	1568	550	228	4952	1155	432
	2	10	524	150	144	1496	370	228	4916	825	348
	3	15	532	150	144	1776	275	228	5376	605	348
	4	20	452	150	144	1152	275	228	3880	605	348
	5	25	268	150	144	864	275	228	2784	605	348
	6	30	296	150	144	884	275	228	2852	605	348

Table 15 Test Results for TP2 Using BPIRK with the Hermite Predictor

			G=1E-03			G=1E-05			G=1E-07		
s	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	336	396	770	1036	1302	2420	52821	4164	7645
	2	4	776	462	770	2504	1368	2420	7928	4188	7645
	3	6	704	348	770	2176	1320	2420	6856	4170	7645
	4	8	704	414	770	2176	1320	2420	6856	4170	7645
	5	10	704	348	770	2176	1320	2420	6856	4170	7645
	6	12	1008	516	770	2332	1320	2420	6860	4170	7645
3	1	3	556	366	352	1360	708	583	52821	1368	1210
	2	6	964	420	319	3080	888	561	9760	1920	1210
	3	9	644	378	319	2024	714	561	6404	1458	1210
	4	12	700	384	319	2184	768	561	6892	1620	1210
	5	15	488	384	319	2184	768	561	6892	1620	1210
	6	18	632	372	319	2184	768	561	6892	1626	1210
4	1	4	556	366	220	1360	702	396	3808	1356	572
	2	8	1076	450	198	3448	954	352	10924	2052	528
	3	12	880	306	198	2644	570	352	8280	1936	528
	4	16	721	384	187	2244	768	352	7096	1614	528
	5	20	688	402	187	2184	768	352	6890	1614	528
	6	24	704	384	187	2184	774	352	6890	1620	528
5	1	5	556	366	231	1360	702	352	3808	1356	539
	2	10	1176	474	198	3784	1008	319	11980	2184	451
	3	15	1228	270	165	3772	966	319	11860	1963	440
	4	20	948	396	165	3048	798	319	9700	1692	440
	5	25	688	384	165	2136	768	319	6760	1620	440
	6	30	632	426	165	2184	768	319	6892	1620	440

Table 16 Test Results for TP3 Using BPIRK with the Hermite Predictor

			G=1E-03			G=1E-05			G=1E-07		
s	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	1160	1128	2232	3461	3339	6513	40944	30044	22236
	2	4	748	1122	2232	2348	3528	7044	7404	11124	22236
	3	6	740	1116	2232	2348	3528	7044	7400	11124	22236
	4	8	740	1116	2232	2348	3528	7044	7400	11124	22236
	5	10	740	1116	2232	2348	3528	7044	7400	11124	22236
	6	12	740	2886	2232	2348	3528	7044	7392	11124	22236
3	1	3	1160	546	684	3519	1194	1476	40944	2598	3156
	2	6	520	366	684	1284	666	1476	3132	1530	3156
	3	9	264	360	684	544	750	1476	1264	1590	3156
	4	12	244	348	684	492	744	1476	996	1590	3156
	5	15	244	354	684	444	744	1476	964	1590	3156
	6	18	440	396	684	464	744	1476	984	1590	3156
4	1	4	1160	546	348	3527	1194	600	40944	2604	1116
	2	8	540	372	324	1296	690	612	3192	1326	1116
	3	12	260	258	312	700	438	612	1516	798	1116
	4	16	184	228	336	352	306	612	636	546	1116
	5	20	224	270	432	412	330	612	676	546	1116
	6	24	424	282	516	460	288	612	612	546	1116
5	1	5	1160	546	300	3531	1194	420	40944	2604	624
	2	10	548	390	276	1308	720	396	3212	1338	600
	3	15	364	258	252	748	474	360	1592	858	564
	4	20	228	246	264	364	330	312	780	510	564
	5	25	192	234	312	248	276	336	512	366	564
	6	30	320	282	372	336	318	384	620	396	465

Table 17 Test Results for TP4 Using BPIRK with the Hermite Predictor

			G=1E-03			G=1E-05			G=1E-07		
S	M	N	m=2	m=4	m=10	m=2	m=4	m=10	m=2	m=4	m=10
2	1	2	801	390	790	3636	1005	2420	17100	4090	10505
	2	4	336	380	770	1050	1005	1980	2803	3009	7216
	3	6	336	380	770	1050	1005	1980	2803	3009	7216
	4	8	336	380	770	1050	1005	1980	2803	3009	7216
	5	10	336	380	770	1050	1005	1980	2803	3009	7216
	6	12	336	380	770	1050	1005	1980	2803	3009	7216
3	1	3	801	345	390	3636	730	660	17100	1995	1595
	2	6	336	300	290	930	520	550	3803	995	1265
	3	9	336	295	290	930	520	550	3803	995	1265
	4	12	336	295	290	930	520	550	3803	995	1265
	5	15	336	295	290	930	520	550	3803	995	1265
	6	18	336	295	290	930	520	550	3803	995	1265
4	1	4	801	345	231	3600	730	330	17010	1995	605
	2	8	336	300	198	930	520	297	3803	995	550
	3	12	336	295	209	930	520	319	3803	995	572
	4	16	336	295	209	930	520	319	3803	995	605
	5	20	336	295	209	930	520	319	3803	995	605
	6	24	336	295	209	930	520	319	3803	995	605
5	1	5	801	345	198	3600	730	286	17010	1995	462
	2	10	336	300	154	930	520	242	3803	990	374
	3	15	336	295	176	930	520	253	3803	995	385
	4	20	336	295	176	930	520	253	3803	995	385
	5	25	336	295	176	930	520	253	3803	995	385
	6	30	336	295	176	930	520	253	3803	995	385

Table 18 Test Results for TP5 Using BPIRK with the Hermite Predictor

N	TP1			TP2			TP3			TP4			TP5		
	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07	1E-03	1E-05	1E-07
2	125	345	1080	450	1680	5300	400	1060	3470	945	2935	9265	395	1205	3790
3	143	264	572	260	555	1170	352	583	1210	616	1353	2893	286	682	1485
4	105	187	286	161	287	518	220	396	572	330	594	1023	220	341	583
5	77	121	198	121	209	330	231	352	539	297	440	715	176	297	451
6	75	210	480	160	295	585	305	561	1210	410	925	1990	260	550	1190
8	78	180	366	150	288	558	198	352	528	319	550	1023	209	319	572
9	70	210	475	180	370	775	319	561	1210	225	567	1315	255	550	1185
10	77	121	198	132	209	319	198	319	451	253	374	561	176	264	396
12	65	187	286	150	275	605	198	352	528	250	460	865	198	330	583
15	65	121	198	132	209	319	165	319	440	242	363	539	165	275	396
16	60	187	286	155	275	605	187	352	528	183	260	445	209	330	583
18	65	210	475	180	370	775	319	561	1210	108	318	738	255	555	1190
20	45	121	198	132	275	605	187	352	528	126	213	336	165	275	396
24	55	187	286	155	275	605	187	352	528	96	171	261	255	330	583
25	45	121	198	132	209	319	165	319	440	123	225	290	187	275	396

Table 19 Dependence of Solution Speed on Number of Processors

Chapter 5 Concluding Remarks

5.1 Conclusions

Our various observations from this study have been discussed in earlier sections (e.g., sections 3.2 and 4.4.2). A summary of the most significant points is provided below:

i) the BPIRK^+ family of methods, as proposed in this thesis, has somewhat larger stability bounds than the BPIRK family (see Figures 3, 4, 7, 8, 11, and 12);

ii) the stability bound for the BPIRK and BPIRK^+ families is significantly larger than for the BPC and PBPC families (see Figures 3 through 14);

iii) the solution speed of the BPIRK and BPIRK^+ families is inferior to that which is provided by the BPC and PBPC families (see Tables 5, 6, 7, and 8);

iv) there is a significant penalty in solution speed if a corrector is not applied (i.e., m=0) in solving the implicit equation (2.3.2) which is associated with both the BPIRK and BPIRK^+ methods;

v) the PIRK approach, which corresponds to M=1, generally provides inferior solution performance, but its stability bound is larger than that of the BPIRK family.

5.2 Future Work

The work completed in this thesis study makes a contribution to the development of effective methods for the parallel solution of ODE's. Much work remains to be done in this area. Some topics that are worthy of investigation are given below:

i) The testing activity in this research work was carried out in a single processor environment and relied on an idealized measure of computation speed; namely, a count of derivative function evaluations per processor. Testing using actual multiprocessor hardware should be undertaken to more realistically evaluate performance of the various solution methods.

ii) Investigation of the strategies for implementing variable stepsize procedures for BPIRK and BPIRK$^+$ families of methods should be undertaken to achieve automatic stepsize control.

iii) The use of RK methods other than those considered in this thesis may enlarge the stability bounds and/or improve efficiency. This needs to be investigated.

iv) Stiff systems frequently occur in continuous system simulation studies and parallel methods specifically designed for such systems need to be developed.

v) The test problems used in this study are representative of a collection of problems widely used in testing ODE codes. These problems are not, in any sense, "large" and hence do not exploit the computing potential of multiprocessor machines. Test problems designed to more fully exploit multiprocessor computing power, need to be formulated.

Appendix I

A summary of the five test problems used in the numerical experiments is given in this Appendix.

Test Problem 1(TP1):

$$y' = y\cos(t); \qquad y_1(0) = 1$$

with

$$t_f = 20$$

Analytic solution:

$$y = e^{\sin t}$$

Test Problem 2(TP2):

$$y'_1 = -y_2 - y_1 y_3/r\,; \qquad y_1(0) = 3$$
$$y'_2 = y_1 - y_2 y_3/r\,; \qquad y_2(0) = 0$$
$$y'_3 = y_1/r\,; \qquad y_3(0) = 0$$

with

$$r = (\,y_1^2 + y_2^2)^{1/2}; \qquad t_f = 20$$

Analytic solution:

$$y_1 = (2+\cos(t))\cos(t)$$
$$y_2 = (2+\cos(t))\sin(t)$$
$$y_3 = \sin(t)$$

Test Problem 3(TP3):

$y'_1 = y_2;$ $y_1(0) = 1$
$y'_2 = -y_1/r^3;$ $y_2(0) = 0$
$y'_3 = y_4;$ $y_3(0) = 0$
$y'_4 = -y_3/r^3;$ $y_4(0) = 1$

with

$r = (y_1^2 + y_3^2)^{1/2};$ $t_f = 25$

Analytic solution:

$y_1 = \cos(t)$
$y_2 = -\sin(t)$
$y_3 = \sin(t)$
$y_4 = \cos(t)$

Test Problem 4(TP4):

$y'_1 = y_1/(2(1+t)) - 2ty_2;$ $y_1(0) = 1$
$y'_2 = y_2/(2(1+t)) + 2ty_1;$ $y_2(0) = 0$

with

$t_f = 6$

Analytic solution:

$y_1 = (1+t)^{1/2}\cos(t^2)$
$y_2 = (1+t)^{1/2}\sin(t^2)$

Test Problem 5(TP5):

$y'_1 = y_2;$ $y_1(0) = 0$

$y'_2 = -2Ay_2 - (A^2+B^2)\, y_1;$ $y_2(0) = 1$

$y'_1 = y_4;$ $y_3(0) = 0$

$y'_2 = y_1 - 2Cy_4 - (C^2+D^2)\, y_3;$ $y_4(0) = 0$

with

$t_f = 20$

Analytic solution:

$y_1(t) = e^{-At}\sin(Bt)/B\ ;$

$y_2(t) = (A^2+B^2)^{1/2}\, e^{-At}\cos(Bt+\theta_1)\ /B\ ;$

$y_3(t) = [De^{-At}\sin(Bt-\psi) + Be^{-Ct}\sin(Dt-\phi)]/Q\ ;$

$y_4(t) = [D(A^2+B^2)^{1/2}\, e^{-At}\cos(Bt-\psi+\theta_1) + B(C^2+D^2)^{1/2}\, e^{-Ct}\cos(Bt-\phi+\theta_2)]/Q\ ;$

with

$Q = BD[(C-A)^2 + D^2 - B^2)^2 + 4(C-A)^2B^2)]^{1/2}$

$\psi = \arctan[2B(C-A)/((C-A)^2 + D^2 - B^2))]$

$\phi = \arctan[2D(A-C)/((C-A)^2 - D^2 + B^2))]$

$\theta_1 = \arctan(A/B)$

$\theta_2 = \arctan(C/D)$

where we define

$A = 1, B = 10, C = 2, D = 5$

Test Problem 6(TP6):

$y'_1 = 2ty_1\log(\max\{\ y_2\ , 10^{-3}\})\ ;$ $y_1(0) = 1$

$y'_2 = -2ty_2\log(\max\{\ y_1\ , 10^{-3}\})\ ;$ $y_2(0) = e$

with

$t_f = 20$

Analytic solution:

$$y_1(t) = \exp(\sin(t^2))$$
$$y_2(t) = \exp(\cos(t^2))$$

Appendix II

The numerical experiments utilize four implicit Runge-Kutta methods as base formulas. These have stage values of s=2, 3, 4 and 5 and have orders 4, 6, 8 and 10 respectively (see [9]). These base formulas are given below:

$s = 2$

$0.5-3^{1/2}/6$	0.25	$0.25-3^{1/2}/6$
$0.5+3^{1/2}/6$	$0.25+3^{1/2}/6$	0.25
	0.5	0.5

$s = 3$

$1/2-15^{1/2}/10$	$5/36$	$2/9-15^{1/2}/15$	$5/36-15^{1/2}/30$
$1/2$	$5/36+15^{1/2}/24$	$2/9$	$5/36-15^{1/2}/24$
$1/2-15^{1/2}/10$	$5/36+15^{1/2}/30$	$2/9+15^{1/2}/15$	$5/36$
	$5/18$	$4/9$	$5/18$

$s = 4$

$1/2 - w_2$	w_1	$w_1' - w_3 + w_1'$	$w_1' - w_3 - w_4'$	$w_1 - w_5$
$1/2 - w_2'$	$w_1 - w_3' + w_1$	w_1'	$w_1' - w_5'$	$w_1 - w_3' - w_4$
$1/2 + w_2'$	$w_1 + w_3' + w_4$	$w_1' - w_5'$	w_1'	$w_1 + w_3' - w_4$
$1/2 + w_2$	$w_1 + w_5$	$w_1' + w_3 + w_1'$	$w_1' + w_3 - w_1'$	w_1
	$2w_1$	$2w_1'$	$2w_1'$	$2w_1$

where

$w_1=1/8-30^{1/2}/144$, $w_2=((15+2\times 30^{1/2})/35)^{1/2}/2$, $w_3=w_2\,(1/6+30^{1/2}/24)$,
$w_1'=1/8+30^{1/2}/144$, $w_2'=((15-2\times 30^{1/2})/35)^{1/2}/2$, $w_3'=w_2'\,(1/6-30^{1/2}/24)$,
$w_4=w_2(1/21+5\times 30^{1/2}/168)$, $w_5=w_2-2\times w_3$,
$w_4'=w_2'\,(1/21-5\times 30^{1/2}/168)$, $w_5'=w_2'-2\times w_3'$.

$s = 5$

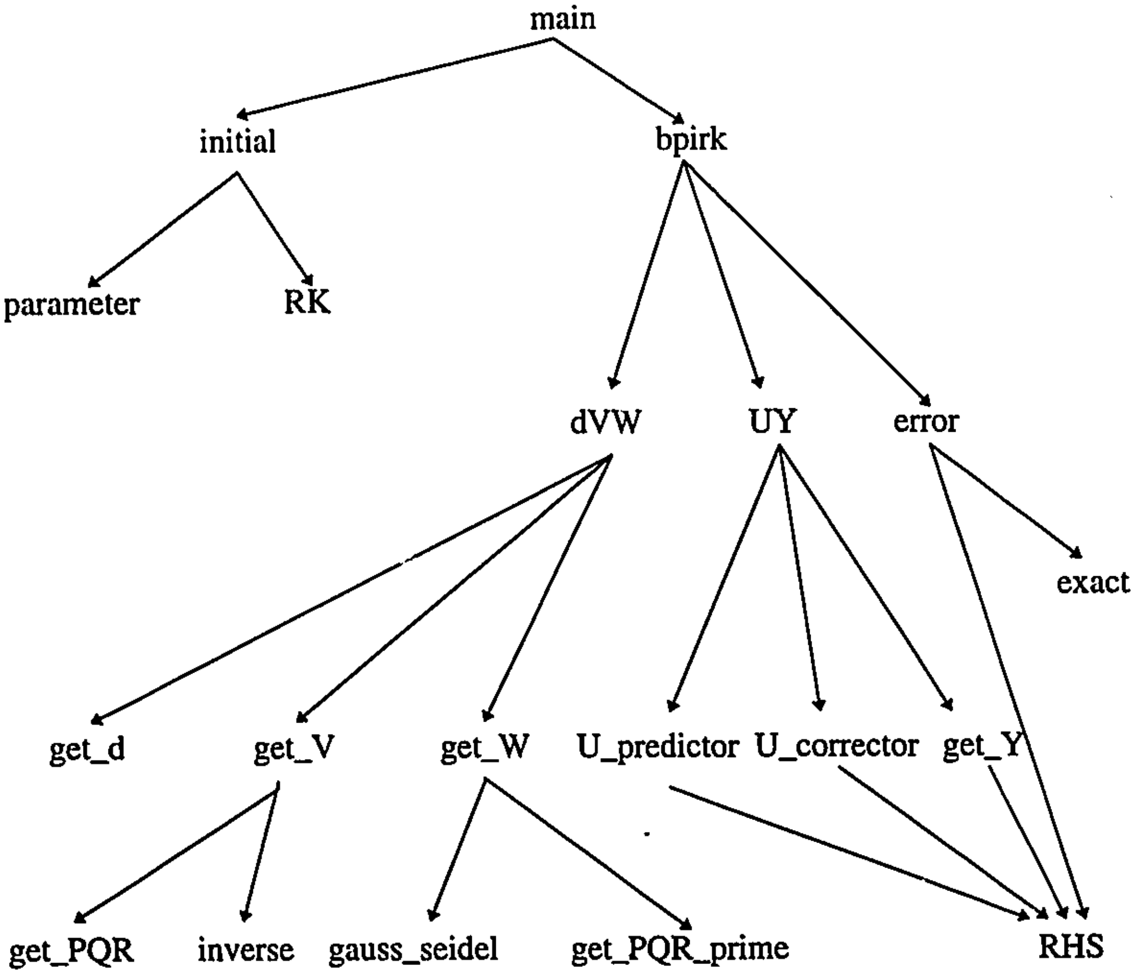
$1/2-w_2$	w_1	$w_1'-w_3+w_4'$	$32/225-w_5$	$w_1'-w_3-w_4'$	w_1-w_6
$1/2-w_2'$	$w_1-w_3'+w_4$	w_1'	$32/225-w_5'$	$w_1'-w_6'$	$w_1-w_3'-w_4$
$1/2$	w_1+w_7	$w_1'+w_7'$	$32/225$	$w_1'-w_7'$	w_1-w_7
$1/2+w_2'$	$w_1+w_3'+w_4$	$w_1'+w_6'$	$32/225+w_5'$	w_1'	$w_1+w_3'-w_4$
$1/2+w_2$	w_1+w_6	$w_1'+w_3+w_4'$	$32/225+w_5$	$w_1'+w_3-w_4'$	w_1
	$2w_1$	$2w_1'$	$64/225$	$2w_1'$	$2w_1$

where

$w_1 = (322-13\times 70^{1/2})/3600$	$w_1' = (322+13\times 70^{1/2})/3600$
$w_2 = ((35+2\times 70^{1/2})/63)^{1/2}/2$	$w_2' = ((35-2\times 70^{1/2})/63)^{1/2}/2$
$w_3 = w_2\,(452+59\times 70^{1/2})/3240$	$w_3' = w_2'\,(452-59\times 70^{1/2})/3240$
$w_4 = w_2\,(64+11\times 70^{1/2})/1080$	$w_4' = w_2'\,(64-11\times 70^{1/2})/1080$
$w_5 = 8w_2\,(23-70^{1/2})/405$	$w_5' = 8w_2'\,(23+70^{1/2})/405$
$w_6 = w_2-2w_3-w_5$	$w_6' = w_2'-2w_3'-w_5'$
$w_7 = w_2\,(308-23\times 70^{1/2})/960$	$w_7' = w_2'(308+23\times 70^{1/2})/960$

Appendix III

In this Appendix we provide the programming structure of the BPIRK family of methods implemented in this thesis. A brief description for each routines is also listed.



bpirk() : control routine for the solution procedure

exact(t, y) : routine to compute the exact solution y at point t

get_d(d) : routine to get the M-vector d

initial() : initialize the calculation

inverse() : compute the inverse of a nonsingular matrix

U_predictor(U_old) : predict the solution value

U_corrector(d, U_old, U_new): update the solution

get_Y(d) : compute the approximation value y

RHS(t, y, f) : compute the right hand side value at (t, y) and return the result in f

parameter() : input the parameters m, M, t, y, h, eps error, accuracy and t_final

get_V() : compute matrix V

get_W(d) : compute matrix W

get_PQR(d) : compute matrices P, Q, and R

get_PQR_prime(d) : compute matrices $\overline{P}$, $\overline{Q}$, and $\overline{R}$

RK: set the base RK formula

dVW: control routine for the matrices V, W and d

UY: routine to update U and compute solution value Y

error: routine to compute error and control accuracy

gauss_seidel: routine to solve the linear equations using Gauss-Seidel iteration

Bibliography

[1] O. Abou-Rabia, L. G. Birta and M. Chen, A Comparative Evaluation of BPC and PPC Methods for Parallel Solution of ode's, Trans. of Society for Computer Simulation, 6, No. 4(1989), pp. 265-290

[2] R. Alexander, The Modified Newton Method in the Solution of Stiff Ordinary Differential Equations, Math. Comput. 57(1991), pp. 673-701

[3] C. T. H. Baker and C. Phillips, The Numerical Solution of Nonlinear Problems, Clarendon Press, Oxford, 1981

[4] T. A. Bickart and Z. Picel, High Order Stiffly Stable Composite Multistep Methods for Numerical Integration of Stiff Differential Equations, BIT, 13(1973), pp. 272-286

[5] L. G. Birta and O. Abou-Rabia, Parallel Block Predictor-Corrector Methods for ode's, IEEE Trans. Computer, C-36, No. 3(1987), pp.299-311

[6] L. G. Birta and L. Yang, An Evaluation of Some $P(EC)^mE$ Methods for the Parallel Solution of ODE's, Summer Computer Simulation Conference, Boston (1993), pp. 43-48

[7] K. Burrage, A Special Family of Runge-Kutta Methods for Solving Stiff Differential Equations, BIT, 18(1978), pp. 22-41

[8] K. Burrage, Predictor-Corrector Methods for Solving ODEIVP's, Applied Numerical Mathematics, 11(1993), pp.125-141

[9] J. C. Butcher, Implicit Runge-Kutta Processes, Math. of Comput. 18(1964), pp. 50-64

[10] C. H. Cooke, On Stiffly Stable Implicit Linear Multistep Methods, SIAM J. Numer. Anal. 9(1972), pp. 29-34

[11] K. Dekker and J. G. Verwer, Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland, 1984

[12] M. A. Franklin, Parallel Solution of Ordinary Differential Equation, IEEE Trans. Comput., C-27, No. 5(1978)

[13] C. W. Gear, The Potential for Parallelism in Ordinary Differential Equations, Tech. Report UIUC-DCS-R-86-1246, Computer Science Dept., Univ. of Illinois, Urbana, IL, 1986

[14] C. W. Gear, Parallel Methods in Ordinary Differential Equations,Techniccal Report UIUC-DCS-R-87-1396, Computer Science Dept., Univ. of Illinois, Urbana, IL, 1987

[15] N. Ghasem-Aghaee, L. G. Birta and D. King, Practical Performance of Some Parallel Variable Stepsize ODE Solvers, Summer Computer Simulation Conference, San Diego (1994), pp3-8

[16] E. Hairer, S. P. Nørsett and G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer Verlag, 1987

[17] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II -- Stiff and Differential-Algebraic Problems, Springer-Verlag, 1991

[18] G. Hall and J. M. Watt, Modern Numerical Methods for Ordinary Differential Equations, Clarendon Press, Oxford, 1976

[19] P. J. van der Houwen and B. P. Sommeijer, Parallel Iteration of High-Order Runge-Kutta Methods with Stepsize Control, J. of Comput. and Applied Mathematics 29(1990), pp.111-127

[20] P. J. van der Houwen and N. huu Cong, Parallel Block Predictor-Corrector Methods of Runge-Kutta Type, Applied Numerical Mathematics 13(1993), pp. 109-123

[21] P. J. van der Houwen, B. P. Sommeijer and W. Couzy, Embedded Diagonally Implicit Runge-Kutta Algorithms on Parallel Computers, Math. Comput., 58(1992), pp. 135-159

[22] K. R. Jackson and S. P. Nørsett, The Potential for Parallelism in Runge-Kutta Method. Part 1: RK Formulas in Standard Form, SIAM J. Numer. Anal. 32(1995), pp. 49-82

[23] K. R. Jackson, A Survey of Parallel Numerical Methods for Initial Value Problems for Ordinary Differential Equations, IEEE Trans. Magnetics, 2(1991), pp. 3792-3797

[24] K. R. Jackson and S. P. Nørsett, The Potential for Parallelism in Runge-Kutta Method. Part 2: RK Predictor-Corrector Formulas, In Preparation

[25] K. R. Jackson, A. Kværnø, and S. P. Nørsett, An Analysis of the Order of Runge-Kutta Methods That Use an Iterative Scheme to Compute Their Internal Stage Values, BIT, Submitted

[26] L. N. Katz, M. A. Franklin and A. Sen, Optimally Stable Parallel Predictors for Adams-Moulton Correctors, Comput. Math. Appl., 3(1977), pp.217-233

[27] E. Lelarasmee, A. Ruehli and A. L. Sangiovanni-Vincentelli, The Wave-Form Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits, IEEE Trans. on Computer-Aided Design, Vol. CAD-1 (1982), pp.131-145

[28] W. L. Miranker and W. Liniger, Parallel Methods for the Numerical Integration of Ordinary Differential Equations, Math. Comput. 21(1967), pp.303-320

[29] O. Nevanlinna and F. Odeh, Remarks on the Convergence of Waveform Relaxation Method, Numer. Funct. Anal. and Optimiz., 9(1987), pp. 435-445

[30] S. P. Nørsett and H. H. Simonsen, Aspects of Parallel Runge-Kutta Methods, in Numerical Methods for Ordinary Differential Equations, A. Bellen, C. W. Gear, and E. Russo, eds., Lecture Notes in Mathematics #1386, Springer-Verlag, Berlin, 1989, pp. 103-107

[31] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, 1970

[32] J. M Ortega, Matrix Theory : A Second Course, Plenum Press, 1987

[33] A. M. Ostrowski, Solution of Equations in Euclidean and Banach Spaces, Academic Press,1973

[34] J. B. Rosser, A Runge-Kutta for All Seasons, SIAM Review, 9(1967), pp.417-452

[35] L. F. Shampine and H. A. Watts, Block Implicit One-Step Methods, Math. Comput. 23(1969), pp.731-740

[36] H.W. Tam, Parallel Methods for the Numerical Solution of Ordinary Differential Equations, Report No. UIUC-DCS-R-89-1516, Computer Science Dept, Univ. of Illinois, Urbana, IL, 1989