Dottorato di Ricerca in Matematica. XV ciclo, 2000-2004.
Università degli studi di Firenze

Tesi di Dottorato di
Cecilia Magherini

Numerical Solution of Stiff ODE-IVPs via Blended Implicit Methods: Theory and Numerics.

Direttore della Ricerca: Prof. Luigi Brugnano

Coordinatore del dottorato: Prof. Mario Primicerio

Contents

Preface

Ordinary Differential Equations (ODEs) play a central role in the mathematical modelling of real world phenomena. The solution of such equations allows to find answers to such questions as how a physical system evolves or what are the possible effects of changes in the system. In general, it is extremely difficult, if not impossible, to obtain an analytic solution of an ODEs. It is for this reason that the research concerning numerical methods for the approximate solution of such equations became so important. In particular, the solution of Initial Value Problems (IVPs) for ODEs has been, and continues to be, one of the most active field of investigation in Numerical Analysis. This is shown by the very rich amount of significant contributions during the last fifty years. In addition, many of the obtained results have been collected in several books, among which we quote [5, 20, 25, 35, 47, 58, 59, 76, 94].

Across the years, the required properties for a numerical method have had an interesting evolution. Indeed, until the fifties, accuracy requirements were considered as the most important for the methods. After that, stability requirements became focal, in particular in connection with the numerical solution of stiff problems. More recently, attention has been devoted to methods well suited for particular differential problems (like, for example, Delay Differential Equations [8], Hamiltonian problems [61, 93], and Stochastic Differential Equations [26]), and to methods well suited for an efficient implementation on modern computers, including parallel computers. In the latter context, properties of the methods such as the definition of efficient splittings, degree of parallelism, etc. have become focal, especially in connection with the solution of large-size problems, and the present dissertation deal with this topic. The thesis, in fact, is devoted to the so-called *Blended Implicit Methods*. In addition to classical requirements, such as high order of accuracy and "good" stability properties, the latter are methods defined in order to favourably meet implementation requirements. The generated discrete problem, in fact, may be efficiently solved by means of an iterative procedure based on a corresponding nonlinear splitting which is "naturally" defined. The main result of the developed research consists in the new code `BiM` for the numerical solution of stiff problems.

The thesis is organized as follows. Chapter 1 is devoted to a brief introduction on the reference continuous problem and on numerical methods for its approximate solution. Some of the most important results concerning the theory of numerical methods for ODEs are also reported, in particular in connection with the solution of stiff problems.

Chapter 2 is devoted to discuss the most efficient techniques currently used for the implementation of Block Implicit Methods. In more details, the issue of the solution of the discrete problem generated, at each step of integration, by a method in that class is addressed.

Blended Implicit Methods are then presented in Chapter 3 together with the linear analysis of convergence of the associated iteration, which they naturally define, for the solution of the discrete problem.

The implementation strategies, used in the development of the variable-stepsize, variable-order code `BiM`, are, then, discussed in full details in Chapter 4. It will be shown that almost all of such strategies are supported and justified by the results obtained through the previously mentioned linear analysis of convergence.

The numerical results obtained by using the new code are reported and analyzed in Chapter 5. In particular, such results are compared with those provided by some of the best codes for stiff ODEs currently available. Finally, some directions for future researches concerning Blended Implicit Methods are briefly sketched.

Acknowledgements

I would like to take this opportunity to thank my supervisor Prof. Luigi Brugnano for his support and valuable advice. He invested a lot of time in me during this period of research and he helped me an incredible amount to shape my ideas on how research should be approached. I would also like to thank him for having believed in me.

Chapter 1

Introduction

This chapter is intended to present the basic notions concerning numerical methods for the approximate solution of ODEs. In particular, some of the most important results concerning the theory of Linear Multistep Formulae and Runge-Kutta methods are recalled. Stiff problems and their numerical solution are then, briefly, discussed in the last section.

1.1 The reference problem

The reference problem of the thesis is the first-order ODE

$$y'(t) = f(t, y(t)), \qquad t \in [t_0, T]. \tag{1.1}$$

In the previous equation one can distinguish the independent variable t which, often, in the described physical system, represents the *time* and the dependent variable $y(t)$ which constitute the solution of the problem. Frequently, $y(t)$ is a vector valued function, i.e.,

$$y(t) : \mathbb{R} \to \mathbb{R}^m, \qquad f(t, y(t)) : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m,$$

where m is the dimension of the system.

In general, the solution of (1.1), in the case it exists, is not unique. An additional requirement on the solution is necessary to obtain its uniqueness. One of the most widely used is, certainly, to prescribe the value the solution must assume at the initial time t_0. The corresponding problem

$$\begin{cases} y'(t) = f(t, y(t)), & t \in [t_0, T], \\ y(t_0) = y_0 \in \mathbb{R}^m, \end{cases} \tag{1.2}$$

where y_0 is the prescribed initial value, is known as an Initial Value Problem (IVP) for the ODE (1.1). This kind of problems occur very frequently in the applications since, in many cases, the state of the system is known at a

certain time and one is interested to know the state at a certain time in the future.

The existence and uniqueness of the solution of the IVP (1.2) are (locally) guaranteed by the following well-known theorem.

Theorem 1.1 *Suppose that in the region* $D \subset \mathbb{R}^{m+1}$, *defined by*

$$D = \{(t, y) : |t - t_0| < a, ||y - y_0|| < b\},$$

the function $f(t, y)$ *is continuous and satisfies the* Lipschitz condition

$$||f(t, y) - f(t, z)|| \leq L||y - z||.$$

Then, there exists a unique solution of problem (1.2). Moreover, if

$$M \equiv \sup_{(t,z)\in D} \left(||f(t, z)||\right),$$

the solution is defined in the interval $|t - t_0| \leq \min(a, b/M)$.

In the following, the hypotheses of the previous theorem will be always assumed to be satisfied.

1.2 Numerical Methods for ODEs

The numerical solution of the IVP (1.2) is usually carried out by formally executing the following three steps:

1. the definition of a suitable discrete set (or *mesh*) $\{t_n\}_{n=0}^{n=N}$ in the interval $[t_0, T]$;
2. the replacement of the continuous problem by a discrete one, defined on such a discrete set;
3. the solution of the discrete problem.

Concerning the first step, the mesh may be predetermined or, as it happens more frequently, generated dynamically during the integration process. Actually, the problem of appropriately select the mesh points $\{t_n\}_{n=0}^{n=N}$ plays a central role on the possibility of obtaining, in an efficient way, a numerical approximation to the solution of the differential equation. This argument will be addressed in details in Chapter 4. Until then, for the sake of simplicity, the simplest mesh, given by the following set of uniformly distributed grid-points in $[t_0, T]$,

$$t_n = t_0 + nh, \quad n = 0, 1, \ldots, N, \qquad h = \frac{T - t_0}{N}, \tag{1.3}$$

will be always considered. The parameter h in (1.3) is often called the *step-size* or the *steplength*.

The third point, in the previous scheme, may either be a trivial or a difficult task according to the discrete problem defined in the second step. Actually, the main subject of the present dissertation will be the definition of efficient techniques for its solution.

Finally, the discrete problem, replacing the continuous one on the discrete set, strictly depends on the particular numerical method. As a matter of fact, the latter defines the "rules" used in performing such a replacement. From an historical point of view, the first method for ODEs is known as the *explicit Euler method* due to Euler in the early days of calculus (1768). The discrete problem generated by such method is the following one,

$$y_{n+1} = y_n + h\, f_n, \qquad f_n \equiv f(t_n, y_n), \qquad n = 0, \ldots, N-1, \tag{1.4}$$

where y_n represents, for each n, the numerical approximation of the solution at t_n, i.e.,

$$y_n \approx y(t_n).$$

Thus, at each step, such method assumes the knowledge of the incoming data y_n and the new approximation is obtained by assuming the slope of the y function constant throughout the interval $[t_n, t_{n+1}]$. Equivalently, the numerical integration proceeds by considering, after each step n, a new (local) IVP to be approximated, with initial value given by $y(t_n) = y_n$. This initiates the idea of *local error* where after each step the incoming data is assumed to be exact. The accuracy of the numerical solution is then measured by comparing the approximation, after one single step of integration, with the Taylor series expansion of the local exact solution, given by

$$y(t_{n+1}) = y(t_n) + h\, y'(t_n) + \frac{h^2}{2!} y''(t_n) + \cdots .$$

In particular, the Euler method is a first order one since it agrees with such an expansion up to the first power of h. According to the approach adopted for increasing the accuracy of the approximate solution, nowadays numerical methods for ODEs may be subdivided into two main classes of methods:

- *Multistep methods;*
- *One-step (Multistage) methods.*

Multistep methods obtain higher accuracy by allowing the approximate solution at a point to depend on the values of the solution and of the derivatives before the immediately previous point. One-step (multistage) methods, instead, build up y_{n+1} from values of the solution, and the corresponding derivatives, at several internal points (or *stages*) between t_n and t_{n+1}.

1.3 Linear Multistep Formulae

The most popular multistep methods are, certainly, Linear Multistep Formulae (LMF), also called Linear Multistep Methods (LMMs). They generate discrete problems with the following general form:

$$\sum_{j=0}^{k} \alpha_j \, y_{n+j} = h \sum_{j=0}^{k} \beta_j \, f_{n+j}, \qquad n = 0, \ldots, N-k, \tag{1.5}$$

where $f_n \equiv f(t_n, y_n)$, and k is called the *stepnumber* of the method. Thus, a k-step LMF transforms the differential equation (1.2) in a linear, with respect to y_n and f_n, difference equation of order k. Usually an IVP for the continuous equation is solved by means of an IVP for the discrete one, that is, a set of k initial values,

$$y_0, \, y_1, \, \cdots \, y_{k-1}, \tag{1.6}$$

is always associated to (1.5). Since only y_0 is provided by the continuous problem, a starting procedure is required to obtain the remaining ones. Then, a recursive procedure may be applied to compute the overall numerical solution. In particular, *explicit* methods are, by definition, those methods having $\beta_k = 0$. In this case, the algorithm for the solution of the discrete problem turns out to be a trivial and cheap one. On the other hand, when $\beta_k \neq 0$ (i.e. the method is an *implicit* one) the solution of an algebraic equation in $\mathbb{R}^m$ is required, at each step, to get the new approximation.

A general theory concerning multistep methods was started by the work of Dahlquist [44, 45] and became famous through the classical book of Henrici [63]. In particular, in the 1956 paper [44], Dahlquist introduced the fundamental concepts of *consistency*, 0-*stability* and *convergence*. The latter property describes the asymptotical behaviour of the numerical solution, with respect to the continuous one, when an increasing number of mesh-points in $[t_0, T]$ is used. More precisely, a LMF is said to be *convergent* in $[t_0, T]$ if, starting from "sufficiently" accurate values and assuming that the discrete problem is solved exactly, it provides approximations such that,

$$\lim_{N \to \infty} \max_{n=k,\ldots,N} || \, y(t_n) - y_n \, || = 0, \qquad h = \frac{T - t_0}{N}.$$

When looking for the properties that a convergent LMF has to satisfy, it was found fundamental the convergence of the numerical method in correspondence of the following three problems:

- $y'(t) = 0, \quad y(0) = 0;$
- $y'(t) = 0, \quad y(0) = 1;$

- $y'(t) = 1, \quad y(0) = 0.$

In particular, by introducing the polynomials,

$$\rho(z) \equiv \sum_{j=0}^{k} \alpha_j z^j, \qquad \sigma(z) \equiv \sum_{j=0}^{k} \beta_j z^j, \tag{1.7}$$

it was found that convergence of a LMF for the first problem necessarily requires $\rho(z)$ to be a Von Neumann polynomial (i.e. all zeros of $\rho(z)$ lie in the unit disc and all zeros on the boundary are simple). LMF with such property are called *stable* or 0-*stable*. Concerning the second problem in the previous list, it can be proved that the numerical solution may converge uniformly to the continuous one, only in the case where

$$\rho(1) = 0. \tag{1.8}$$

The previous condition is, sometimes, referenced as the "*pre-consistency condition*". Finally, a 0-stable and pre-consistent LMF is able to compute the exact solution of the third problem, in the limit, only in the case where

$$\rho'(1) = \sigma(1). \tag{1.9}$$

A LMF is said to be *consistent* when it satisfies the *consistency conditions* (1.8)-(1.9). Therefore, convergence for a LMF requires both consistency and 0-stability. In [44] Dahlquist proved that the last two properties are, indeed, sufficient for convergence, thus leading to the well-known result:

$$\text{convergence} \qquad \Leftrightarrow \qquad \text{consistency} \quad + \quad \text{0-stability.}$$

For a general IVP, the residual obtained when the sequence $\{y(t_n)\}$, consisting of the values that the exact solution assumes at the mesh-points, is inserted into the discrete problem,

$$\tau_n \equiv \sum_{j=0}^{k} \alpha_j \, y(t_{n+j}) - h \sum_{j=0}^{k} \beta_j \, f(t_{n+j}, y(t_{n+j})), \tag{1.10}$$

is called the *truncation error* of the method. By considering the Taylor series expansion of the continuous solution at t_n, it can be seen that, for a consistent method, the truncation error depends, at least, quadratically on h. The *order of accuracy* for a LMF is, then, defined as the largest p such that

$$\tau_n = O(h^{p+1}).$$

The same Taylor expansion allows to prove that p is given by the largest integer such that the following *order conditions* hold true:

$$\sum_{j=0}^{k} \left(j^s \alpha_j - s j^{s-1} \beta_j \right) = 0, \qquad s = 0, 1, \ldots, p. \tag{1.11}$$

In this case, the truncation error can be expressed as

$$\tau_n = v_{p+1} h^{p+1} y^{(p+1)}(t_n) + O(h^{p+2}),$$

where

$$v_{p+1} \equiv \frac{1}{(p+1)!} \sum_{j=0}^{k} \left(j^{p+1} \alpha_j - (p+1) j^p \beta_j \right) \tag{1.12}$$

is called the *principal error coefficient* of the method (obviously, the continuous problem is assumed to be sufficiently smooth).

The following is a well-known result concerning the accuracy of the numerical solution provided by a method of order p (see, for example, [20, 63]).

Theorem 1.2 *If the continuous problem is sufficiently smooth and the initial conditions (1.6) are, at least, $O(h^p)$ accurate, then the numerical solution provided by a 0-stable LMF of order $p \geq 1$ is such that, for each n,*

$$\| y(t_n) - y_n \| \leq C h^p,$$

where the parameter C is independent of h.

Thought it is possible to find LMF of order $p = 2k$, in [44] Dahlquist proved the following restriction on the maximum attainable order of a 0-stable (and, therefore, convergent) LMF. This result is known as the *first Dahlquist barrier.*

Theorem 1.3 *A 0-stable k-step LMF has order not larger than $k+1$, if k is odd, and not larger than $k+2$, if k is even.*

Let us now, briefly, discuss some of the most famous families of LMF. The first one was derived in the 1883 paper by Bashforth and Adams, [2]: such methods are now known as the *Adams-Bashforth methods.* The basic idea, used in deriving such methods, has been that of using the fundamental theorem of calculus for a scalar equation,

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} y'(s)\, ds,$$

and then to approximate the integrand with the interpolating polynomial through (t_{n-k}, f_{n-k}), $\ldots$, (t_{n-1}, f_{n-1}). The obtained methods were, therefore, explicit. An implicit version of the Adams methods was also introduced

in the cited paper by Bashforth and Adams. However, such implicit methods were studied, in their own-right, in 1926 by Moulton in [82], and, nowadays, they are known as the *Adams-Moulton methods*. For each value of k, both the explicit and the implicit Adams methods are convergent methods of orders, respectively, $p = k$ and $p = k + 1$, [76].

In 1952, Curtiss and Hirshfelder introduced another important family of LMF known as the *Backward Differentiation Formulae (BDF)*, [43]. Like Adams methods are based on numerical integration, BDF are based on numerical differentiation. In fact, the discrete problem generated by such methods has the following form:

$$f_{n+k} = \frac{1}{h} \sum_{j=0}^{k} \alpha_j y_j.$$

It is well-known that the BDF are convergent methods of order $p = k$ provided that $k \leq 6$ (see, for example, [76]).

In order to define a "good" method, convergence is, obviously, a necessary requirement. However, there exist important differential problems for which such property is certainly not enough. Convergence, in fact, does not take into account the effects that perturbations, like, for example, the ones due to round-off errors, produce on the numerical solution. Moreover, by definition, convergence is a limit property for values of h approaching 0 and, on the contrary, in the practice, the used stepsize is a fixed nonzero value. The *midpoint method*,

$$y_{n+2} = y_n + 2\,h\,f_{n+1}, \tag{1.13}$$

is a classical example that is frequently used to show how, even for arbitrarily small stepsizes, convergence may not provide useful indications on the accuracy of the numerical solution. Such method, in fact, is convergent of order $p = 2$. In spite of this, when it is used to approximate the solution of the IVP,

$$y'(t) = 2 \cdot 10^4 \left(\mathrm{e}^{-t} - y(t)\right) - \mathrm{e}^{-t}, \quad t \in [0, 1], \qquad y(0) = 1, \tag{1.14}$$

"large errors" are obtained in the numerical solution computed in standard double precision, regardless the (nonzero) value of the stepsize (see, for example, [20]).

A theory for error propagation for a fixed value of the stepsize h, is, therefore, needed. The development of such a theory requires, in general, the analysis of the stability properties of solutions of nonlinear difference equations and, unfortunately, the available mathematical tools do not provide suitably simple instruments for this task. However, when the solution

belongs to a suitable neighbourhood of a uniformly asymptotically stable equilibrium point, the *first approximation stability theorem* may be applied thus allowing to confine the previous analysis to linear problems, [20]. In addition to this, a consideration on the interval of existence of the solutions is required. It is, in fact, obvious that the case where h is finite and $n \to \infty$, requires the existence of the continuous solution for all $t \geq t_0$ and the existence of the numerical solution for all $t_n = t_0 + nh$. The previous requirements are fulfilled when both the exact and the numerical solutions belong to a neighbourhood of a uniformly asymptotically stable constant solution. All such arguments justify the study of the methods on the well-known *Dahlquist test equation*

$$\begin{cases} y'(t) = \mu y(t), & t \geq t_0, \qquad \mathrm{Re}(\mu) < 0, \\ y(t_0) = y_0, \end{cases} \tag{1.15}$$

whose solution is given by $y(t) = y_0 \mathrm{e}^{\mu(t-t_0)}$. Therefore, the continuous problem, admits $y(t) \equiv 0$ as asymptotically stable equilibrium point. Moreover, from (1.5), one can verify that the discrete problem for (1.15) admits the constant solution $y_n \equiv 0$ as equilibrium point and that the corresponding stability properties are determined by the roots of the *stability polynomial* associated to the method (see (1.5) and (1.7)),

$$\pi(z, q) \equiv \rho(z) - q\sigma(z), \qquad q \equiv h\,\mu. \tag{1.16}$$

In particular, the zero sequence is an asymptotically stable equilibrium point, for the discrete problem, provided all the roots of $\pi(z, q)$ lie inside the unit disk (i.e. $\pi(z, q)$ is a Schur polynomial). This lead to the definition of the region $\mathcal{D}$ of *Absolute stability* for a LMF as the region of the complex q-plane for which $\pi(z, q)$ is a Schur polynomial.

A LMF is able to provide qualitatively correct results for (1.15) only in the case where $q \in \mathcal{D}$. In this context, the midpoint method (1.13) certainly represents a limit case, since its region of Absolute stability is empty. In other cases, like, for example, for the Adams methods (with the only exception of the Trapezoidal rule), the intersection of $\mathcal{D}$ with the left-half complex plane is a bounded region and, if the method is 0-stable, the origin belongs to the boundary of $\mathcal{D}$. When this happen, the stability properties of the numerical method determines an upper bound for the allowed stepsize.

In 1963, Dahlquist understood the great advantage gained, in solving certain classes of problems, by the use *A-stable* methods, namely methods with a stability region which includes all the left-half complex plane. As it will be discussed in Section 1.5, stiff problems represent an important class of differential problems, since frequent in the applications, whose numerical integration effectively requires the use of an A-stable method. However in

[46], the same author proved the well-known *second Dahlquist barrier* which states a severe restriction on the possibility of obtaining high order A-stable LMF. More precisely, the following results were proved in that paper.

Theorem 1.4 *There are no explicit LMF which are A-stable. The maximum order of an A-stable implicit LMF is two.*

In looking for "nearly" A-stable methods, the property of $A(\alpha)$-stability, with $\alpha \approx \pi/2$, turns out to be one of the most desirable. By definition, in fact, the previous property holds when the region of Absolute stability contains the sector

$$\mathbb{C}_\alpha \equiv \{q \in \mathbb{C} : |\pi - \arg(q)| \leq \alpha\}. \tag{1.17}$$

In such a case, the method is able to provide qualitatively correct results for all values of μ in (1.15) such that $|\pi - \mu| \leq \alpha$, without requiring any restriction on the stepsize. The already mentioned BDF are, for example, $A(\alpha)$-stable method for each $k \leq 6$, [76]. As a consequence, many numerical codes, designed for the solution of stiff differential problems, are based on such formulae or subsequent modifications of them [11, 12, 40, 65].

In attempting to circumvent the Dahlquist's barriers, many approaches have been adopted. Among them we quote the approach based on the use of higher derivatives of the solution, as in the case of the *Second Derivative Multistep Methods* of Enright [51]; the approach based on suitable combinations of two or more methods, as for the *Blended Multistep Methods* of Skeel and Kong [100], and the approach based on the use of further stages, additional nodes or off-step points, as in the case of the *Modified Extended BDF* of Cash [38].

Another important and recent contribution to the analysis of multistep methods is, certainly, due to Brugnano and Trigiante. In the 1998 book [20], the authors introduced *Boundary Value Methods (BVMs)*. The basic idea, on which such methods rely, is to adopt alternative choices for the additional conditions required by the discrete problem (1.5). In more detail, this is done by approximating the continuous IVP (1.2) by means of a discrete Boundary Value Problem (BVP). In the preface of that book, in fact, the authors write:

"*Even if initial value problems are easier in the realm of infinite precision arithmetic (i.e. real or complex numbers), boundary value problems are safer in the realm of finite precision*".

By means of an appropriate choice for the boundary conditions, methods with very good stability properties were then obtained. Among them, we mention the *Generalized Adams Methods* (GAMs) and the *Generalized Backward Differentiation Formulae* (GBDF).

1.4 Runge-Kutta Methods

Runge-Kutta (RK) methods are generally considered as the most popular one-step (multistage) methods. The first method adopting the "multistage philosophy" to obtain higher accuracy, is generally attribute to Runge in 1895, [91]. Further early contributions, to what are now known as Runge-Kutta methods, are those due to Heun, Kutta and Nyström, [64, 75, 85]. In particular, the famous fourth-order method in Kutta's paper is often referred to as *the* Runge-Kutta method.

At each step of integration, an r-stage Runge-Kutta method advances the numerical solution as follows:

$$y_{n+1} = y_n + h \sum_{i=1}^{r} b_i f(t_n + c_i h, y_{in}), \tag{1.18}$$

where

$$y_{in} = y_n + h \sum_{j=1}^{r} a_{ij}\, f(t_n + c_j h, y_{jn}), \qquad i = 1, \ldots, r. \tag{1.19}$$

Here, the quantities y_{in}, called the *internal stages*, represent approximations to the solution at the points $t_n + c_i h$, generally internal to the interval $[t_n, t_{n+1}]$. The coefficients of a RK method are, usually, collected into the following *Butcher array,*

$$\begin{array}{c|c} \mathbf{c} & \mathcal{A} \\ \hline & \mathbf{b}^T \end{array}$$

where,

$$\mathbf{c} \equiv \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix}, \qquad \mathbf{b} \equiv \begin{pmatrix} b_1 \\ \vdots \\ b_r \end{pmatrix}, \qquad \mathcal{A} \equiv \begin{pmatrix} a_{11} & \cdots & a_{1r} \\ \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rr} \end{pmatrix}.$$

The previous notation allows to rewrite the discrete problem (1.18)-(1.19) in the more compact form,

$$\begin{aligned} \mathbf{y}_n &= \mathbf{1} \otimes y_n + h(\mathcal{A} \otimes I_m)\mathbf{f}_n, & (1.20) \\ y_{n+1} &= y_n + h(\mathbf{b}^T \otimes I_m)\mathbf{f}_n, & (1.21) \end{aligned}$$

where I_m is the identity matrix of order m, $\mathbf{1} \equiv (1, \ldots, 1)^T \in \mathbb{R}^r$, and

$$\mathbf{y}_n \equiv \begin{pmatrix} y_{1n} \\ \vdots \\ y_{rn} \end{pmatrix}, \qquad \mathbf{f}_n \equiv \begin{pmatrix} f_{1n} \\ \vdots \\ f_{rn} \end{pmatrix}, \qquad f_{in} \equiv f(t_n + c_i h, y_{in}).$$

A RK method is called explicit when the matrix $\mathcal{A}$ is strictly lower triangular, implicit otherwise. As for LMF, the procedure for the solution of the discrete problem greatly simplifies in the case of explicit methods. This is much more true for RK methods since, at each step, the new approximation depends on r new unknowns in $\mathbb{R}^m$.

The *order* of accuracy, for a RK method, is defined on the base of the asymptotical behaviour, as h approaches 0, of the *local error*. The latter is given by the difference between the exact and the numerical solution after one step of integration, under the assumption of an exact starting value. In particular, a RK method has order p provided that, for a sufficiently smooth function f defining the continuous problem, there exist a constant C, independent of h, such that

$$\|y(t_0 + h) - y_1\| \leq C\, h^{p+1}.$$

The analysis of the order conditions for the coefficients of a RK method is definitely much more complicated, with respect to the same for LMF. The main reason is the fact that, in general, the numerical solution, at each step, is built up from the derivatives evaluated at stage values having a lower accuracy. The basic idea, used for such analysis, is to compare, term by term, the series expansions, in powers of h, for the exact and the numerical solutions at the end of a single step of integration. However, the terms involved in such expansions become greatly complicated quite soon and this has been one of the main difficulties encountered in the early time of the research on such methods. The major contribution concerning the analysis of the order conditions for RK methods is due to Butcher. In his 1963 paper [27], based on the earlier work by Gill [55] and Merson [80], he related the various terms involved in the Taylor series expansions, of both the exact and the approximated solution computed by a Runge-Kutta method, to the graphs of the so-called *rooted trees*. Making use of the resulting theory, in [30, 34], Butcher proved quite complicated relationships between the minimum number of stages r to obtain explicit methods of order $p > 4$.

In the 1964 paper [28], on implicit RK methods, Butcher introduced the so-called *simplifying assumptions* consisting in a set of conditions which, when satisfied, reduce, significantly, the number of conditions needed to obtain a method with a prescribed order. This, in turn, made it possible to derive methods of higher order. In [28], in fact, Butcher introduced implicit RK methods based on the Gaussian quadrature formulae of order $p = 2r$, while in [29] the same author introduced the Radau I and Radau II methods, of orders $p = 2r - 1$, and the Lobatto III methods of orders $p = 2r - 2$.

When a RK method is applied to the test equation (1.15), the obtained

numerical solution satisfies

$$y_{n+1} = g(q)\, y_n, \tag{1.22}$$

where it can be proved, (see [59]), that $g(q)$, called the *stability function* of the method, is given by

$$g(q) = \frac{\det\left(I_r - q\mathcal{A} + q\mathbf{1}\mathbf{b}^T\right)}{\det\left(I_r - q\mathcal{A}\right)}, \tag{1.23}$$

being I_r the identity matrix of order r. The region of absolute stability $\mathcal{D}$ for a RK method is, therefore, defined as

$$\mathcal{D} \equiv \{q \in \mathbb{C} : |g(q)| < 1\}.$$

Consequently, explicit RK methods always have a bounded stability domain since, for such methods, $g(q)$ is a polynomial (see (1.23)). Implicit methods, instead, have a rational stability function and methods of arbitrarily high order can be A-stable. In particular, in 1969, Ehle proved the implicit Gauss RK methods to be A-stable while the Radau I, Radau II and Lobatto III methods to be not, [49]. Moreover, Ehle took up the ideas of Butcher and constructed the well-known A-stable Radau IA, Radau IIA, Lobatto IIIA, and Lobatto IIIB methods. In the same year, the Radau IIA methods were found, independently, by Axelsson together with an elegant proof of their A-stability, [6]. The general definition of the Lobatto IIIC methods is due to Chipman [42]; see also the paper by Axelsson [7].

The linear stability theory, based on the analysis of the methods on the test equation (1.15), seems to suggest that methods with a stability domain which exactly coincides with the left-half complex plane (i.e. *perfectly A-stable* methods) have to be considered as "optimal" methods. The previous property, however, turns out to be not as desirable as it may appear. It can be proved, in fact, that the stability function of perfectly A-stable methods is such that

$$\lim_{q\to\infty} |g(q)| = 1.$$

This means that, when q is close to the real axis and has a very large negative real part, the continuous solution of (1.15) fast decays to zero while the modulus of the numerical solution is very slowly damped. Therefore, in order to reflect the behaviour of the continuous solution, one should have $|g(q)| \ll 1$ as $q \to -\infty$. This leads Ehle to introduce the following property for a method [49]:

Definition 1.1 *A method is called* L-stable *if it is A-stable and if, in addition,*

$$\lim_{q\to\infty} g(q) = 0.$$

1.5 Stiff Differential Equations

Stiff differential equations arise in a countless amount of applications and their numerical solution has challenged Numerical Analysts as well as Applied Mathematicians during the last fifty years.

The first appearance of the term "stiff", in connection with the numerical solution of ODEs, is in the paper by Curtiss and Hirschfelder [43] published in 1952. In that work, the authors showed that certain types of problems, arising from chemical kinetics, are best solved by means of appropriately selected numerical methods. The analysis carried out in that paper was the first example of the "tailoring" of the method to the properties of the continuous problem to be solved, which has become common practice nowadays. Since then, the phenomenon of stiffness for ODEs has been one of the most studied subject in Numerical Analysis. Nevertheless, a precise mathematical characterization of stiffness, able to cover the most important facets of the phenomenon, has not yet been given. As a matter of fact, in the Lambert book [76], five different definitions of stiffness can be found.

From the early time of the research on stiff problems, there has been a large agreement on the fact that stiffness occurs when very different time scales are present in a problem. The term itself, in fact, seems to derive from such peculiarity since it seems to descend from mechanical models of systems of weights connected with springs having very different rigidity constants (stiff constants). The solutions of the corresponding equations are, therefore, characterized by fast modes, corresponding to the effects of the stronger springs, and slow modes, corresponding to the effect of the soft ones.

The classical example, which is always used to discuss the phenomenon of stiffness, is the linear autonomous equation,

$$y'(t) = A\,y(t), \qquad t \in [t_0, T], \tag{1.24}$$

where the coefficient matrix A has distinct, real and negative eigenvalues,

$$\mu_{max} \equiv \mu_1 < \mu_2 < \ldots < \mu_m \equiv \mu_{min} < 0.$$

The general solution of such equation takes the form

$$y(t) = \sum_{i=1}^{m} c_i \mathrm{e}^{\mu_i\,(t-t_0)} \mathbf{v}_i,$$

where, for each i, $\mathbf{v}_i \in \mathbb{C}^m$ is an eigenvector corresponding to μ_i and the coefficient $c_i \in \mathbb{C}$ depends on the initial value $y(t_0)$. In particular, when the extreme eigenvalues of A are such that

$$|\mu_{max}| \gg |\mu_{min}|,$$

the general solution of (1.24) is made up by "fast" modes, corresponding to the eigenvalues of largest modulus, and "slows", modes corresponding to the smallest modulus ones. In such a case, to get a complete information on the system, it is necessary to keep integrating until the slowest modes became negligible. This requires to take, at least, $T - t_0 \approx |\mu_{min}|^{-1}$. On the other hand, the fast modes significantly contribute to the solution only during a very short initial period, say $[t_0, t_0 + |\mu_{max}|^{-1}]$. Therefore, the different time scales for (1.24), giving rise to the phenomenon of stiffness, are given by the modulus of the extreme eigenvalues and the *stiffness ratio*,

$$\frac{|\mu_{max}|}{|\mu_{min}|},$$

is traditionally used as a measure of the stiffness of the problem. More generally, when the spectrum of the coefficient matrix in (1.24) is contained in $\mathbb{C}^-$, the problem is stiff when the eigenvalues of A have very different real parts.

Looking at the solution curves of a stiff scalar equation, one often recognize a smooth "slowly varying" solution (the *steady-state* solution) which is approached by the other ones after a rapid *transient phase*. A well-known example, showing such behaviour, is provided by the following equation, [89]:

$$y'(t) = \mu(y(t) - \phi(t)) + \phi'(t), \qquad y(t_0) = y_0, \tag{1.25}$$

where $\mu \ll 0$ and $\phi(t)$ is a slowly varying smooth function. It is not difficult to verify that the corresponding solution curves are given by

$$y(t) = (y_0 - \phi(t_0))\mathrm{e}^{\mu(t-t_0)} + \phi(t),$$

(see the plots in Figure 1.1 for the case $\mu = -50$ and $\phi(t) = \cos(t)$). The different time scales, giving rise to the phenomenon of stiffness, are recognized to be $|\mu|$, which measure the rapidity at which $\phi(t)$ is approached by the other solutions, and a "measure" of the rate at which the solution $\phi(t)$ varies. The latter, in turn, often determines the required length of the integration interval for obtaining a complete information about the behaviour of the solution.

Probably, the difficulties in formulating a unifying definition of "stiff problems" are mainly due to the fact that it is better understood what goes wrong when numerical methods, not designed for such problems, are used to try to solve them. In the first line of the first section of the Hairer and Wanner's book [59], one of the most comprehensive on the subject, the authors write:

"*Stiff equations are problems for which explicit methods don't work*".

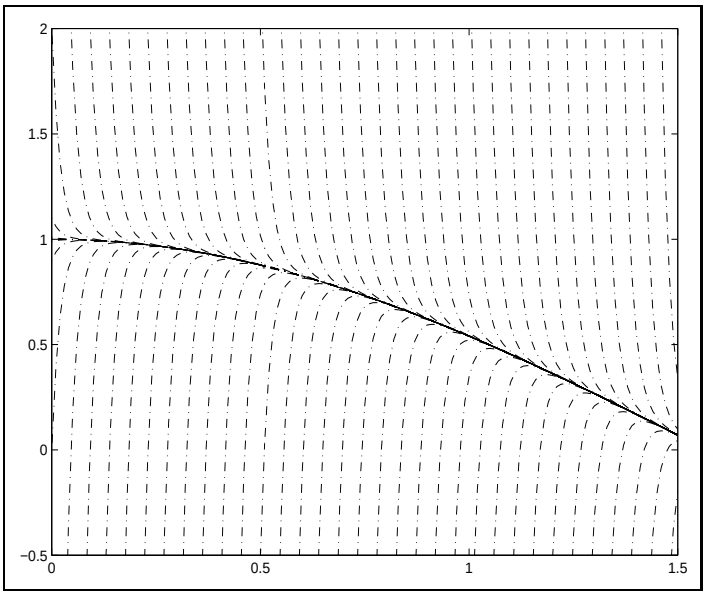


Figure 1.1: Solution curves for (1.25) with $\mu = -50$ and $\phi(t) = \cos(t)$.

Obviously, this is only an empirical definition and, consequently, not mathematically acceptable. Nevertheless, the authors refer to explicit methods since they always have a bounded stability domain. Moreover, an immediate consequence of the different time scales present in a stiff ODE is that such equation are best solved, both in terms of efficiency and of error accumulation, when an appropriate strategy for the definition of the discrete partition is adopted. On the base of the required accuracy for the numerical solution, the above strategy has to be able to select the most suitable value for the stepsize h. In particular, this implies the need for a fine mesh during the transient phase and the possibility of using a much more coarser mesh in the stationary one. The previous arguments, however, are based only on considerations concerning the required accuracy for the numerical solution. On the other hand, when a method with a bounded stability domain is used, the constraints on the stepsize, arising from the lack of stability properties, have to be respected by the stepsize variation strategy. In particular, when such types of methods are used to integrate stiff ODEs, the stability properties of the method often force the use of stepsizes which, in the stationary phase, are "unrealistically" small with respect to the smoothness of the continuous solution. For this reason, the numerical approximation of stiff equations requires the use of A-stable, and therefore implicit, methods.

Chapter 2

The implementation issue of Block Implicit Methods

In the recent years the implementation issue has become focal for numerical methods for ODEs. Indeed, since a number of stable, high order methods are currently available, one of the main reasons to use a method in place of another is given by its computational cost. In particular, for block implicit methods the main problem to be addressed for an efficient implementation consists in the definition of "suitable" strategies for the solution of the discrete problem generated at each step of integration. The present chapter is devoted to discuss the most efficient techniques currently used for such purpose.

2.1 Introduction

When applied to the IVP

$$\begin{cases} y'(t) = f(t, y(t)), & t \in [t_0, T], \\ y(t_0) = y_0 \in \mathbb{R}^m, \end{cases} \tag{2.1}$$

an r-Block Implicit Method generates, at each step of integration n, a discrete problem in the form:

$$F(\mathbf{y}_n) \equiv A \otimes I_m \mathbf{y}_n - hB \otimes I_m \mathbf{f}_n - \boldsymbol{\eta}_n = \mathbf{0}, \tag{2.2}$$

where the matrices $A, B \in \mathbb{R}^{r \times r}$ define the method, I_m is the identity matrix of order m, h is the stepsize and the vector $\boldsymbol{\eta}_n$ only depends on

already known quantities. The block vectors

$$\mathbf{y}_n = \begin{pmatrix} y_{1n} \\ \vdots \\ y_{rn} \end{pmatrix}, \qquad \mathbf{f}_n = \begin{pmatrix} f_{1n} \\ \vdots \\ f_{rn} \end{pmatrix}, \qquad f_{in} = f(t_{in}, y_{in}),$$

contain r values of the discrete solution or the internal stage values of the step.

Instances of methods falling in this class are RK methods, a number of General Linear methods [35, 58, 59] and, more recently, block BVMs [20].

In the following, we shall always assume the two matrices A and B to be nonsingular so that the underlying method is an implicit one. More precisely, in the case of RK methods with *explicit* stages, like the Lobatto schemes, r equals the number of *implicit* stages and the matrix B is obtained by considering only the corresponding coefficients, [68].

First of all, it must be observed that in (2.2) a multiplication from the left by $\left(A^{-1} \otimes I_m\right)$ of both sides of the equation allows to normalize the first coefficient matrix to the identity. Nevertheless, sometimes it could be preferable to keep the more general formulation (2.2), as in the case, for example, of block BVMs [20]. Moreover, as discussed in full details in the next chapter, the more general formulation in (2.2) presents some advantages in discussing the implementation issues of the method.

For a nonlinear differential equation, the implementation of block implicit methods requires, therefore, the solution of an algebraic equation of size rm at each step of integration. This is the reason for which, for many years, it was generally believed that, in spite of their better stability properties, block implicit methods would never be competitive with respect, for example, to $A(\alpha)$-stable LMF with $\alpha \approx \frac{\pi}{2}$.

As a consequence, the problem of devising efficient algorithms for the solution of (2.2) has been extensively studied for various classes of methods (see, e.g.,[4, 14, 33, 68, 69]), also with reference to the implementation on different computer platform [25, 52, 53], and this is still an active field of research in the area. In the sequel, for sake of simplicity, the step index n will be always omitted since the reported analysis equally applies to each step of integration. Therefore, without loss of generality, we can analyze the first step of integration.

During the early time of stiff computation people were usually thinking of a simple fixed-point iteration to solve (2.2). Nevertheless a similar

approach essentially transforms the method into an explicit one, thus destroying the good stability properties of the underlying implicit one.

Then, the use of procedures based on Newton's type methods, in particular those based on the simplified Newton method, and procedures based on suitable nonlinear splittings for the nonlinear equation (2.2) has become a common practice. The following sections are devoted to a review of such implementation techniques.

2.2 The simplified Newton method

The simplified Newton method is characterized by the following approximation of the Jacobian matrix of the function F in (2.2)

$$J_F \approx (A \otimes I_m - hB \otimes J_0),$$

where

$$J_0 \equiv \frac{\partial f}{\partial y}(t_0, y_0)$$

denotes the Jacobian matrix of f at the initial point of the step. The discrete problem (2.2) is, therefore, solved by means of the following iteration:

$$\begin{cases} (A \otimes I_m - hB \otimes J_0)\Delta\mathbf{y}^{(i)} = -F(\mathbf{y}^{(i)}), \\ \mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}, \qquad i = 0, 1, \ldots. \end{cases} \tag{2.3}$$

Obviously, the constant coefficient matrix in (2.3),

$$M \equiv (A \otimes I_m - hB \otimes J_0), \tag{2.4}$$

has to be evaluated only once and, in addition, this requires only one evaluation of the Jacobian matrix of f. In spite of this, the use of direct solvers for solving the linear systems in (2.3) turns out to be extremely costly since the factorization of the $rm \times rm$ matrix M is required. If we do not consider (for sake of simplicity) the terms due to function and Jacobian evaluations then, at least for large-size problems, the leading term in the arithmetic complexity of the iteration (2.3) is given by $\frac{2}{3}(r \cdot m)^3$ flops, where we count as *one flop* one of the four basic floating point binary operations with real quantities. This cost is considerably higher if compared, for example, with the complexity of the procedure for the solution of the discrete problem generated by a LMF.

The first attempts to reduce the cost for the solution of the Newton systems (2.3) were based on the idea of using methods with simple structured

matrices A and B. In particular, since for RK methods $A = I_r$, see (1.19), the research was focused on methods with a lower triangular coefficient matrix B, [3, 83]. The obtained methods have been variously named across the years. Today, it is usual to call them *diagonally implicit Runge-Kutta methods* (DIRK) or, in the case of equal diagonal entries, *singly diagonally implicit Runge-Kutta methods* (SDIRK). The above methods have the obvious advantage of allowing to solve the linear systems in (2.3) in r successive stages with only m-dimensional systems to be solved at each stage. However, they also have disadvantages. One of the most important is given by their low stage order which, in view of the order reduction phenomenon (see [89]), make them not really appropriate for the solution of stiff problems.

According to the general ways of solving linear systems, that is by using direct or iterative procedures, it is possible to classify the currently used algorithms into the following main categories:

- diagonalization (or block diagonalization) of the matrices A and B;
- definition of suitable linear splittings for the systems in (2.3);

The following sections are devoted to discuss the two possibilities.

2.2.1 Diagonalization

The algorithm described in the present section has been proposed by Butcher in his 1976 paper [33] on the implementation of implicit RK methods. As already observed, each Newton iteration in (2.3) requires the solution of the linear system (the index i has been omitted for simplicity),

$$M\Delta\mathbf{y} = -F(\mathbf{y}), \tag{2.5}$$

where, for RK methods (see (2.4)), the matrix M becomes

$$M = I_r \otimes I_m - hB \otimes J_0.$$

The main idea of the algorithm proposed in [33] has been that of using the Jordan form of the matrix B to define two nonsingular $r \times r$ matrices P and Q such that

$$PQ = \begin{pmatrix} 1 & & & \\ \varepsilon_2 & 1 & & \\ & \ddots & \ddots & \\ & & \varepsilon_r & 1 \end{pmatrix}, \qquad PBQ = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_r \end{pmatrix}.$$

Here $\varepsilon_i = 0, 1$ while $\sigma(B) = \{\lambda_1, \ldots, \lambda_r\}$ represents the spectrum of B. The system (2.5) was then transformed into the following equivalent one,

$$\tilde{M}\Delta\tilde{\mathbf{y}} = -\tilde{F}(\mathbf{y}), \tag{2.6}$$

where

$$\Delta\tilde{\mathbf{y}} \equiv \left(Q^{-1}\otimes I_m\right)\Delta\mathbf{y}, \qquad \tilde{F}(\mathbf{y}) \equiv (P\otimes I_m)\,F(\mathbf{y}), \tag{2.7}$$

$$\tilde{M} \equiv (PQ)\otimes I_m - h(PBQ)\otimes J_0. \tag{2.8}$$

The matrix $\tilde{M}$ is therefore composed by diagonal blocks of the form $I_m - h\lambda J_0$ with possibly complex λ and subdiagonal blocks of either the zero or the identity matrix. Since each of the transformations in (2.7) requires $O(m)$ operations, the overall advantage, in terms of arithmetic complexity, of this procedure is determined by the spectrum of the matrix B. In particular, the higher the number of real and multiple eigenvalues of the matrix B, the lower the computational cost for solving (2.6). In order to take full advantage from the Butcher procedure, in [22, 84] the so-called *singly implicit Runge-Kutta methods* (SIRKs), namely methods with a real one-point spectrum matrix B, were introduced. However, the obtained methods were less favourable than Runge-Kutta methods with complex eigenvalues in terms of accuracy and stability properties.

A slight different procedure is the one currently used in the RADAU5 and RADAU codes both implementing the Radau IIA implicit Runge-Kutta methods [59]. The basic idea, used in such codes, essentially consists in reducing B to a block diagonal matrix by means of a real similarity transformation. That is

$$T^{-1}BT = \Lambda \equiv \begin{pmatrix} \Lambda_1 & & \\ & \ddots & \\ & & \Lambda_s \end{pmatrix}, \tag{2.9}$$

where $\Lambda_i = \lambda_i$, if λ_i is a real eigenvalue of B, while

$$\Lambda_i = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix},$$

if $\lambda_i = \alpha \pm \mathrm{i}\,\beta$ is a complex conjugate pair. In addition, the linear subsystem arising in (2.6), in correspondence of a complex conjugate pair $\alpha \pm \mathrm{i}\,\beta$, given by,

$$\begin{pmatrix} I_m - h\alpha\, J_0 & h\beta\, J_0 \\ -h\beta\, J_0 & I_m - h\alpha\, J_0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

is transformed in the following equivalent m-dimensional complex one:

$$\Big((I_m - h\alpha\, J_0) - \mathrm{i}\,h\beta\, J_0\Big)(u_1 + \mathrm{i}\,u_2) = (z_1 + \mathrm{i}\,z_2).$$

It follows that, if the matrix B has k distinct real eigenvalues and $l = (r-k)/2$ distinct conjugate pairs, the corresponding procedure requires the factorization of k $m \times m$ real matrices and l complex matrices of the same dimension. As a consequence, since the complexity for the factorization of complex matrices is approximately 4 times the complexity for the factorization of real matrices, the leading term in the arithmetic complexity of the overall procedure is approximately given by

$$\frac{2}{3}(k+4l)m^3 \quad \text{flops.}$$

As an example, the spectrum of the matrix B corresponding to the Radau IIA method of order 5 is composed by one real eigenvalue and one complex conjugate pair. In such a case, when compared to the $\frac{2}{3}(3 \cdot m)^3$ operations required for the factorization of M in (2.4), a factor of about 5 has been gained.

The approach discussed in the present section is very popular for RK methods since the matrix A in (2.2) is the identity matrix. The generalization to block methods with nonsingular matrices A and B requires such matrices to be diagonalizable by means of the same similarity transformation.

However, a severe drawback of the described approach consists in the possible ill-conditioning of the matrix T in (2.9). This is especially true for methods with large blocksize r (see, for example, [9, 10]).

2.2.2 Linear Splittings

The main idea of the procedure described in this section consists in using an iterative solver for the Newton systems in (2.3). More precisely, instead of solving the linear system for $\Delta\mathbf{y}^{(i)}$,

$$(A \otimes I_m - hB \otimes J_0)\Delta\mathbf{y}^{(i)} = -F(\mathbf{y}^{(i)}), \tag{2.10}$$

required by the *outer* (or *primary*) iteration in (2.3), the following *inner* (or *secondary*) iteration is applied

$$\begin{aligned}(A^* \otimes I_m - hB^* \otimes J_0)\,\Delta\mathbf{y}^{(j,i)} = \\ \Big((A^* - A) \otimes I_m - h(B^* - B) \otimes J_0\Big)\Delta\mathbf{y}^{(j-1,i)} - F(\mathbf{y}^{(i)}),\end{aligned} \tag{2.11}$$

$j = 1, \ldots, \nu$, where ν is a suitable, possibly small, integer while A^* and B^*

are two nonsingular $r \times r$ matrices. The vector $\Delta\mathbf{y}^{(\nu,i)}$ is then adopted as the solution of (2.10) and the numerical solution is updated as (see (2.3)),

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \Delta\mathbf{y}^{(\nu,i)}.$$

Concerning the choice of the splitting matrices A^* and B^* in (2.11), the competitiveness of the inner iteration is the commonly used criterion for their definition. As a consequence, it must be observed, first of all, that a simple structure is a necessary requirement for them, since the arithmetic complexity for the solution of the linear systems in (2.11) is expected to be much lower than that of the original one.

However, this is certainly not enough for competitiveness. As matter of fact, the efficiency of such inner-outer iteration scheme strictly depends on the convergence properties of the inner iteration. Concerning this point, the common practice [68, 69] is to carry out a *linear analysis of convergence* of the iteration, thus studying its behaviour on the linear problem

$$y'(t) = J\, y(t).$$

For such problem, the simplified Newton method globally converges in one iteration. It follows that one has to consider only the behaviour of the inner iteration. Moreover, since the iteration matrix in (2.11) is a function of the Jacobian matrix J, convergence is determined by the behaviour of the iteration matrix in correspondence of each eigenvalue μ of J. The scalar test equation

$$y'(t) = \mu\, y(t), \qquad \mu \in \mathbb{C}, \tag{2.12}$$

is, therefore, always adopted as the reference problem for the linear analysis of convergence. In such a case, by setting, as usual,

$$q = h\mu,$$

the iteration (2.11) will converge to the solution of (2.10) provided that the spectral radius, say $\rho(q)$, of the *iteration matrix* or *amplification matrix*,

$$Z(q) \equiv I_r - (A^* - qB^*)^{-1}(A - qB), \tag{2.13}$$

is smaller than 1. The *region of convergence* of the iteration is therefore defined as

$$\Gamma = \{q \in \mathbb{C} : \quad \rho(q) < 1\}. \tag{2.14}$$

Obviously, it would be desirable the region of convergence to be as large as possible and the ideal case would be that of a globally convergent inner iteration. Nevertheless, this cannot be accomplished by using constant splitting

matrices A^* and B^* with a suitably "simple" structure. The region Γ is, therefore, always strictly contained in $\mathbb{C}$. A first reasonable requirement, that is always demanded, is the convergence of the iteration for all values of $q \approx 0$. This is the case, for example, when the spectral radius $\rho(q)$ is such that

$$\rho(0) = 0, \qquad \rho(q) \text{ analytical in } \mathcal{B}(0,\varepsilon), \tag{2.15}$$

where $\mathcal{B}(0,\varepsilon)$ is a suitable neighbourhood of the origin. Under such assumptions, in fact, for each values of μ, the procedure is effective provided a sufficiently small stepsize h is used. Moreover, the assumptions (2.15) on $\rho(q)$ do not impose severe restrictions on the possible choices of the splitting matrices. For example, they obviously hold true when $A^* = A$. In the sequel, therefore, we shall always assume them to be verified.

Evidently, when the continuous problem is a stiff differential equation, additional requirements on the convergence of the inner iteration are necessary. As a matter of fact, the use of A-stable methods has been preferred since their stability properties do not impose any restriction on the stepsize h to be used when $\mathrm{Re}(\mu) < 0$. In order not to introduce restrictions on h at the implementation level of the method, it is therefore desirable the use of iterative procedures converging for all values of $q \in \mathbb{C}^-$. These arguments lead to the following definitions.

Definition 2.1 *The iteration (2.11) is said to be* A-convergent *if*

$$\mathbb{C}^- \subseteq \Gamma.$$

Similarly, the iteration is said to be $A(\alpha)$-convergent *if the sector* $\mathbb{C}_\alpha$*, defined in (1.17), is contained in* Γ*.*

Clearly, the iteration (2.11) cannot be A-convergent if the pencil $A^* - qB^*$ is singular for some values of $q \in \mathbb{C}^-$. Therefore, the following condition on the spectrum of the matrix pencil is a pre-requirement for the A-convergence of the iteration:

$$\lambda(A^*, B^*) \equiv \{q \in \mathbb{C} : \det(A^* - qB^*) = 0\} \subset \mathbb{C}^+. \tag{2.16}$$

The splitting matrices are always chosen in order to satisfy this requirement. Moreover, when (2.16) holds true, $\rho(q)$ is analytical in $\mathbb{C}^-$ so that, by the maximum-modulus principle, A-convergence is equivalent to require

$$\rho^* \equiv \max_{\arg(q)=\frac{\pi}{2}} \rho(q) < 1. \tag{2.17}$$

The parameter ρ^* is called the *maximum amplification factor* of the iteration. This is a first important evaluation parameter measuring the convergence

properties of the iteration. In fact, since it refers to the worst case situation, when $\mathrm{Re}(\mu) < 0$, it serves as an indicator of the robustness of the procedure. Nevertheless, when stiff differential equations are to be solved, one has to consider that stiff and nonstiff modes are present in the iteration error components. Therefore, it is also very important to consider the behaviour of the iteration for values of q close to 0 and values of q approaching infinity. Then, the following parameters are defined to measure additional convergence properties of the iteration:

- the *nonstiff amplification factor*,

$$\tilde{\rho} \equiv \lim_{q \to 0} \frac{\rho(q)}{|q|}; \tag{2.18}$$

- the *stiff amplification factor*,

$$\rho^{(\infty)} \equiv \lim_{q \to \infty} \rho(q). \tag{2.19}$$

Concerning the nonstiff amplification factor, it must be stressed that, since

$$q \approx 0 \qquad \Rightarrow \qquad \rho(q) \approx \tilde{\rho}\ |q|,$$

a moderate value for $\tilde{\rho}$ would be desirable. Regarding the stiff amplification factor, instead, in [68, 69] the authors underlined the fact that a strong damping of the stiff error components is crucial for a fast overall convergence of the iteration. The competitiveness of the algorithm requires, therefore, a small valued or, possibly, a zero valued, parameter $\rho^{(\infty)}$. This lead to the following definition

Definition 2.2 *An A-convergent iteration such that* $\rho^{(\infty)} = 0$ *is called L-*convergent.

In some cases, the previously defined amplification factors may not provide sufficient information. This often happens when the matrix $Z(q)$ in (2.13) is highly nonnormal so that parameters defined trough the eigenvalues of the involved matrices do not give insight into the behaviour of the iteration during the initial phase of the procedure. The so-called *averaged amplification factors*, corresponding to ν inner iterations, are therefore also considered. In detail, by considering a suitable matrix norm $\|\cdot\|$, and by defining,

$$\rho_\nu(q) \equiv \|Z(q)^\nu\|^{\frac{1}{\nu}},$$

the averaged amplification factors are defined as,

$$\rho_\nu^* \equiv \sup_{\arg(q)=\frac{\pi}{2}} \rho_\nu(q), \qquad \tilde{\rho}_\nu \equiv \lim_{q \to 0} \frac{\rho_\nu(q)}{|q|}, \qquad \rho_\nu^{(\infty)} \equiv \lim_{q \to \infty} \rho_\nu(q). \tag{2.20}$$

We conclude the present section with a discussion of some of the recently proposed linear splittings for the solution of the Newton linear systems in (2.3). The main idea used in the derivation of the splitting matrices has been that of defining B^* as a lower triangular matrix L obtained from a suitable factorization of B. Concerning the matrix A no splitting has been yet considered for it (i.e. $A^* \equiv A$), since it has always a very simple (and convenient) structure.

Some of such schemes were proposed by Van der Houwen and De Swart in [68, 69]. In particular, for the *Parallel Triangularly Implicit Runge-Kutta* (PTIRK) method the matrix L was defined as the lower triangular factor in the Crout LU decomposition of B, i.e.

$$B = LU,$$

where

$$L = \begin{pmatrix} \ell_1 & & & \\ \vdots & \ell_2 & & \\ \vdots & & \ddots & \\ \cdot & \cdots & \cdots & \ell_r \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & \cdots & \cdots & \cdot \\ & 1 & & \vdots \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix}. \tag{2.21}$$

The unit diagonal entries on the main diagonal of U determines the equivalence between the A-convergence and the L-convergence properties of the iteration. This can be easily seen by considering that, when $A^* = A = I_r$ and since $B^* = L$, the iteration matrix (2.13) reduces to

$$Z(q) = q(I_r - qL)^{-1}(B - L) = q(I_r - qL)^{-1}L(U - I_r).$$

Consequently,

$$q \to \infty \qquad \Rightarrow \qquad Z(q) \to (I_r - U)\,, \tag{2.22}$$

and, see (2.19),

$$\rho^{(\infty)} \equiv \lim_{q\to\infty} \rho(q) = \rho\,(I_r - U) = 0. \tag{2.23}$$

Moreover, see (2.20), the stiff components are removed from the iteration error within r iterations, i.e. $\rho_r^{(\infty)} = 0$. In addition, for RK methods based on collocation with positive and distinct abscissae, the authors proved the A-convergence of the iteration, see also [66]. The asymptotic amplification factors of the PTIRK method for some RK methods are listed in Table 2.1.

Table 2.1: Asymptotic amplification factors for the PTIRK method

Method	Order	r	ρ^*	$\tilde{\rho}$	$\rho^{(\infty)}$
Gauss	4	2	0.14	0.08	0
Radau IIA	3	2	0.18	0.15	0
	5	3	0.37	0.19	0
	7	4	0.50	0.17	0
Lobatto IIIA	4	2	0.14	0.08	0
	6	3	0.30	0.12	0

Concerning the arithmetic complexity of the iteration, the diagonal entries in L were found to be distinct. As a consequence, see (2.21), the corresponding inner-outer iteration (2.10)-(2.11) requires the factorization of the following $m \times m$ matrices,

$$(I_m - h\ell_i J_0), \qquad i = 1, \ldots, r.$$

However, in [68] the authors do not consider this as a severe limitation for the algorithm since all the above factorizations are each other independent and they were concerned with a parallel implementation on r processors of the algorithm.

A relevant improvement on the described procedure for an efficient implementation on sequential computers was found by Amodio and Brugnano in [4] for methods having the first coefficient matrix A equal to the identity. In fact, the authors proved that, whenever $\det(B) > 0$, as it is the case for an A-stable method with $A = I_r$, a transformation matrix T exists such that

$$\hat{B} \equiv TBT^{-1} = LU,$$

where L and U are defined according to (2.21), with

$$\ell_1 = \ell_2 = \ldots = \ell_r = \det(B)^{\frac{1}{r}}.$$

The authors, therefore, proposed to solve the linear systems in the Newton iteration (2.10) by first performing the variable transformation

$$\hat{\mathbf{y}}^{(i)} = (T \otimes I_m)\mathbf{y}^{(i)}, \tag{2.24}$$

which requires $O(m)$ operations, and then, for the obtained linear system,

$$\left(I_r \otimes I_m - h\hat{B} \otimes J_0\right) \Delta\hat{\mathbf{y}}^{(i)} = -\left(T \otimes I_m\right) F(\mathbf{y}^{(i)}),$$

by using an inner iteration with splitting matrices $A^* = I_r$ and $B^* = L$. The leading term in the arithmetic complexity of such iteration was, therefore, reduced to $\frac{2}{3}m^3$ flops required for the factorization of only one $m \times m$ matrix. The proof of competitiveness, with respect to the PTIRK method, was completed by means of a comparison based on the amplification factors of the two iterations which, for many RK methods, shows comparable convergence properties, [4].

2.3 Nonlinear splittings

The roles of the primary and secondary iteration in (2.10)-(2.11) may be exchanged. As matter of fact, one may think of first performing a nonlinear splitting on (2.2) to obtain a "simple structured" system to be solved by an appropriate method for nonlinear equations. The most famous nonlinear iterative processes are of course the extensions to nonlinear systems of the well-known iterative methods, namely the Jacobi, Gauss-Seidel and SOR methods. Convergence results for these schemes may be found in [86].

In particular, for equation (2.2), a nonlinear block-splitting process is often applied. This is obtained from the following decompositions of the matrices A and B

$$A = A^* - R_A, \qquad B = B^* - R_B, \tag{2.25}$$

where A^* and B^* are nonsingular matrices with a "simple" structure. The nonlinear equation (2.2) is then solved by means of the following iteration

$$\begin{aligned} & A^* \otimes I_m \mathbf{y}^{(i)} - hB^* \otimes I_m \mathbf{f}^{(i)} \\ = \; & (A^* - A) \otimes I_m \mathbf{y}^{(i-1)} - h(B^* - B) \otimes I_m \mathbf{f}^{(i-1)} + \boldsymbol{\eta}, \end{aligned} \tag{2.26}$$

$i = 1, \ldots, \nu$. At each iteration, the equation (2.26) still represents a nonlinear system for $\mathbf{y}^{(i)} \in \mathbb{R}^{rm}$. A Newton type method is always adopted for its solution and the most widely used is, as before, the simplified Newton method. However, since the matrices A^* and B^* are chosen with a simple structure, the arising linear systems are much more cheaply solvable. As an example, the matrices A^* and B^* are chosen lower triangular with constant entries on the main diagonal, so that the simplified Newton iteration for solving (2.26) only requires to factor one $m \times m$ real matrix.

Obviously, for each iteration in (2.26), one may iterate the simplified Newton method until convergence. However, in many cases a single-inner iteration has been found to perform better and the corresponding process has been called *one-step splitting-Newton process*, [71].

We observe that, when the continuous problem is a linear differential equation with constant coefficient matrix, the one-step splitting-Newton process is equivalent to the Newton-splitting one described in the previous section. This is because, the simplified Newton method exactly solves (2.2) and (2.26) in one iteration. It follows that the results obtained with the linear analysis of convergence applied to (2.11) can be directly extended to (2.26). In particular, for the test equation (2.12), the iteration matrix corresponding to (2.26) coincides with the one specified in (2.13).

A nonlinear splitting has been used, for example, in the code GAM implementing methods in the family of BVMs, namely the *Generalized Adams Methods* of orders 3,5,7,9, [20, 71]. The first coefficient matrix of such methods is given by

$$A = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}_{r\times r}.$$

Consequently, the first splitting matrix A^* has been chosen equal to A. Concerning the matrix B^*, the factorization

$$B = LV$$

has been used to define $B^* \equiv L$. Here L is a lower triangular matrix with diagonal entries all equal to

$$\ell = \det(B)^{\frac{1}{r}},$$

and V is a real matrix such that $\det(V) = 1$. More precisely, see (2.22), since as $q \to \infty$, the iteration matrix (2.13) approaches

$$I_r - V = I_r - L^{-1}B,$$

the strictly lower triangular entries in L were found by means of a suitable minimization technique over the quantities

$$\rho(I_r - L^{-1}B), \qquad \|(L^{-1}B)^r\|^{\frac{1}{r}}.$$

The asymptotic amplification factors of the iteration used in GAM have been reported in Table 2.2. As one can seen from the last two rows, the iterations corresponding to the last two higher order method were not A-convergent, though $A(\alpha)$-convergent with $\alpha \approx \pi/2$.

Table 2.2: Asymptotic amplification factors for the iteration used in the code GAM

Order	r	ρ^*	$\rho^*(\pi/2.14)$	$\rho^*(\pi/2.64)$	$\tilde{\rho}$	$\rho^{(\infty)}$
3	4	0.2562	0.2305	0.1806	0.1819	0.0019
5	6	0.5929	0.5326	0.4173	0.2585	0.0212
7	8	1.0048	0.9007	0.7038	0.3064	0.0629
9	9	1.3563	1.2113	0.9390	0.3014	0.0753

2.4 Remarks

The approach described in Section 2.3 may be very competitive provided that suitable matrices A^* and B^* can be obtained. Nevertheless, when satisfactory convergence properties are required, their derivation may be very difficult. In particular, many difficulties have been encountered when a zero-valued stiff amplification factor is required in order to make the iteration well-suited for the solution of discrete problems corresponding to the solution of stiff differential equations.

In the next chapter, it will be shown how the use of *blended schemes* allows to "naturally" derive iterative procedures with "low" arithmetic complexity per step and very "good" convergence properties.

Chapter 3

Blended Implicit Methods

Blended Implicit Methods are methods which, in addition to classical requirement, such as high order of accuracy and "good" stability properties, do have favourable properties from the implementation point of view. They are obtained by means of a suitable combination of two component methods, so that efficient nonlinear splittings are naturally defined for the solution of the obtained discrete problem.

In the past years, many attempts have been made to derive numerical methods for ODEs as the combination of two methods. A well-known example is the popular θ-method. Additional examples are provided by the *Blended Linear Multistep Formulas* of Skeel and Kong [100] and by the *Blended Block BVMs* of Brugnano [14]. However, slight different aims were pursued in doing this:

- in the case of the θ-method and of the blended linear multistep formulas, the only aim was that of getting a method with better stability properties than the two component ones;
- in the case of blended block BVMs, the above aim was coupled with that of getting an efficient implementation of the resulting method.

Blended Implicit Methods, instead, are obtained by means of a suitable combination of discrete problems derived from the same basic method so that, with the latter, they share the same accuracy and stability properties. For this reason, we shall also speak about the *blended implementation* of the basic method.

3.1 Blended Implementation of Block Implicit Methods

In order to unnecessarily complicate the notation and to carry out the linear analysis of convergence, we shall consider the application of the methods to

the classical test equation

$$y' = \mu y, \qquad y(t_0) = y_0, \qquad \mathrm{Re}(\mu) < 0, \tag{3.1}$$

for which, by setting as usual $q = h\mu$, the discrete problem (2.2), at step n, assumes the form:

$$(A - qB)\mathbf{y}_n = \boldsymbol{\eta}_n. \tag{3.2}$$

We observe that the solution of the previous equation is not affected by left-multiplication by A^{-1} or B^{-1} of both sides of the equation,

$$(I_r - qA^{-1}B)\mathbf{y}_n = A^{-1}\boldsymbol{\eta}_n, \qquad (B^{-1}A - qI_r)\mathbf{y}_n = B^{-1}\boldsymbol{\eta}_n. \tag{3.3}$$

The basic idea for the blended implementation of the method (3.2) relies on the fact that, by combining equations in the form (3.3), the discrete solution does not change. In more detail, let A_1 be a nonsingular matrix with a "simple" structure. By multiplying on the left both sides of the first equation in (3.3), we then obtain

$$(A_1 - qB_1)\mathbf{y}_n = \boldsymbol{\eta}_{1n}, \tag{3.4}$$

where

$$B_1 = A_1A^{-1}B, \qquad \boldsymbol{\eta}_{1n} = A_1A^{-1}\boldsymbol{\eta}_n. \tag{3.5}$$

Similarly, by considering another nonsingular and "simple structured" matrix B_2, by multiplying on the left the second equation in (3.3) we obtain

$$(A_2 - qB_2)\mathbf{y} = \boldsymbol{\eta}_{2n}, \tag{3.6}$$

where

$$A_2 = B_2B^{-1}A, \qquad \boldsymbol{\eta}_{2n} = B_2B^{-1}\boldsymbol{\eta}_n. \tag{3.7}$$

Obviously, both equations (3.4) and (3.6) do have the same solution as equation (3.2), since they are derived from the same method. In addition to this, let us define a suitable *weighting function* $\theta(q)$ such that

$$\theta(0) = I, \qquad \theta(q) \to O, \text{ as } q \to \infty, \tag{3.8}$$

being, hereafter, I and O the $r \times r$ identity and the zero matrix respectively. Then, also the following equation,

$$M(q)\mathbf{y}_n - \boldsymbol{\eta}_n(q) \equiv (A(q) - qB(q))\,\mathbf{y}_n - \boldsymbol{\eta}_n(q) \;=\; \mathbf{0}, \tag{3.9}$$

where

$$\begin{aligned} A(q) &\equiv \theta(q)A_1 + (I-\theta(q))A_2, \\ B(q) &\equiv \theta(q)B_1 + (I-\theta(q))B_2, \\ \boldsymbol{\eta}_n(q) &\equiv \theta(q)\boldsymbol{\eta}_{1n} + (I-\theta(q))\boldsymbol{\eta}_{2n}, \end{aligned} \tag{3.10}$$

does have the same solution as (3.2). Equation (3.9) defines a *blended implicit method* associated with the block method (3.2), due to the fact that the discrete problem is obtained as the "blending" of two equivalent forms of the same block method. We observe that, since the numerical solution has not been affected, the blended implicit method (3.9) does have the same accuracy and stability properties of the method in (3.2) .

The key point concerning a blended implicit method is that its structure naturally induces the choice of a nonlinear splitting for iteratively solving (3.9). As matter of fact, from (3.8), one easily verifies that the matrix $M(q)$ in (3.9) is such that:

- $M(q) = A_1 + O(q) \approx A_1, \qquad$ when $\quad q \approx 0;$
- $M(q) = -q\left(B_2 + O(q^{-1})\right) \approx -qB_2, \quad$ as $\qquad q \to \infty.$

Consequently, instead of solving (3.9), one may think to solve iteratively

$$N(q)\mathbf{y}_n^{(i+1)} = (N(q) - M(q))\mathbf{y}_n^{(i)} + \boldsymbol{\eta}_n(q), \qquad i = 0, 1, \ldots, \tag{3.11}$$

where

$$N(q) \equiv A_1 - qB_2. \tag{3.12}$$

This is because

$$N(0) = M(0), \qquad \text{and} \qquad N(q) \approx M(q), \;\text{ when } |q| \gg 1. \tag{3.13}$$

We shall call (3.11) the *blended iteration* associated with the blended method (3.9). The corresponding iteration matrix is then given by

$$Z(q) \equiv N(q)^{-1}\,(N(q) - M(q)) = I - N(q)^{-1}M(q), \tag{3.14}$$

and the iteration will converge if and only if the spectral radius $\rho(q)$ of $Z(q)$ is smaller than 1. We observe that, from (3.13), one immediately obtains,

$$Z(0) = O, \qquad Z(q) \to O, \qquad \text{as } q \to \infty. \tag{3.15}$$

Consequently, see (2.19),

$$\rho(0) = 0, \qquad \rho^{(\infty)} \equiv \lim_{q\to\infty} \rho(q) = 0. \tag{3.16}$$

Moreover, see (2.20), the second property in (3.15) implies that

$$\rho_\nu^{(\infty)} = 0, \qquad \text{for all } \nu \geq 1,$$

so that the iteration (3.9) is particularly well-suited for stiff problems.

It must be stressed that the properties (3.16) of the blended iteration do not depend on the particular choice of the matrices A_1 and B_2 used in (3.4) and (3.6). They are only due to the blended implementation of the method, namely to the particular structure of the matrices $M(q)$ and $N(q)$ in (3.9) and (3.12). As a consequence, additional convergence properties of the iteration may be improved by means of an appropriate choice of the two matrices A_1 and B_2. As an example, a possible criterion to be adopted for their definition is the minimization of the maximum amplification factor ρ^* of the iteration in order to possibly obtain an A-convergent (and, then, L-convergent) iteration. Concerning this point, in the sequel, we shall always assume the weighting function $\theta(q)$ to be analytical in $\mathbb{C}^-$ and the spectrum of the matrix pencil (3.12) to be contained in $\mathbb{C}^+$, so that the maximum amplification factor of the blended iteration is given by

$$\rho^* = \max_{x\geq 0} \rho(\mathrm{i}x), \tag{3.17}$$

where, as usual, i denotes the imaginary unit. Moreover, we will assume the iteration matrix to be well-defined in a neighbourhood of the origin. Consequently, from the first equation in (3.16), it follows that

$$\rho(q) \approx \tilde{\rho}\,|q|, \qquad \text{when } q \approx 0, \tag{3.18}$$

where $\tilde{\rho}$ is the *nonstiff amplification factor* defined in (2.18).

Concerning the two "simple structured" matrices A_1 and B_2, we have considered the following choice, though different ones are possible,

$$A_1 = I + L_A, \qquad B_2 = \gamma I + L_B, \tag{3.19}$$

where L_A and L_B are strictly lower triangular matrices, and γ is a positive parameter. With such assumptions, we have that the linear systems required by the iteration (3.11) are lower triangular (block lower triangular when the method is applied to a system of equations). Moreover, in the case of systems, one only needs to factor one matrix having the same size of the continuous problem.

Finally, in order to keep low the computational cost, the weight function $\theta(q)$ is defined as

$$\theta(q) \equiv \mathrm{diag}(N(q))^{-1} = (I - q\gamma I)^{-1}, \tag{3.20}$$

so that the properties (3.8) are satisfied, the iteration (3.11) is well-defined for all $q \in \mathbb{C}^-$, and, in the case of systems, no additional factorizations are required, besides the one needed for $N(q)$.

With such assumptions, the only key-point which we need to clarify are the following ones:

1. the choice of an appropriate basic method (3.2),
2. the choice of corresponding "simple structured" matrices A_1 and B_2 in (3.19) (the remaining matrices B_1 and A_2 being defined by (3.5) and (3.7), respectively).

The first point will be discussed in the next section, whereas the second one will be addressed in Section 3.3.

3.2 Choice of the component methods

Let now introduce the methods that we shall implement in blended form, according to what has been said in the previous section. Even though different choices are possible, we have considered methods which have been already introduced in the past years by Watts and Shampine [104]. Such methods are characterized by the fact that each one of the r equations which define the method itself corresponds to a linear multistep formula with the same order of accuracy. The numerical solution is therefore advanced by a block of r equally accurate new values at a time approximating the solution on a set of r uniformly distributed mesh-points. In more details, if we assume, for

simplicity, the following uniform partition for the entire integration interval $[t_0, T]$:

$$t_k = t_0 + k \cdot h, \qquad k = 0, \ldots, N \equiv lr, \qquad h = \frac{T - t_0}{N}, \tag{3.21}$$

then for each n multiple of r the block method provides the following r approximations to $y(t)$,

$$y_{n+i} \approx y(t_{n+i}), \qquad i = 1, \ldots, r,$$

starting from the approximation y_n to $y(t_n)$. Consequently, for theoretical purposes, the block procedure may be consider to be a one-step method.

As a consequence, such schemes posses features of both RK methods and LMF. In particular, with RK methods, they share good stability properties for high order methods and a stepsize variation strategy typical of one-step schemes (which is simpler than those for LMF). With LMF, instead, they share the same simple representation of the local truncation error, which allows to define efficient strategies for a variable-order implementation of the methods.

We now discuss how block methods with "good" classical requirements can be obtained. Even though the methods could be also derived in the framework of Runge-Kutta methods (by means of the "V-transform" [22, 23, 59]) we prefer to use the same framework originally used in [104] (see also [21]). Let, therefore, define the following $r \times (r+1)$ matrices,

$$\begin{aligned} \hat{A} = [\mathbf{a} \,|\, A] \quad &\equiv \quad \left(\begin{array}{c|ccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \ldots & \alpha_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(r)} & \alpha_1^{(r)} & \ldots & \alpha_r^{(r)} \end{array} \right), \\ \hat{B} = [\mathbf{b} \,|\, B] \quad &\equiv \quad \left(\begin{array}{c|ccc} \beta_0^{(1)} & \beta_1^{(1)} & \ldots & \beta_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(r)} & \beta_1^{(r)} & \ldots & \beta_r^{(r)} \end{array} \right), \end{aligned} \tag{3.22}$$

where the coefficients on the ith row of the two matrices define a suitable r-step LMF. In the following, both the two matrices A and B will be always assumed to be nonsingular. Then, for each $n = 0,\ r,\ 2r,\ \ldots$, the new block of values is obtained as the solution of the following discrete problem:

$$F(\mathbf{y}_n) \equiv A \otimes I_m \mathbf{y}_n - hB \otimes I_m \mathbf{f}_n + (\mathbf{a} \otimes y_n - h\mathbf{b} \otimes f_n) = \mathbf{0}, \tag{3.23}$$

where

$$\mathbf{y}_n = \begin{pmatrix} y_{n+1} \\ \vdots \\ y_{n+r} \end{pmatrix}, \qquad \mathbf{f}_n = \begin{pmatrix} f_{n+1} \\ \vdots \\ f_{n+r} \end{pmatrix}, \qquad f_j = f(t_j, y_j).$$

Here the vector $\boldsymbol{\eta}_n$ in (2.2) is then given by

$$\boldsymbol{\eta}_n = -(\mathbf{a} \otimes y_n - h\mathbf{b} \otimes f_n). \tag{3.24}$$

The following is a first important result concerning the accuracy of such methods, [21].

Theorem 3.1 *Let the matrices (3.22) satisfy the following set of equations,*

$$\hat{A}\hat{\mathbf{q}}_i = i\hat{B}\hat{\mathbf{q}}_{i-1}, \qquad i = 0, \ldots, p, \tag{3.25}$$

where

$$\hat{\mathbf{q}}_{-1} \equiv \mathbf{0}, \qquad \hat{\mathbf{q}}_i = \begin{pmatrix} 0^i \\ 1^i \\ \vdots \\ r^i \end{pmatrix} \equiv \begin{pmatrix} 0^i \\ \mathbf{q}_i \end{pmatrix}, \quad i = 0, 1, \ldots. \tag{3.26}$$

Then the LMF defining the block method (3.22) have a truncation error which is at least $O(h^{p+1})$.

Proof The equations (3.25) are nothing but the usual order p conditions for LMF, see (1.11), simultaneously imposed for all the r LMF corresponding to (3.22). □

By considering in (3.25) the equations corresponding to $i = 0, 1$, the above result implies that when all methods in (3.22) are consistent ($p \geq 1$) the first two columns of the augmented matrices $\hat{A}$ and $\hat{B}$ are related to the corresponding square matrices A and B by means of the following relations

$$\mathbf{a} = -A\mathbf{1}, \qquad \mathbf{b} = A\mathbf{q}_1 - B\mathbf{1}, \tag{3.27}$$

where $\mathbf{1} \equiv \mathbf{q}_0$ denotes the vector with all unit entries (see (3.26)). As a consequence, attention can be driven to the square matrices A and B alone provided that, as we obviously assume, consistent LMF are used. In particular, it is an easy matter to verify the following result.

Corollary 3.1 *Let the matrices defined in (3.22) satisfy (3.27) and the following set of equations*

$$A\mathbf{q}_i = iB\mathbf{q}_{i-1}, \qquad i = 2, \ldots, p. \tag{3.28}$$

Then the LMF defining the block method (3.22) have a truncation error which is at least $O(h^{p+1})$. □

Let now define, for each $j = 1, 2, \ldots,$ the following matrices,

$$D_j = \mathrm{diag}\,(1,\, 2,\, \ldots,\, j)\,, \qquad Q_j = (\,\mathbf{q}_1\, \ldots\, \mathbf{q}_j\,)\,. \tag{3.29}$$

Then, the set of equations in (3.28) may be collected into the following one:

$$AD_rQ_{p-1} = BQ_{p-1}\,(I_{p-1} + D_{p-1})\,. \tag{3.30}$$

Obviously, in the above equation it must be $p \leq r + 1$. In addition, when $p = r + 1$, the following result holds true (see also [14]).

Theorem 3.2 *If* $p = r + 1$ *then the matrix* $A^{-1}B$ *is uniquely determined.*

Proof In fact, when $p = r + 1$, the matrix Q_r in (3.30) is, essentially, a nonsingular Vandermonde matrix. Consequently, one obtains that

$$A^{-1}B = D_rQ_r\,(I_r + D_r)^{-1}\,Q_r^{-1}. \tag{3.31}$$

whose right-hand side only depends on r. □

As already observed, the nonsingularity of the matrices A and B implies that methods sharing the same matrix

$$C = A^{-1}B, \tag{3.32}$$

provides the same numerical solution and, as a consequence, have the same accuracy and stability properties. In this sense, in [14] such methods have been called *equivalent* methods. Then, from Theorem 3.2, it follows that all block methods defined by a set of LMF with the highest order $p = r + 1$ are equivalent.

Let us now look at the stability properties of such equivalent methods. As already observed in the introduction of the section, block methods are considered as one-step methods for theoretical purposes. Consequently, as for RK methods, the stability properties are studied by considering the stability function of the method. In particular, see (3.23), since the discrete problem corresponding to the test equation (3.1) is given by

$$(A - qB)\mathbf{y}_n = (q\mathbf{b} - \mathbf{a})y_n, \tag{3.33}$$

and the starting point for the subsequent application of the method is the last entry in $\mathbf{y}_n$, the stability function of the method is given by

$$g(q) \equiv \mathbf{e}_r^T (A - qB)^{-1} (q\mathbf{b} - \mathbf{a}) = \frac{\det(W(q))}{\det(I_r - qC)}, \tag{3.34}$$

where $\mathbf{e}_r$ is the last unit vector in $\mathbb{R}^r$ while $W(q)$ is obtained from the matrix $I_r - qC$, whose last column has been substituted by $\mathbf{1} + q(\mathbf{q}_1 - C\mathbf{1})$. The second equality in (3.34) follows from the nonsingularity of A and B, the consistency conditions in (3.27), and the Cramer's rule. The method is therefore A-stable provided that

$$\mathrm{Re}(q) < 0 \qquad \Rightarrow \qquad |g(q)| < 1.$$

A necessary requirement for the above property to hold is the function $g(q)$ to be well-defined in the left-half complex plane. This leads to the following definition.

Definition 3.1 *A block method is said to be* pre-stable *if the spectrum of the corresponding matrix pencil is contained in* $\mathbb{C}^+$.

This fact implies that the result of Theorem 3.2 is useful only to define pre-stable methods up to $r = 8$; as matter of fact, by direct inspection one verifies that the matrix on the right-hand side of (3.31) has eigenvalues with negative real part when $r \geq 9$. Consequently, the corresponding methods cannot be pre-stable: in fact, the spectrum of the pencil $(A - qB)$ coincides with that of C^{-1} (see (3.32)), since both the two matrices A and B are assumed to be nonsingular.

In order to obtain alternative criteria for choosing C, we shall relax the order conditions for the LMF on each row of the block method. In particular, it will be convenient to impose only the order r conditions: i.e. (see (3.30) and (3.32))

$$D_r Q_{r-1} = C Q_{r-1} (I_{r-1} + D_{r-1}). \tag{3.35}$$

It remains one more condition to be imposed and it will be used to fix the spectrum of the matrix C. Concerning this point, the following result applies.

Theorem 3.3 *The matrix C defined as*

$$C = Q_r G^{-1} F G Q_r^{-1}, \tag{3.36}$$

where Q_r is defined according to (3.29), and

$$G = \begin{pmatrix} 1! & & \\ & \ddots & \\ & & r! \end{pmatrix}, \qquad F = \left(\begin{array}{ccc|c} 1 & & & -d_0 \\ & \ddots & & -d_1 \\ & & & \vdots \\ & & 1 & -d_{r-1} \end{array}\right), \tag{3.37}$$

is the unique matrix such that:

(i) the characteristic polynomial is given by

$$d(z) = \sum_{i=0}^{r} d_i z^i, \qquad d_r = 1; \tag{3.38}$$

(ii) each row of the consistent block method with matrices

$$A = I_r, \qquad B = C, \tag{3.39}$$

corresponds to a LMF with an $O(h^{r+1})$ truncation error.

Proof We will prove that if the matrix C satisfies the properties (i) and (ii), then it must be necessarily equal to the matrix on the right-hand side of equation (3.36). As matter of fact, because of the second requirement, we have already seen that C must satisfy equation (3.35), which is equivalent to (see (3.26) and (3.29)),

$$Q_r \begin{pmatrix} \mathbf{0}^T \\ (I_{r-1} + D_{r-1})^{-1} \end{pmatrix} = C Q_r \begin{pmatrix} I_{r-1} \\ \mathbf{0}^T \end{pmatrix}. \tag{3.40}$$

Let now denote with $\hat{\mathbf{d}}$ the unique vector such that

$$Q_r \hat{\mathbf{d}} = C \mathbf{q}_r. \tag{3.41}$$

Then, we can collect the two previous equations into the following one:

$$Q_r \left(\begin{pmatrix} \mathbf{0}^T \\ (I_{r-1} + D_{r-1})^{-1} \end{pmatrix} \middle| \, \hat{\mathbf{d}} \right) = C Q_r. \tag{3.42}$$

Moreover, see (3.37), we observe that

$$\left(\left(\begin{array}{c} \mathbf{0}^T \\ (I_{r-1} + D_{r-1})^{-1} \end{array} \right) \middle| \, \hat{\mathbf{d}} \right) = G^{-1} \hat{F} G, \tag{3.43}$$

where

$$\hat{F} = \left(\left(\begin{array}{c} \mathbf{0}^T \\ I_{r-1} \end{array} \right) \middle| \, -\mathbf{d} \right), \qquad \mathbf{d} \equiv -\frac{1}{r!} G \hat{\mathbf{d}}. \tag{3.44}$$

The matrix C is therefore similar to the Frobenius-type matrix $\hat{F}$. It follows that the characteristic polynomial of C is given by the polynomial in (3.38) provided that, see (3.37),

$$\mathbf{d} \equiv (\, d_0 \, \ldots \, d_{r-1} \,)^T,$$

or, equivalently, $\hat{F} = F$ so that C must be equal to the matrix on the right-hand side in (3.36). By using similar arguments, it is easily proved that the latter matrix always satisfies the properties in (i) and (ii) so that the proof is complete. □

From the previous theorem it follows that once the desired characteristic polynomial $d(z)$ (or, equivalently, the desired spectrum) for the matrix C has been chosen, one can simply use the formula in (3.36) to derive the block method with the prescribed properties. Let us now discuss how to properly choose the polynomial $d(z)$ in order to obtain a method with "good" stability properties.

We surely will choose it in order to have all the roots contained in $\mathbb{C}^+$, so that the method is pre-stable. This is not enough, however, to define a "good" method. In fact, from (3.33) and (3.34) for $n = 0$, one obtains

$$y_r(q) = \frac{\det(W(q))}{\det(I_r - qC)} y_0 \approx \mathrm{e}^{rq} y_0,$$

so that (see (3.34) and (3.38))

$$\mathrm{e}^{rq} \approx g(q) = \frac{\det(W(q))}{\det(I_r - qC)} = \frac{\varphi(q)}{q^r d(q^{-1})} \equiv \frac{\varphi(q)}{\mu(q)}, \tag{3.45}$$

where $\varphi(q) = \det(-W(q))$ is a polynomial of maximum degree r and

$$\mu(q) = \sum_{i=0}^{r} d_i q^{r-i}, \qquad d_r = 1,$$

is a polynomial of exact degree r since we assume $(I_r - qC)$ to be nonsingular in the left-half complex plane.

Remark 3.1 *Observe that, because of (3.35), the approximation in (3.45) must be at least $O(q^{r+1})$ accurate.*

The characteristic polynomial $d(q)$ of the matrix C coincides, therefore, with the reciprocal and scaled polynomial at the denominator of a rational approximation to the exponential. One of the most classical ones is the Padé (ν, r),

$$\mathrm{e}^z \approx \frac{\varphi_{\nu,r}(z)}{\mu_{\nu,r}(z)},$$

where $\varphi_{\nu,r}(z)$ and $\mu_{\nu,r}(z)$ are the unique polynomials of degree ν and r, respectively, such that

$$\varphi_{\nu,r}(z) = \mu_{\nu,r}(z)\mathrm{e}^z + O(z^{\nu+r+1}). \tag{3.46}$$

The expression of the two polynomials is well-known and is given by

$$\begin{aligned} \varphi_{\nu,r}(z) &= \sum_{i=0}^{\nu} \frac{(\nu+r-i)!\,\nu!}{(\nu+r)!\,i!\,(\nu-i)!}\, z^i, \\ \mu_{\nu,r}(z) &= \sum_{i=0}^{r} (-1)^i \frac{(\nu+r-i)!\,r!}{(\nu+r)!\,i!\,(r-i)!}\, z^i. \end{aligned} \tag{3.47}$$

Moreover, the following properties hold true for such polynomials (see [92] and the references therein).

Theorem 3.4 *For all $\nu, r \geq 0$:*

1. *$\mu_{\nu,r}(z) \equiv \varphi_{r,\nu}(-z)$;*
2. *if $r \geq 1$, all the zeros of the polynomial $\mu_{\nu,r}$ lie in the annulus*

$$(r+\nu)\xi < |z| < r + \nu + 4/3,$$

where $\xi \approx 0.278465$ is the unique positive root of $x\mathrm{e}^{x+1} = 1$. □

By considering (3.45) and (3.46), the following choice for the characteristic polynomial $d(q)$ of the matrix C seems, therefore, appropriate

$$q^r d(q^{-1}) = \mu_{\nu,r}(rq). \tag{3.48}$$

As matter of fact, we observe, first of all, that from Remark 3.1 and (3.46) it follows that, if $d(q)$ is defined as in (3.48), the polynomial $\varphi(q)$ in (3.45) is necessarily given by

$$\varphi(q) = \varphi_{\nu,r}(rq).$$

As a consequence, the stability function $g(q)$ of the method obtained through the choice (3.48) coincides with the (ν, r) Padé approximation to the exponential evaluated at rq. From the *Ehle conjecture* [50], subsequently proved in [57] by Hairer, Wanner and Nørsett, it is known that methods with such stability function are A-stable, for each $r \geq 3$, iff $\nu \in \{r-2, r-1, r\}$. Moreover, such methods are also L-stable only if $\nu < r$. In the present case, we look for L-stable methods and, consequently, we need to choose appropriate values for the couples (ν, r), $\nu \in \{r-2, r-1\}$. In order to make the proper choice, we observe that, for the test equation, we have

$$y_r = \frac{\varphi_{\nu,r}(rq)}{\mu_{\nu,r}(rq)}\, y_0 \approx e^{rq} y_0.$$

We know that such an approximation is exact at $q = 0$ and as $q \to \infty$ (due to the L-stability of the methods). In addition to this, we also require, for $\mu < 0$ (see (3.1)), the discrete solution to have the same sign as the continuous one (which is the sign of y_0), whatever the stepsize h used. This restrict the range of choices for the couple (ν, r) to the following ones:

- $(r-2, r)$ when r is even,
- $(r-1, r)$ when r is odd,

since it is known that only for such values, when $\nu \in \{r-2, r-1\}$, the Padé approximation is analytic in $\mathbb{C}^-$ with no real and negative zeros.

Let now discuss the order of accuracy of the corresponding block methods. For this purpose, let us denote by

$$\hat{\mathbf{y}} \equiv \begin{pmatrix} y(t_1) \\ \vdots \\ y(t_r) \end{pmatrix}, \qquad \hat{\mathbf{f}} \equiv \begin{pmatrix} f(t_1, y(t_1)) \\ \vdots \\ f(t_r, y(t_r)) \end{pmatrix},$$

where $y(t)$ is the solution of the IVP (2.1). From (3.23) one then obtains

$$A \otimes I_m \hat{\mathbf{y}} - hB \otimes I_m \hat{\mathbf{f}} + \mathbf{a} \otimes y_0 - h\mathbf{b} \otimes f_0 = \boldsymbol{\tau}, \tag{3.49}$$

where $\boldsymbol{\tau}$ is the vector with the truncation errors of the method. By assuming that $y(t)$ is sufficiently smooth, the entries of the latter vector are given by

$$\begin{aligned}\tau_i &= \sum_{j>r} \frac{y^{(j)}(t_0)}{j!} h^j \left(\sum_{k=0}^{r} k^{j-1}(k\alpha_k^{(i)} - j\beta_k^{(i)}) \right) \\ &\equiv \sum_{j>r} y^{(j)}(t_0) h^j v_{ji}, \qquad i = 1, \ldots, r, \end{aligned} \tag{3.50}$$

because of the order r conditions (3.35). Consequently, by subtracting (3.23) from (3.49), we obtain

$$A \otimes I_m(\hat{\mathbf{y}} - \mathbf{y}) - hB \otimes I_m(\hat{\mathbf{f}} - \mathbf{f}) = \boldsymbol{\tau}.$$

By introducing the vector $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$ of the local error, one then concludes that the latter satisfies the equation

$$\left(A \otimes I_m - hB \otimes I_m \, \hat{\mathbf{J}} \right) \mathbf{e} = \boldsymbol{\tau}, \tag{3.51}$$

where

$$\begin{aligned}\hat{\mathbf{J}} &= \begin{pmatrix} \hat{J}_1 & & \\ & \ddots & \\ & & \hat{J}_r \end{pmatrix}, \\ \hat{J}_i &= \int_0^1 J(t_i, sy(t_i) + (1-s)y_i) ds \equiv J_0 + O(h), \end{aligned} \tag{3.52}$$

$J(t,y) = \frac{\partial}{\partial y} f(t,y)$ and $J_0 = J(t_0, y_0)$. Like any one-step method, the order of accuracy is defined as follows.

Definition 3.2 *The block method corresponding to (3.51) has order p provided that $e_r = O(h^{p+1})$, where e_r is the last block entry of the vector* $\mathbf{e}$.

Obviously, from (3.50) and (3.51), we have that the order of the method is $p \geq r$. In general, the relations between the order conditions (3.50) and the global order of the method may be very entangled, as the Butcher theory for Runge-Kutta methods shows. Nevertheless, in case we look for values of p only slightly greater than r, the following result may be useful.

Theorem 3.5 *Consider the following possible cases for the method corresponding to (3.50)-(3.51)*

(0) $\mathbf{e}_r^T A^{-1} \mathbf{v}_{r+1} \neq 0$;

(1) $\mathbf{e}_r^T A^{-1}\mathbf{v}_{r+1} = 0$ and $(\mathbf{e}_r^T A^{-1}\mathbf{v}_{r+2} \neq 0$ or $\mathbf{e}_r^T CA^{-1}\mathbf{v}_{r+1} \neq 0)$;

(2) $\mathbf{e}_r^T A^{-1}\mathbf{v}_{r+1} = \mathbf{e}_r^T A^{-1}\mathbf{v}_{r+2} = \mathbf{e}_r^T CA^{-1}\mathbf{v}_{r+1} = 0$,

where, see (3.50),

$$\mathbf{v}_j = \begin{pmatrix} v_{j1} \\ \vdots \\ v_{jr} \end{pmatrix}, \qquad j > r. \tag{3.53}$$

Then the global order of the method is exactly $p = r + i$ in cases $i = 0, 1$; and $p \geq r + 2$ in case 2.

Proof In fact, from (3.32), and (3.50)-(3.52) one obtains

$$\begin{aligned}
\mathbf{e} &= \left(I_r \otimes I_m - hC \otimes I_m\, \hat{\mathbf{J}}\right)^{-1} \left(A^{-1} \otimes I_m\right) \boldsymbol{\tau} \\
&= h^{r+1}(A^{-1}\mathbf{v}_{r+1}) \otimes I_m\, y^{(r+1)}(t_0) + \\
&\quad h^{r+2}\Big((A^{-1}\mathbf{v}_{r+2}) \otimes I_m\, y^{(r+2)}(t_0) + \\
&\qquad C \otimes I_m\, \hat{\mathbf{J}}\, (A^{-1}\mathbf{v}_{r+1}) \otimes I_m\, y^{(r+1)}(t_0)\Big) + O(h^{r+3}) \\
&= h^{r+1}(A^{-1}\mathbf{v}_{r+1}) \otimes I_m\, y^{(r+1)}(t_0) + \\
&\quad h^{r+2}\Big((A^{-1}\mathbf{v}_{r+2}) \otimes I_m\, y^{(r+2)}(t_0) + (CA^{-1}\mathbf{v}_{r+1}) \otimes J_0\, y^{(r+1)}(t_0)\Big) + \\
&\quad O(h^{r+3}),
\end{aligned} \tag{3.54}$$

from which, in view of Definition 3.2, the thesis easily follows. □

By direct inspection, one verifies that the methods obtained with the choice in (3.48) satisfy the hypothesis (1), in the previous Theorem, when r is odd, and the hypothesis (2), when r is even. In addition, in the latter case, some computations allows to prove that the last block entry in the local error is exactly $O(h^{r+3})$ accurate. The order of accuracy of the methods here considered is, therefore, given by

$$\begin{aligned}
&p = r + 1, \qquad \text{when } r \text{ is odd,} \\
&\qquad\qquad\qquad\qquad\qquad\qquad r \geq 3, \quad \nu = r-2, r-1, r. \\
&p = r + 2, \qquad \text{when } r \text{ is even,}
\end{aligned} \tag{3.55}$$

All the previous considerations, lead us to choose as basic methods for the blended implementation the ones listed in Table 3.1. We remark that the blocksize r of each method has been always chosen equal to the order

Table 3.1: Basic block methods

Padé	(2,3)	(2,4)	(4,6)	(6,8)	(8,10)	(10,12)
r	3	4	6	8	10	12
Order	4	6	8	10	12	14

of the previous method in that list. This features, in fact, will be used to derive an efficient variable-order implementation of the methods themselves (see the next chapter). In Figure 3.1 the boundaries of the absolute stability regions of such methods have been plotted.

3.3 Choice of the splitting matrices

In this section we shall study in more detail particular choices of appropriate matrices A_1 and B_2, as defined in (3.19). As explained in Section 3.1, this uniquely defines the whole blended implementation of the underlying block method. The remaining matrices B_1 and A_2 are, in fact, defined according to (3.5) and (3.7), respectively, and the weight function θ is defined according to (3.12) and (3.20).

To begin with, we derive from (3.9), (3.12) and (3.14) the following expression for the amplification matrix of the blended iteration:

$$\begin{aligned} Z(q) &= N(q)^{-1}\,(N(q) - M(q)) \\ &= N(q)^{-1}\Big(\,(I - \theta(q))\,(A_1 - A_2) + q\;\theta(q)(B_1 - B_2)\Big). \end{aligned} \tag{3.56}$$

Let now consider the simpler case where (see (3.19)),

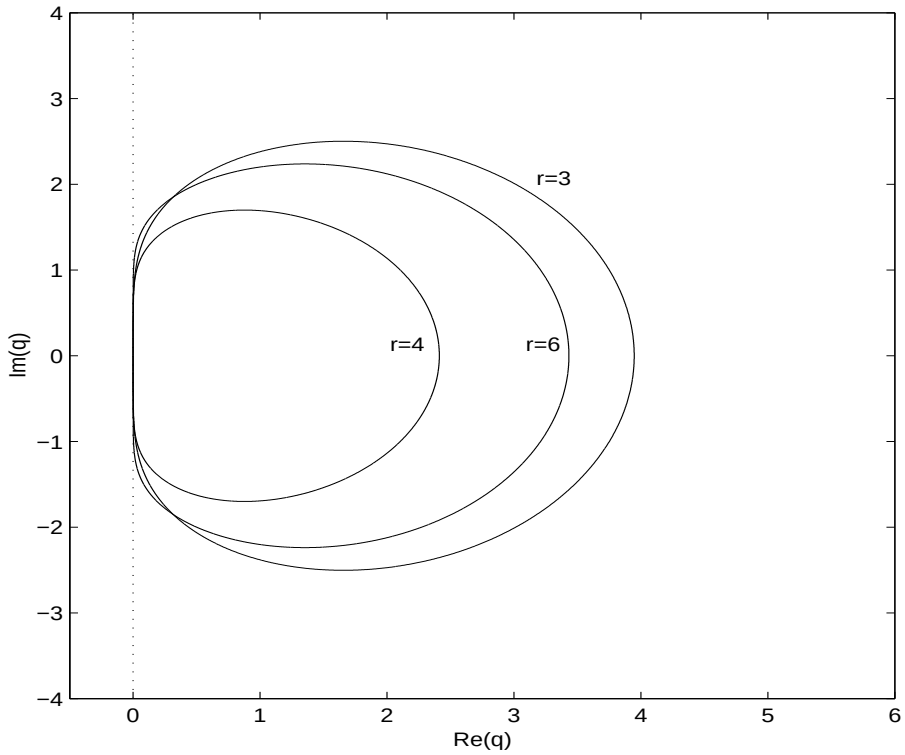
$$L_A = L_B = O, \tag{3.57}$$

since in such a case a complete spectral analysis can be carried out. In fact, in such a case, one obtains that

$$A_1 = I \quad \Rightarrow \quad B_1 = C, \qquad\qquad B_2 = \gamma I \quad \Rightarrow \quad A_2 = \gamma C^{-1}. \tag{3.58}$$

This, in turn, allows us to easily derive the following result.

Theorem 3.6 *Assume that for the blended method (3.9) the previous equalities (3.57) hold true. Then, the eigenvalues of the amplification matrix (3.56)-(3.58) are given by*



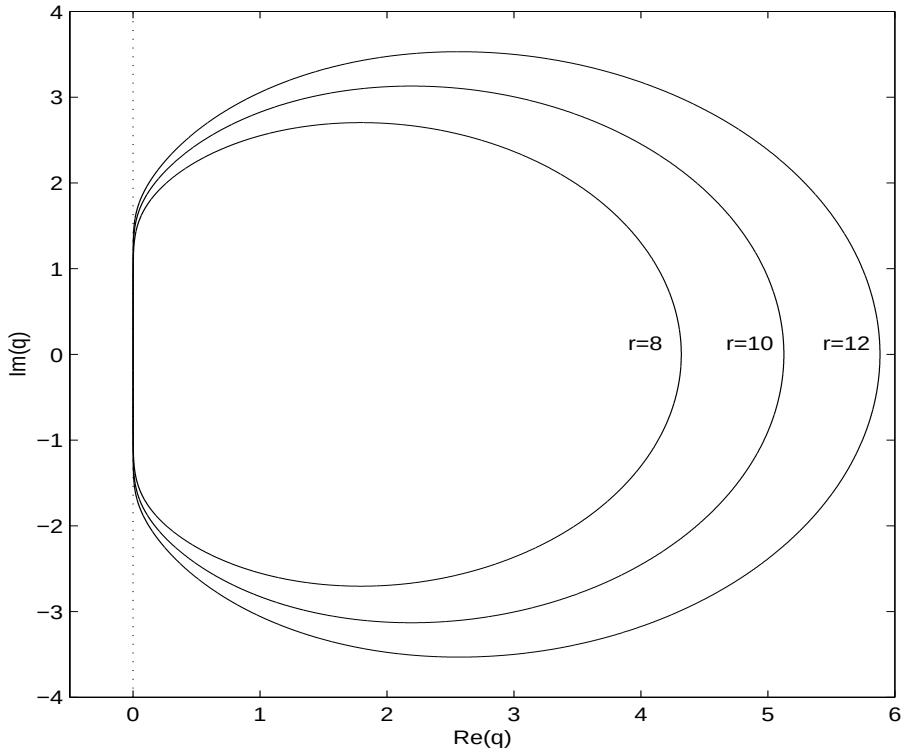


Figure 3.1: Boundaries of the Absolute stability regions of the block methods in Table 3.1

$$\frac{q(\lambda-\gamma)^2}{\lambda(1-\gamma q)^2}, \qquad \lambda \in \sigma(C). \tag{3.59}$$

Proof Since the equalities (3.57) are satisfied, then also (3.58) do. Consequently, by taking into account (3.12) and (3.20), from (3.56) one obtains

$$\begin{aligned} Z(q) &= (1-\gamma q)^{-1}\Big(\gamma q(1-\gamma q)^{-1}(\gamma C^{-1}-I)+q(1-\gamma q)^{-1}(C-\gamma I)\Big) \\ &= q(1-\gamma q)^{-2}\left(C-2\gamma I+\gamma^2 C^{-1}\right) \\ &= q(1-\gamma q)^{-2}C^{-1}\left(C-\gamma I\right)^2. \end{aligned}$$

from which the thesis follows. □

When (3.57) holds true, the above result allows the following easy characterization of the spectral radius $\rho(q)$ of the amplification matrix $Z(q)$:

$$\rho(q) = \max_{\lambda\in\sigma(C)}\left|\frac{q(\lambda-\gamma)^2}{\lambda(1-\gamma q)^2}\right| = \left|\frac{q}{(1-\gamma q)^2}\right| \max_{\lambda\in\sigma(C)}\left|\frac{(\lambda-\gamma)^2}{\lambda}\right|. \tag{3.60}$$

Consequently, a simple expression can be obtained for the two parameters ρ^* and $\tilde{\rho}$ defined in (3.17) and (3.18), respectively. In fact, by expanding (3.60) at $q=0$, one readily obtains that

$$\tilde{\rho}_\gamma = \max_{\lambda\in\sigma(C)}\frac{|\lambda-\gamma|^2}{|\lambda|}. \tag{3.61}$$

Similarly, for $q = \mathrm{i}x$, one has that $\rho(q)$ in (3.60) is given by

$$\frac{x}{(1+x^2\gamma^2)}\tilde{\rho}_\gamma, \qquad x \geq 0,$$

which is strictly monotone increasing in $[0,\gamma^{-1})$, and decreasing in (γ^{-1},∞). As a consequence, one obtains that, at $x=\gamma^{-1}$,

$$\rho^*_\gamma = \frac{\tilde{\rho}_\gamma}{2\gamma}. \tag{3.62}$$

In (3.61) and (3.62) the subscript γ has been used to state that the value of such parameters actually depends on the diagonal entry γ in B_2. The above relations allow the derivation of simple criteria for choosing the parameter γ: indeed one may think to choose it in order to minimize either (3.61), or (3.62), or a combination of the two. Concerning the minimization of (3.61) and (3.62), a corresponding result can be derived. In order to state it, let use set

$$\sigma(C) = \{\lambda_j = \varphi_j \, \mathrm{e}^{\mathrm{i}\zeta_j}, \ j = 1, \ldots, r\}, \tag{3.63}$$

and sort the eigenvalues by decreasing arguments as follows (we recall that $\sigma(C) \subset \mathbb{C}^+$),

$$\frac{\pi}{2} > \zeta_1 \geq \cdots \geq \zeta_r > -\frac{\pi}{2}. \tag{3.64}$$

Since the matrix is real, we shall only consider the first $\ell = \lceil r/2 \rceil$ eigenvalues, in the sequel. Let now assume that the moduli of such eigenvalues are strictly increasing, that is,

$$0 < \varphi_1 < \cdots < \varphi_\ell. \tag{3.65}$$

Consequently, the following preliminary result holds true.

Lemma 3.1 *Assume that (3.57) hold true and the eigenvalues of the matrix C satisfy (3.64)-(3.65). Then, for all values of γ greater than or equal to*

$$\hat{\gamma} \equiv \max_{j \in \{2,\ldots,\ell\}} \Psi_j + \sqrt{\Psi_j^2 + \varphi_1 \varphi_j}, \qquad \Psi_j = \frac{\varphi_1 \varphi_j (\cos\zeta_1 - \cos\zeta_j)}{\varphi_j - \varphi_1}, \tag{3.66}$$

one has that

$$\frac{|\lambda_1 - \gamma|^2}{|\lambda_1|} = \max_{j \in \{1,\ldots,\ell\}} \frac{|\lambda_j - \gamma|^2}{|\lambda_j|}. \tag{3.67}$$

Proof Indeed, in order for (3.67) to be satisfied, for all $j > 1$ one must have

$$\frac{|\lambda_j - \gamma|^2}{|\lambda_j|} \leq \frac{|\lambda_1 - \gamma|^2}{|\lambda_1|}.$$

By multiplying both sides by $|\lambda_1 \lambda_j|$, and taking into account (3.65), one then obtains the following second order inequality,

$$\gamma^2 - 2\gamma\Psi_j - \varphi_1\varphi_j \geq 0,$$

which, considering that, because of (3.64), $\Psi_j \leq 0$ and that the discriminant of the equation is positive, is satisfied for all

$$\gamma \geq \Psi_j + \sqrt{\Psi_j^2 + \varphi_1\varphi_j}. \ \square$$

The previous lemma allows us to state the desired results.

Theorem 3.7 *Assume the hypotheses of Lemma 3.1 to be satisfied and, moreover, let $\hat{\gamma}$ be defined according to (3.66). Then:*

1. *the minimum value of* ρ^* *is obtained at* $\gamma = \varphi_1$, *and it is given by*

$$\min_{\gamma>0} \rho^*_\gamma = 1 - \cos\zeta_1, \tag{3.68}$$

provided that

$$\varphi_1 \geq \hat{\gamma}; \tag{3.69}$$

2. *The minimum value of* $\tilde{\rho}$ *is obtained at* $\gamma = \varphi_1 \cos\zeta_1$, *and it is given by*

$$\min_{\gamma>0} \tilde{\rho}_\gamma = \varphi_1 \sin^2\zeta_1, \tag{3.70}$$

provided that

$$\varphi_1 \cos\zeta_1 \geq \hat{\gamma}. \tag{3.71}$$

Proof Let us consider the first point. By taking into account (3.61) and (3.62), we have to solve the problem

$$\min_{\gamma>0} \max_{j\in\{1,\ldots,\ell\}} \frac{|\varphi_j\, \mathrm{e}^{\mathrm{i}\,\zeta_j} - \gamma|^2}{2\gamma\varphi_j}.$$

If such a minimum would be obtained at a value of $\gamma \geq \hat{\gamma}$ (see (3.66)) then, from Lemma 3.1, the previous problem would reduce to the following simpler one,

$$\min_{\gamma>0} \frac{\varphi_1^2 + \gamma^2 - 2\varphi_1\gamma\cos\zeta_1}{2\gamma\varphi_1} = \min_{\gamma>0} \frac{1}{2}\left(\frac{\varphi_1}{\gamma} + \frac{\gamma}{\varphi_1} - 2\cos\zeta_1\right) \equiv \min_{\gamma>0} g^*(\gamma).$$

Consequently, by considering that the only stationary point of g^* is given by $\frac{\mathrm{d}g^*}{\mathrm{d}\gamma}(\varphi_1) = 0$ and, moreover, $\frac{\mathrm{d}^2 g^*}{(\mathrm{d}\gamma)^2}(\varphi_1) > 0$, from (3.69), one then obtains that at $\gamma = \varphi_1$,

$$\rho^*_{\varphi_1} = g^*(\varphi_1) = 1 - \cos\zeta_1.$$

Similarly, for the second point we obtain that

$$\begin{aligned}\min_{\gamma>0} \max_{j\in\{1,\ldots,\ell\}} \frac{|\varphi_j\, \mathrm{e}^{\mathrm{i}\,\zeta_j} - \gamma|^2}{\varphi_j} &= \min_{\gamma>0} \frac{\varphi_1^2 + \gamma^2 - 2\varphi_1\gamma\cos\zeta_1}{\varphi_1} \\ = \min_{\gamma>0} \left(\varphi_1 + \frac{\gamma^2}{\varphi_1} - 2\gamma\cos\zeta_1\right) &\equiv \min_{\gamma>0} \tilde{g}(\gamma),\end{aligned}$$

provided that the minimum is obtained at a value of $\gamma \geq \hat{\gamma}$. Indeed, by considering that the only stationary point of $\tilde{g}$ is given by $\frac{\mathrm{d}\tilde{g}}{\mathrm{d}\gamma}(\varphi_1 \cos\zeta_1) = 0$ and, moreover, $\frac{\mathrm{d}^2\tilde{g}}{(\mathrm{d}\gamma)^2}(\varphi_1 \cos\zeta_1) > 0$, from (3.71), one then obtains that, at $\gamma = \varphi_1 \cos\zeta_1$,

$$\tilde{\rho}_{\varphi_1 \cos\zeta_1} = \tilde{g}(\varphi_1 \cos\zeta_1) = \varphi_1 \sin^2\zeta_1. \square$$

Remark 3.2 *We observe that the above relation (3.66) and (3.69) can be also written as*

$$\frac{\varphi_j}{\varphi_1} + \frac{\varphi_1}{\varphi_j} < 2(1 + \cos\zeta_j - \cos\zeta_1), \qquad j = 2, \ldots, \ell. \tag{3.72}$$

By taking into account (3.64)-(3.65), the previous inequalities are satisfied when all the eigenvalues of the matrix C are contained in the small annulus with internal and external radii given, respectively, by:

$$\varrho_1 = \varphi_1, \qquad \varrho_2 = \varphi_1(1 + 2(\cos\zeta_2 - \cos\zeta_1)). \tag{3.73}$$

A similar conclusion can be obtained from (3.66) and (3.71),

$$\frac{\varphi_j}{\varphi_1} + \frac{\varphi_1}{\varphi_j}\cos^2\zeta_1 < 1 + \cos^2\zeta_1 + 2\cos\zeta_1(\cos\zeta_j - \cos\zeta_1), \qquad j = 2, \ldots, \ell, \tag{3.74}$$

which, however, is more restrictive than (3.72).

It turns out that both results in Theorem 3.7 apply to the methods listed in Table 3.1 (see Table 3.2). Moreover, according to what was stated in Remark 3.2, the eigenvalues of the matrix C are contained in the suitably small annulus with internal and external radii defined in (3.73) (see Figure 3.2).

Let now consider in more details the convergence properties of the blended iteration when $q \to \infty$. We have already remarked that the stiff amplification factor $\rho^{(\infty)}$ for the blended iteration (3.11) is "automatically" zero-valued because of the second property in (3.15). In addition to this, the previous analysis allows to measure the rate at which $\rho(q)$ decays to 0 as $q \to \infty$. As matter of fact, when $|q| \gg 1$ and $\gamma > \hat{\gamma}$, from (3.60)-(3.61) and (3.67) one easily obtains

$$\rho(q) \approx \frac{\tilde{\rho}}{\gamma^2|q|} \equiv \frac{\tilde{\rho}^{(\infty)}}{|q|}, \qquad \text{where } \tilde{\rho}^{(\infty)} \equiv \frac{\tilde{\rho}}{\gamma^2} = \frac{2\rho^*}{\gamma}. \tag{3.75}$$

As a consequence, the previously defined parameter $\tilde{\rho}^{(\infty)}$ is a further amplification factor describing the convergence properties of the blended iteration

Table 3.2: Values of the parameters $\hat{\gamma}$, φ_1 and $\varphi_1 \cos\zeta_1$ corresponding to the methods in Table 3.1

Order	r	Padé	$\hat{\gamma}$	φ_1	$\varphi_1 \cos\zeta_1$
4	3	(2,3)	.1233	.7387	.4877
6	4	(2,4)	.1517	.8482	.3994
8	6	(4,6)	.1415	.7285	.2696
10	8	(6,8)	.1376	.6745	.2101
12	10	(8,10)	.1356	.6433	.1752
14	12	(10,12)	.1345	.6227	.1519

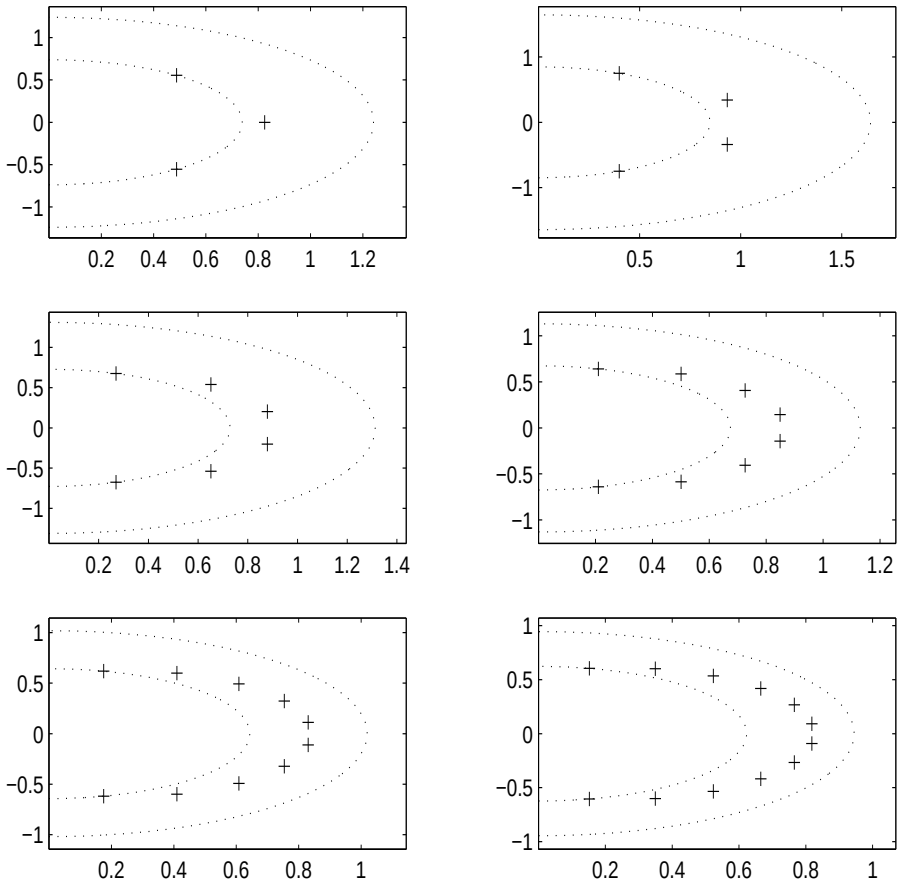


Figure 3.2: Spectrum of the matrix C for the block methods in Table 3.1 and corresponding annuli according to (3.73).

(evidently, the smaller $\tilde{\rho}^{(\infty)}$, the better the L-convergence property of the iteration). In Table 3.3 we list the obtained values for the amplification parameters $\tilde{\rho}$, ρ^* and $\tilde{\rho}^{(\infty)}$ of the iteration corresponding to the two values of γ considered in Theorem 3.7. As one can see, when γ is defined in order to minimize the nonstiff amplification factor, the resulting iteration turns out to be not A-convergent for methods with blocksize greater than 4. Moreover, from (3.61), (3.62) and (3.75) one can easily derive that the value of γ which minimize the maximum amplification factor is the same value which minimize the geometric mean of $\tilde{\rho}$ and $\tilde{\rho}^{(\infty)}$. It represents, therefore, a "good compromise" with the requirement of a fast convergent iteration both when $q \approx 0$ and $|q| \gg 1$. Finally, for completeness, in Table 3.4, the averaged amplification factors (with respect to the infinity matrix norm, see (2.20)) have been also listed.

When the blended implementation does not satisfy (3.58), then the above analysis cannot be applied, since the involved matrices no more commute. In such a case, one must resort to computational techniques in order to minimize either one of the two parameters (3.17) and (3.18). We observe that, from (3.20) and (3.56), one obtains

$$\begin{aligned} Z(q) &= (A_1 - qB_2)^{-1}\Big(\gamma q(1-\gamma q)^{-1}(A_2 - A_1) + q(1-\gamma q)^{-1}(B_1 - B_2)\Big) \\ &= q(1-\gamma q)^{-1}(A_1 - qB_2)^{-1}\Big(B_1 - B_2 + \gamma(A_2 - A_1)\Big), \end{aligned}$$

so that:

- when $q \approx 0$,
$$Z(q) \approx qA_1^{-1}\Big(B_1 - B_2 + \gamma(A_2 - A_1)\Big) \equiv qR; \tag{3.76}$$
- when $|q| \gg 1$,
$$Z(q) \approx \frac{1}{\gamma q}B_2^{-1}\Big(B_1 - B_2 + \gamma(A_2 - A_1)\Big) \equiv \frac{1}{q}R^{(\infty)}. \tag{3.77}$$

It follows that the amplification factors $\tilde{\rho}$ and $\tilde{\rho}^{(\infty)}$ are given, respectively, by the spectral radius of the above matrices R and $R^{(\infty)}$. Concerning alternative choices for the matrices A_1 and B_2, we have considered the case where

$$A_1 = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \qquad B_2 = \gamma I. \tag{3.78}$$

Table 3.3: Asymptotic amplification factors for the methods satisfying (3.58) with the choices $\gamma = \varphi_1$ and $\gamma = \varphi_1 \cos\zeta_1$ respectively.

Order	r	Padé	γ	ρ^*	$\tilde{\rho}$	$\tilde{\rho}^{(\infty)}$
4	3	(2,3)	.7387	.3398	.5021	.9201
6	4	(2,4)	.8482	.5291	.8975	1.2476
8	6	(4,6)	.7285	.6299	.9177	1.7295
10	8	(6,8)	.6745	.6885	.9288	2.0413
12	10	(8,10)	.6433	.7276	.9361	2.2621
14	12	(10,12)	.6227	.7560	.9415	2.4282
4	3	(2,3)	.4877	.4273	.4168	1.7524
6	4	(2,4)	.3994	.8262	.6601	4.1372
8	6	(4,6)	.2696	1.1660	.6287	8.6504
10	8	(6,8)	.2101	1.4492	.6091	13.7918
12	10	(8,10)	.1752	1.6993	.5956	19.3942
14	12	(10,12)	.1519	1.9272	.5856	25.3685

Table 3.4: Averaged amplification factors for the blended iteration with diagonal splitting ($\gamma = \varphi_1$).

Order	r	Padé	ρ_1^*	ρ_3^*	ρ_5^*	ρ_9^*
4	3	(2,3)	1.7311	.5820	.4693	.4066
6	4	(2,4)	2.7844	1.0301	.7895	.6608
8	6	(4,6)	7.2986	1.3512	.9614	.7949
10	8	(6,8)	18.9785	2.2408	1.4146	1.0123
12	10	(8,10)	54.1473	4.8439	1.8673	1.3061
14	12	(10,12)	167.4919	8.9677	2.5791	1.4882
Order	r	Padé	$\tilde{\rho}_1$	$\tilde{\rho}_3$	$\tilde{\rho}_5$	$\tilde{\rho}_9$
4	3	(2,3)	2.5575	.8598	.6933	.6007
6	4	(2,4)	4.7233	1.7473	1.3392	1.1209
8	6	(4,6)	10.6335	1.9686	1.4007	1.1582
10	8	(6,8)	25.6036	3.0230	1.9085	1.3657
12	10	(8,10)	69.6657	6.2321	2.4024	1.6805
14	12	(10,12)	208.5873	11.1680	3.2119	1.8533

Table 3.5: Asymptotic and averaged amplification factors for the methods satisfying (3.78) with minimized maximum amplification factor ρ^*.

Order	r	Padé	γ	ρ^*	$\tilde{\rho}$	$\tilde{\rho}^{(\infty)}$
4	3	(2,3)	.6884	.2686	.3366	.6248
6	4	(2,4)	.8351	.4045	.4513	.6220
8	6	(4,6)	.7677	.5235	.4747	.8598
10	8	(6,8)	.6151	.5468	.6032	2.0516
12	10	(8,10)	.6046	.6482	.6884	3.6684
14	12	(10,12)	.5819	.7417	.7462	5.7378
Order	r	Padé	ρ_1^*	ρ_3^*	ρ_5^*	ρ_9^*
4	3	(2,3)	.9685	.4047	.3462	.3086
6	4	(2,4)	2.2948	.6252	.5285	.4698
8	6	(4,6)	7.3490	.8836	.7174	.6237
10	8	(6,8)	16.2617	1.5806	1.1758	.8672
12	10	(8,10)	52.4037	1.9002	1.4913	1.1513
14	12	(10,12)	167.8829	3.8893	1.9866	1.3105
Order	r	Padé	$\tilde{\rho}_1$	$\tilde{\rho}_3$	$\tilde{\rho}_5$	$\tilde{\rho}_9$
4	3	(2,3)	1.4230	.4912	.4089	.3799
6	4	(2,4)	3.0605	.6621	.5761	.5185
8	6	(4,6)	9.5122	.9024	.6768	.6270
10	8	(6,8)	18.2097	1.1149	.8629	.7251
12	10	(8,10)	63.9458	1.9968	1.3118	.9862
14	12	(10,12)	205.1073	4.1068	1.7443	1.1984

In Table 3.5 and in Table 3.6 we list the obtained results when choosing γ in order to minimize ρ^* and $\sqrt{\tilde{\rho}\|R\|_2}$, respectively. The infinity matrix norm has been used for the computation of the averaged amplification factors. The second criteria, for choosing γ, has been adopted in order to "improve" the convergence properties of the iteration when $q \approx 0$ and, at the same time, to obtain increasing values for $\tilde{\rho}$ when the order of the method increase. This property, in fact, turns out to be useful for an efficient variable order implementation of the methods. As told before, in such a case the parameters have been numerically computed. We observe that the choice of minimizing $\sqrt{\tilde{\rho}\|R\|_2}$ makes the method corresponding to $r = 12$ not A-convergent (though $A(\alpha)$-convergent with $\alpha \approx \pi/2$).

Hereafter, we shall refer to the following three blended schemes:

1. A_1 and B_2 as in (3.58) and γ chosen in order to minimize ρ^*;
2. A_1 and B_2 as in (3.78) and γ chosen in order to minimize ρ^*;

Table 3.6: Asymptotic and averaged amplification factors for the methods satisfying (3.78) with minimized $\sqrt{\tilde{\rho}\|R\|_2}$.

Order	r	Padé	γ	ρ^*	$\tilde{\rho}$	$\tilde{\rho}^{(\infty)}$
4	3	(2,3)	.5802	.3020	.2692	.8638
6	4	(2,4)	.5960	.5441	.3833	1.2018
8	6	(4,6)	.5165	.6719	.4310	1.8221
10	8	(6,8)	.4472	.7860	.4389	2.1402
12	10	(8,10)	.4088	.9112	.4408	2.6821
14	12	(10,12)	.3866	1.0010	.4583	4.1523
Order	r	Padé	ρ_1^*	ρ_3^*	ρ_5^*	ρ_9^*
4	3	(2,3)	.9034	.4194	.3683	.3370
6	4	(2,4)	1.4129	.8798	.7153	.6336
8	6	(4,6)	4.1309	1.1826	.9404	.8096
10	8	(6,8)	8.8690	2.3211	1.5682	1.1289
12	10	(8,10)	24.7268	3.5040	2.3807	1.4794
14	12	(10,12)	77.7625	5.4215	3.5041	1.8642
Order	r	Padé	$\tilde{\rho}_1$	$\tilde{\rho}_3$	$\tilde{\rho}_5$	$\tilde{\rho}_9$
4	3	(2,3)	1.0846	.4445	.3679	.3177
6	4	(2,4)	1.2992	.6259	.5871	.4831
8	6	(4,6)	2.8751	.9453	.6415	.5428
10	8	(6,8)	5.8632	2.2464	.9509	.6319
12	10	(8,10)	14.7973	3.9311	1.9095	.7842
14	12	(10,12)	42.8203	6.6707	3.3158	1.0183

3. A_1 and B_2 as in (3.78) and γ chosen in order to minimize $\sqrt{\tilde{\rho}\|R\|_2}$,

as the *type 1, 2* and *3 schemes*, respectively. A comparative analysis of Table 3.3 and Table 3.4 with Table 3.5 and Table 3.6 puts into evidence the type 2 schemes as the ones with the best features from the point of view of the amplification factors, with the only exception of the factors $\tilde{\rho}^{(\infty)}$ corresponding to the last two higher order methods. On the other hand, the type 1 schemes allows to carry out a complete spectral analysis of the amplification matrix. Moreover, the diagonal splittings characterizing such schemes make them very appealing for an implementation on parallel computers.

3.4 Numerical experiments

In order to compare the performances of the proposed blended schemes on some reference stiff problems taken from the CWI testset [79] for ODE

solvers, a Matlab code had been realized. In particular, the code implemented variable-stepsize and variable-order strategies for the methods in Table 3.1 (we do not discuss here the details of such implementation since they will be fully described in the next chapter).

We here report the results obtained on the following well-known test problems:

- Van der Pol, of size $m = 2$, stiff parameter $\mu = 1000$, and $[t_0, T] = [0, 1000]$;
- Robertson, of size $m = 3$, and $[t_0, T] = [0, 4 \cdot 10^6]$;
- Pollution, of size $m = 20$, and $[t_0, T] = [0, 60]$;
- Ring Modulator, of size $m = 15$, parameter $C_s = 10^{-9}$, and $[t_0, T] = [0, 10^{-3}]$;

In Tables 3.7, 3.8, 3.9, and 3.10, some statistics concerning the integration of the previous four problems with the type 1, 2, 3 schemes previously described have been reported. In such tables, for each run, we list: the values of the input tolerances `atol` and `rtol` for, respectively, the absolute and the relative error of the numerical solution, and the initial stepsize h_0. Moreover, in such tables we count as 1 step one single application of the block methods. Finally, the precision of the numerical solution is measured with the number of *significant correct digits*, defined as

$$\text{scd} \equiv -\log_{10} \|(y - y_{\text{true}})./y_{\text{true}}\|_\infty, \tag{3.79}$$

where y denotes the numerical solution at $t = T$, while y_{true} is a known reference solution. The operator ./ used in (3.79) denotes the componentwise ratio.

In addition, in Figures 3.3, 3.4, 3.5, and 3.6, the corresponding *Work-Precision Diagrams* have been plotted with the work measured either in terms of function evaluations or of solved linear systems. The input tolerances, used for the diagrams, were:

$$\texttt{atol} = \texttt{rtol} = 10^{-(2+k)}, \quad k = 0, \ldots, 10,$$

and the initial stepsize was: $h_0 = 10^{-6}$ for the Van der Pol, the Robertson, and the Pollution problems, and $h_0 = 10^{-8}$ for the Ring Modulator problem.

The previous results show that, in spite of the different values of the amplification factors of the corresponding iteration, the three schemes are able to provide comparable results for the considered test problems.

Table 3.7: Results for the Van der Pol problem.

Type 1 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	3.98	113	102	1583	3178
10^{-6}	10^{-6}	6.50	112	109	2166	4340
10^{-8}	10^{-6}	9.69	178	177	3967	7942
10^{-10}	10^{-6}	11.05	160	158	5405	10814
Type 2 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	3.81	86	75	1383	2770
10^{-6}	10^{-6}	6.53	112	110	2393	4794
10^{-8}	10^{-6}	9.27	166	166	3750	7508
10^{-10}	10^{-6}	10.92	140	138	5271	10554
Type 3 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	4.05	81	71	1350	2700
10^{-6}	10^{-6}	6.12	111	111	2326	4668
10^{-8}	10^{-6}	9.06	146	145	3710	7420
10^{-10}	10^{-6}	11.91	140	138	5389	10778

Table 3.8: Results for the Robertson problem.

Type 1 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	4.06	54	54	723	1446
10^{-6}	10^{-6}	5.67	99	99	1365	2730
10^{-8}	10^{-6}	8.01	86	86	2192	4384
10^{-10}	10^{-6}	9.80	130	130	3624	7248
Type 2 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	3.91	55	55	690	1380
10^{-6}	10^{-6}	5.96	106	106	1321	2642
10^{-8}	10^{-6}	8.30	81	81	2103	4206
10^{-10}	10^{-6}	10.10	110	110	3386	6772
Type 3 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	3.37	55	55	690	1380
10^{-6}	10^{-6}	5.69	108	108	1346	2692
10^{-8}	10^{-6}	8.04	81	81	2124	4248
10^{-10}	10^{-6}	10.48	106	106	3406	6812

Table 3.9: Results for the Pollution problem.

Type 1 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	4.08	17	17	198	396
10^{-6}	10^{-6}	5.99	31	31	381	762
10^{-8}	10^{-6}	7.96	48	48	767	1534
10^{-10}	10^{-6}	9.31	49	49	1116	2232
Type 2 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	4.49	17	17	189	378
10^{-6}	10^{-6}	5.36	30	30	354	708
10^{-8}	10^{-6}	8.24	48	48	724	1448
10^{-10}	10^{-6}	10.16	49	49	1100	2200
Type 3 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-6}	4.64	17	17	192	384
10^{-6}	10^{-6}	5.63	24	24	351	702
10^{-8}	10^{-6}	7.62	31	31	681	1362
10^{-10}	10^{-6}	10.17	49	49	1083	2166

Table 3.10: Results for the Ring Modulator problem.

Type 1 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-8}	3.09	1366	1341	25982	52028
10^{-6}	10^{-8}	5.01	1831	1765	43176	86500
10^{-8}	10^{-8}	6.65	2068	1982	65376	131004
10^{-10}	10^{-8}	9.51	2581	2506	96084	192376
Type 2 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-8}	2.54	1359	1325	26309	52754
10^{-6}	10^{-8}	4.77	1580	1500	41217	82626
10^{-8}	10^{-8}	6.98	2008	1925	64140	128612
10^{-10}	10^{-8}	9.40	2295	2199	92015	184334
Type 3 scheme						
atol=rtol	h_0	scd	steps	accept	f-eval	lin-sys
10^{-4}	10^{-8}	2.71	1413	1398	23376	46760
10^{-6}	10^{-8}	4.93	1590	1518	40391	80842
10^{-8}	10^{-8}	7.57	2198	2126	65124	130388
10^{-10}	10^{-8}	8.49	2937	2845	97731	195626

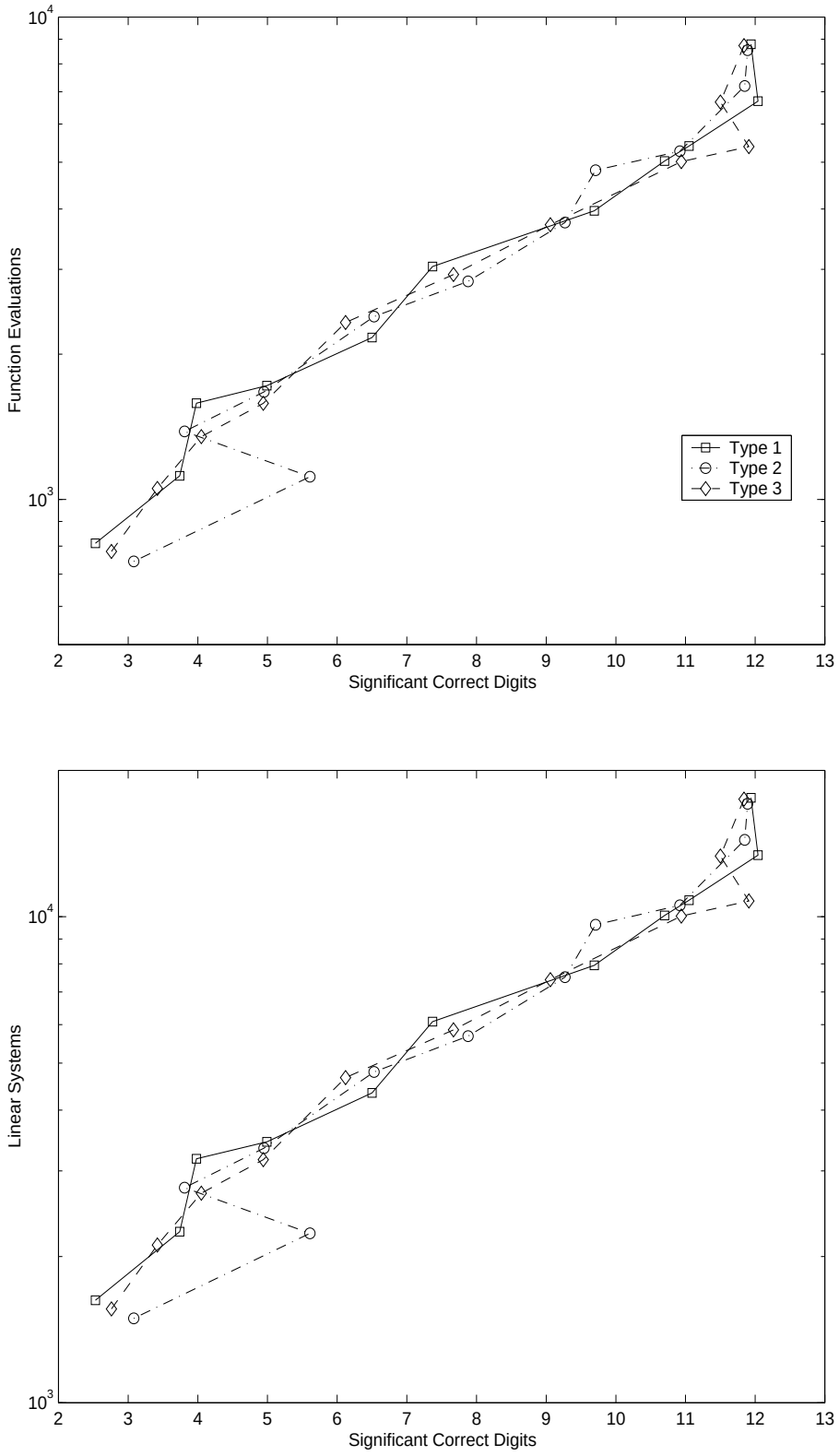


Figure 3.3: Work-Precision Diagrams for the Van der Pol problem.

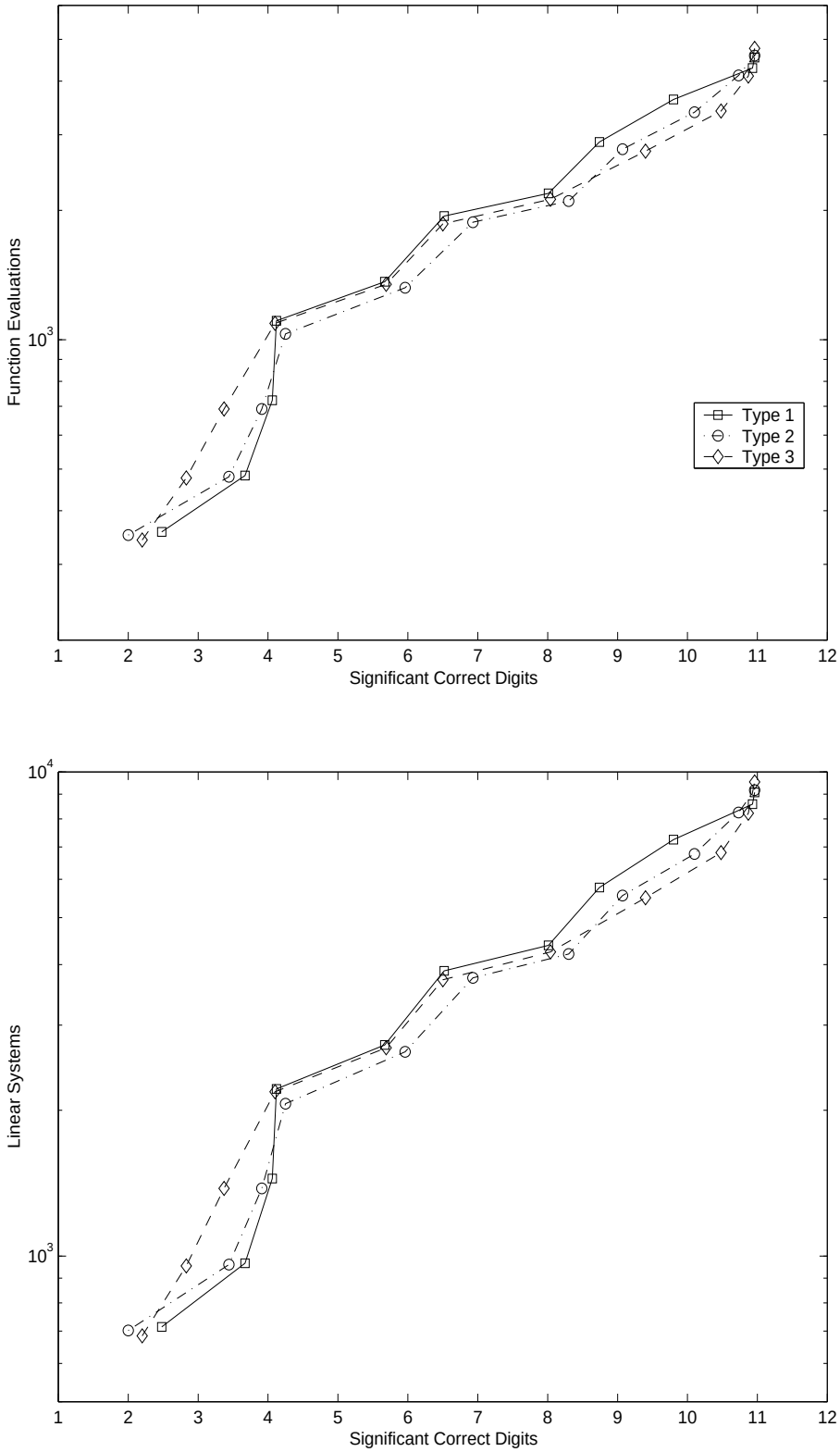


Figure 3.4: Work-Precision Diagrams for the Robertson problem.

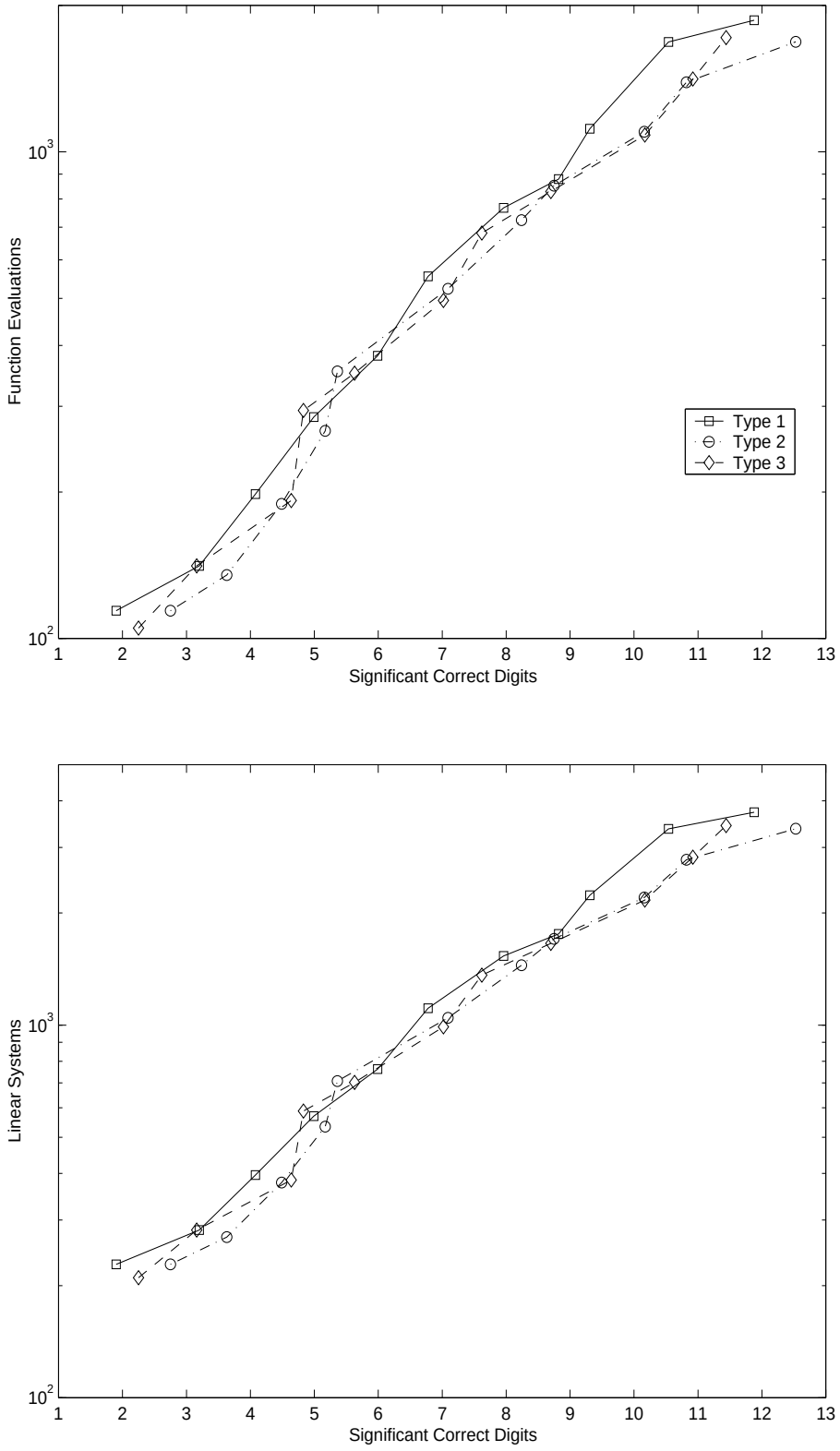


Figure 3.5: Work-Precision Diagrams for the Pollution problem.

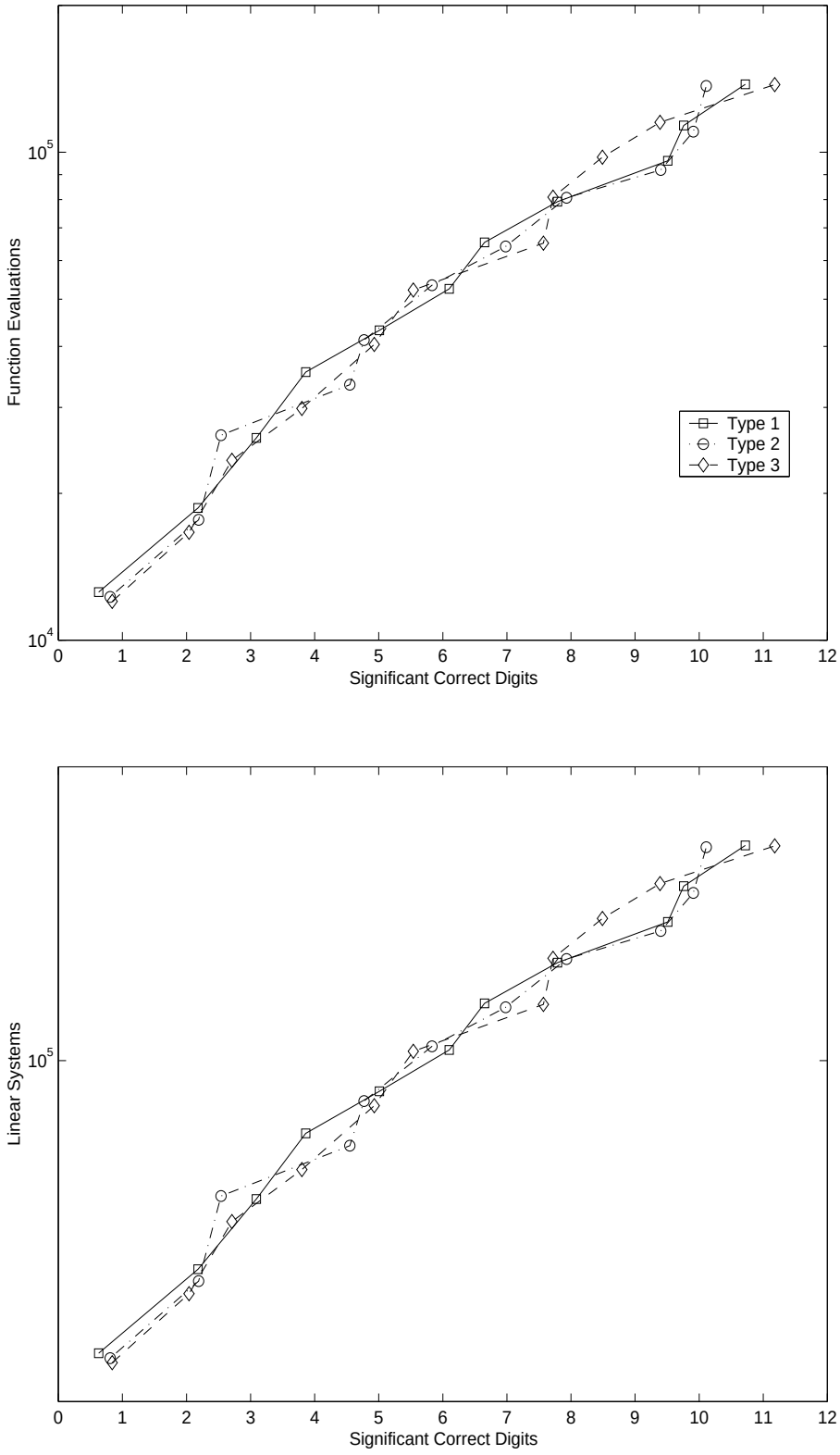


Figure 3.6: Work-Precision Diagrams for the Ring Modulator problem.

Chapter 4

The code BiM

Computational codes represent an outstanding technological aspect of the Mathematical Sciences. Moreover, these codes constitute basic tools for *problem solving* in applied fields. The construction of such codes requires, in turn, the systematic solution of a number of related sub-problems, which constitute the intermediate steps to reach the desired goal. This aspect of Numerical Mathematics is usually underestimated and considered to be only of secondary importance. On the contrary, it is a source of new trends of investigation and a necessary *building block* to make Mathematics usable from people involved in solving real-life problems.

With this premise, and in light of the numerical results provided by the Matlab prototype mentioned in Section 3.4, we decided to implement in the code `BiM` the blended implicit methods with splitting matrices given by

$$A_1 = I_r, \qquad B_2 = \gamma\, I_r, \tag{4.1}$$

i.e., the *type 1 schemes* introduced in the previous chapter. As we are going to discuss in full details in the present chapter, the diagonal structure of the corresponding nonlinear splittings has allowed to construct a computational code for which almost all of the implementation strategies are supported by a linear analysis of convergence of the iteration. In addition to this, the perfect degree of parallelism of such splittings, for what concern the function evaluations and the system solvings, makes these schemes very appealing for an implementation on parallel computers.

For later reference, we recall that when the parameter γ is selected in order to minimize the maximum amplification factor of the iteration, the following results apply (see Section 3.3):

$$\gamma = |\lambda_1| \equiv \min_{\lambda \in \sigma(C)} |\lambda|, \qquad \rho^* = 1 - \cos\zeta_1, \qquad \tilde{\rho} = 2\gamma\rho^*. \tag{4.2}$$

Moreover, for sake of brevity, hereafter the following notation will be used for the current block of integration, since the reported analysis equally applies to each application of the block method:

- (t_0, y_0) for the initial point of the block,
- $t_1, \ldots, t_r$ for the internal abscissae,
- the vectors

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_r \end{pmatrix}, \quad f_j = f(t_j, y_j),$$

  for the current numerical solution.

The organization of this chapter is the following: in Section 4.1 we discuss the nonlinear blended iteration applied to problem (1.2); Section 4.2 concerns with the local error estimate used in the code, on which both the variation of the stepsize and of the order of the method rely. The details of the latter are then discussed in Section 4.3. The problem of the eventual re-evaluation of the Jacobian and/or of the factorization involved in the nonlinear splitting is addressed in Section 4.4.

4.1 The nonlinear iteration

We start considering in full detail the nonlinear iteration generated by the blended implicit methods with splitting matrices as in (4.1) applied to problem (2.1). In such a case, the blended iteration (3.11) becomes

$$\begin{aligned} \Delta\mathbf{y}^{(i)} &= -\theta\Big(\theta\left((I-\gamma C^{-1})\otimes I_m\mathbf{y}^{(i)} - h(C-\gamma I)\otimes I_m\mathbf{f}^{(i)}\right) \\ &\quad +\gamma\left(C^{-1}\otimes I_m\mathbf{y}^{(i)} - hI\otimes I_m\mathbf{f}^{(i)}\right) - \boldsymbol{\eta}\Big), \qquad (4.3) \\ \mathbf{y}^{(i+1)} &= \mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}, \qquad i = 0, 1, \ldots, \end{aligned}$$

where

$$\mathbf{y}^{(i)} = \begin{pmatrix} y_1^{(i)} \\ \vdots \\ y_r^{(i)} \end{pmatrix}, \qquad \mathbf{f}^{(i)} = \begin{pmatrix} f_1^{(i)} \\ \vdots \\ f_r^{(i)} \end{pmatrix}, \quad f_j^{(i)} = f(t_j, y_j^{(i)}),$$

the vector $\boldsymbol{\eta}$ only depends on the initial condition, and, if J_0 denotes the Jacobian of f at (t_0, y_0),

$$\theta = I\otimes\Omega^{-1}, \qquad \Omega = (I_m - h\gamma J_0). \qquad (4.4)$$

Consequently, if ν iterations are performed to obtain convergence, the overall computational cost is approximately given by:

- the evaluation of the Jacobian matrix J_0,
- the factorization of the $m \times m$ matrix Ω in (4.4),
- $r\nu$ function evaluations and
- $2r\nu$ system solvings with the factors of the matrix Ω.

Let us now briefly sketch the choice of the starting vector $\mathbf{y}^{(0)}$ and the stopping criterion for the iteration (4.3). Concerning the first point, the adopted strategy is similar to that used in most of the available codes: the default profile is obtained by using the interpolating polynomial over the previous block of points; alternatively, we use a constant initial vector (namely, the starting point repeated r times) in either one of the following cases:

- when we integrate over the very first block;
- after a failure of the iteration;
- when the solution is very slowing varying. This last condition is recognized when, on the last block (whose size is r, if the order has not been changed), the following test is true:

$$\forall j = 1, \dots m : \frac{|y_{rj} - y_{0j}|}{1 + |y_{0j}|} < \min\{10^{-2}, 10^2 * \mathtt{tol}_j\} \quad \text{and} \quad \|f_r\|_\infty < 0.5, \tag{4.5}$$

where $\mathtt{tol}_j \equiv \mathtt{rtol}$ (the prescribed relative tolerance) if $|y_{0j}| > 10^{-1}$, $\mathtt{tol}_j \equiv \mathtt{atol}$ (the prescribed absolute tolerance) if $|y_{0j}| \leq 10^{-1}$ and, in general, $y_{\ell j}$ is the jth entry of y_ℓ.

Let us now analyze the stopping criterion for the iteration (4.3). Let us consider the vector $\Delta\mathbf{y}^{(i)}$, as defined in that equation, and introduce the norm

$$\|\Delta\mathbf{y}^{(i)}\| \equiv \max_{\ell=1,\dots,r} |\Delta\mathbf{y}_\ell^{(i)}| \equiv \max_{\ell=1,\dots,r} \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\Delta y_{\ell j}}{1 + \mathtt{ratol}|y_{0j}|}\right)^2}, \tag{4.6}$$

where $\mathtt{ratol} = \frac{\mathtt{rtol}}{\mathtt{atol}}$ is the ratio between the specified relative (`rtol`) and absolute (`atol`) tolerances, and y_0 is the starting point for the current block. Then, the iteration ends as soon as the following condition is satisfied,

$$\|\Delta\mathbf{y}^{(i)}\| \leq \max\left\{c, \frac{\mathtt{uround}}{\mathtt{rtol}}\right\} * \mathtt{atol}, \tag{4.7}$$

where `uround` is the machine precision (on input, $\mathtt{rtol} > \mathtt{uround}$) and the parameter $c = 0.1$. Moreover, in order to make more restrictive the stopping

Table 4.1: Values of various parameters for the methods implemented in the code `BiM`.

p	r	Padé	γ	ρ^*	$\tilde{\rho}$	$\tilde{\rho}^{(\infty)}$	`maxit`	`faterr`
4	3	(2,3)	.7387	.3398	.5021	0.9201	10	7
6	4	(2,4)	.8482	.5291	.8975	1.2476	12	6
8	6	(4,6)	.7285	.6299	.9177	1.7295	14	5
10	8	(6,8)	.6745	.6885	.9288	2.0413	16	4
12	10	(8,10)	.6433	.7276	.9361	2.2621	18	3
14	12	(10,12)	.6227	.7560	.9415	2.4282	20	–

criterion when the solution has small entries and/or is slowly varying, the value of the parameter c may be decreased as follows:

- when $\|y_0\|_{-\infty} \equiv |y_{0s}| < 10^{-2}$, $|f_{0s}| < 10^{-4}$, and $\|f_0\|_\infty < 10^{-3}$, then $c = 5 \cdot 10^{-3}$;
- when (4.5) holds true, then $c = \min\{c, 5 \cdot 10^{-2}\}$.

The iteration (4.3) fails if the condition (4.7) is not satisfied within `maxit` iterations, where this parameter depends on the method currently used, according to Table 4.1. The iteration also fails if $i > 2$ and $\rho^{(i)} > 0.99$, where $\rho^{(i)}$ is the estimate of the spectral radius of the iteration matrix at the ith iterate. Such an estimate is obtained, after at least two iterations, as follows:

$$\rho^{(1)} = \frac{\|\Delta \mathbf{y}^{(1)}\|}{\|\Delta \mathbf{y}^{(0)}\|}, \qquad \rho^{(i)} = \sqrt{\rho^{(i-1)} \frac{\|\Delta \mathbf{y}^{(i)}\|}{\|\Delta \mathbf{y}^{(i-1)}\|}}, \quad \text{if } i \geq 2. \tag{4.8}$$

In case of failure of the iteration (4.3) the order of the method is decreased (if $r > 3$) and the stepsize is halved.

4.2 The local error estimate

The algorithm used in the code `BiM` for the estimate of the local error is based on deferred correction. We observe that the latter is a useful framework for error estimation when solving ODEs [39, 41, 77, 78, 87, 88, 98, 99, 101, 102, 105]. Its main use is to provide a tool for the iterative improvement of the numerical solution. This approach has been successfully used in numerical codes for BVPs (see, for example, [41, 77]), where it is used to obtain an approximation of the global error. Nevertheless, when solving

IVPs, such an approach may be also used to estimate local errors, in connection with mesh-selection (see, e.g., [13, 20]). This is exactly the use of deferred correction which has been considered in the code `BiM`.

We remark, once more, that, since equivalent methods provide the same numerical solution, they do have the same corresponding local error. We can, therefore, assume, without loss of generality, the following normalization for the matrix A and consequently, from consistency (see (3.27)), for the vector $\mathbf{a}$ in (3.22),

$$A = I_r, \qquad \mathbf{a} = -\mathbf{1} \equiv -\begin{pmatrix} 1 \dots 1 \end{pmatrix}^T. \tag{4.9}$$

Conversely, one easily realizes that the vector $\boldsymbol{\tau}$ with the truncation errors (see (3.49)) depends on the particular form of the discrete problem. As a matter of fact, from the definition given in (3.49), it is not difficult to prove that if the couples of matrices (A_1, B_1) and (A_2, B_2) define two equivalent methods, then the following equality must hold

$$(A_1^{-1} \otimes I_m)\boldsymbol{\tau}_1 = (A_2^{-1} \otimes I_m)\boldsymbol{\tau}_2,$$

where $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are the vectors with the truncation errors of the two methods, respectively. In the sequel, we will denote with $\boldsymbol{\tau}$ the vector corresponding to the block method with the normalization in (4.9), i.e. the method written in the first equivalent form (see (3.58)) in the blended implementation. The vector corresponding to the second equivalent form is, therefore, given by

$$\gamma\, C^{-1} \otimes I_m \boldsymbol{\tau}. \tag{4.10}$$

Moreover, as discussed in Section 3.2, the basic block method (3.23) is defined in order to have the equations on each row with an $O(h^{r+1})$ local truncation error. Therefore, provided the local continuous solution $y(t)$ is suitably regular, $\boldsymbol{\tau}$ admits the expansion (see (3.50))

$$\boldsymbol{\tau} = \mathbf{v}_{r+1} \otimes h^{r+1} y^{(r+1)}(t_0) + \mathbf{v}_{r+2} \otimes h^{r+2} y^{(r+2)}(t_0) + \dots. \tag{4.11}$$

Consequently, (see, for example, [59, pag. 123]) a first order approximation to the local error is given by (see (4.4))

$$\mathbf{e} = \theta\, \boldsymbol{\tau}. \tag{4.12}$$

It follows that we can obtain an efficient estimate of the local error once an estimate of the local truncation error $\boldsymbol{\tau}$ is available. For this purpose, let us recall that it is possible to uniquely define two $r \times (r+1)$ matrices $[\tilde{\mathbf{a}} \,|\, \tilde{A}]$ and $[\tilde{\mathbf{b}} \,|\, \tilde{B}]$,

$$
\begin{aligned}
[\tilde{\mathbf{a}} \,|\, \tilde{A}] &\equiv \left(\begin{array}{c|ccc} -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{array}\right), \\
[\tilde{\mathbf{b}} \,|\, \tilde{B}] &\equiv \left(\begin{array}{c|ccc} \tilde{\beta}_0^{(1)} & \tilde{\beta}_1^{(1)} & \dots & \tilde{\beta}_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \tilde{\beta}_0^{(r)} & \tilde{\beta}_1^{(r)} & \dots & \tilde{\beta}_r^{(r)} \end{array}\right),
\end{aligned}
\tag{4.13}
$$

such that the coefficients on each row of the two matrices define an r-step LMF with an $O(h^{r+2})$ truncation error (see Theorem 3.2). Deferred correction is then implemented by *plugging in* the numerical solution in the discrete problem defined by the block method (4.13), thus obtaining (see, for example, [13, 20])

$$
I_r \otimes I_m \mathbf{y} - h\tilde{B} \otimes I_m \mathbf{f} - \mathbf{1} \otimes y_0 - h\tilde{\mathbf{b}} \otimes f_0 \approx -\boldsymbol{\tau}. \tag{4.14}
$$

The leading term in the arithmetic complexity for the local error estimate is therefore given by the solution of the r linear systems with the factors of Ω required in (4.12) (see also (4.4)). Moreover, the estimate of $\boldsymbol{\tau}$ requires to include the matrix $\tilde{B}$ in the data structure of the code. We are now going to prove that, because of the properties of the methods used in the code `BiM`, deferred correction allows a noticeable *short cut* in its actual implementation. This result will be proved in the more general case of block implicit methods with internal abscissae:

$$
t_0 + c_1 h, \quad \dots, \quad t_0 + c_r h,
$$

where $0 < c_1 < \ldots < c_r$ (in particular, for the methods implemented in the code `BiM`, one has $c_i = i$, $i = 1, \ldots, r$). From the analysis reported in Section 3.2, it can be seen that the matrix D_r in (3.29) and the vectors $\mathbf{q}_i$ in (3.26) generalize to

$$
D_r = \mathrm{diag}(\, c_1 \, \ldots \, c_r \,), \qquad \mathbf{q}_i = D_r^i \mathbf{1}.
$$

The order conditions (3.27) and (3.28), with $p = r$, then become:

$$
D_r \mathbf{1} - \mathbf{b} - B\mathbf{1} = \mathbf{0}, \tag{4.15}
$$

$$
D_r^i \mathbf{1} - iBD_r^{i-1}\mathbf{1} = \mathbf{0}, \qquad i = 2, \ldots, r. \tag{4.16}
$$

We observe that, from (4.15), the vector $\mathbf{b}$ turns out to be uniquely determined, provided all LMF are consistent, by the choice of the matrix B. The

latter turns out to be uniquely determined by the order conditions (4.16) and by fixing its spectrum (see Section 3.2). Moreover, for $i = r+1$, (4.16) becomes

$$D_r^{r+1}\mathbf{1} - (r+1)\, B\, D_r^r \mathbf{1} = \mathbf{w}_{r+1} \equiv \begin{pmatrix} w_{r+1,1} \\ \vdots \\ w_{r+1,r} \end{pmatrix}, \tag{4.17}$$

where

$$\mathbf{w}_{r+1} \equiv (r+1)!\ \mathbf{v}_{r+1}. \tag{4.18}$$

The vector $\mathbf{v}_{r+1}$ contains the leading coefficients of the truncation errors of the LMF corresponding to each equation of the block method (see (3.50) and (3.53) for the particular case $c_i = i$, $i = 1, \ldots, r$). Then, from (4.16) and (4.17), it is not difficult to obtain

$$D_r^2 V - B D_r V G = \mathbf{w}_{r+1}\mathbf{e}_r^T, \tag{4.19}$$

where

$$G = \mathrm{diag}(\,2\ \ldots\ r+1\,), \qquad V = \begin{pmatrix} 1 & c_1^1 & \ldots & c_1^{r-1} \\ \vdots & \vdots & & \vdots \\ 1 & c_r^1 & \ldots & c_r^{r-1} \end{pmatrix}. \tag{4.20}$$

Since the abscissae $\{c_i\}$ are supposed to be distinct, the Vandermonde matrix V in (4.20) turns out to be nonsingular. Consequently, one immediately obtains

$$B = \left(D_r^2 V - \mathbf{w}_{r+1}\mathbf{e}_r^T\right) G^{-1} V^{-1} D_r^{-1}. \tag{4.21}$$

Now, in order to apply deferred correction, we need an additional couple of matrices in the form (4.13), whose rows define r-step LMF of order (at least) $r+1$, defined over the same set of abscissae $\{c_i\}$. The corresponding order conditions are, therefore, given by:

$$D_r\mathbf{1} - \tilde{\mathbf{b}} - \tilde{B}\mathbf{1} = \mathbf{0}, \tag{4.22}$$

$$D_r^i\mathbf{1} - i\,\tilde{B}\, D_r^{i-1}\mathbf{1} = \mathbf{0}, \qquad i = 2, \ldots, r+1. \tag{4.23}$$

Similarly to what seen in (4.15), now (4.22) uniquely defines the vector $\tilde{\mathbf{b}}$, once $\tilde{B}$ is fixed. For the latter matrix, from (4.23) one readily obtains that

$$\tilde{B} = D_r^2 V G^{-1} V^{-1} D_r^{-1}, \tag{4.24}$$

that is, the matrix is uniquely determined by the order conditions. The latter equation generalizes the result of Theorem 3.2, concerning the particular case $c_i = i,\ i = 1, \ldots, r$.

Let us now report some results concerning the factorization of a Vandermonde matrix (actually, its transpose as it is the matrix V), to be used later. Though some of them are partially known (see, for example, [1]), such results are here cast in the most general and appropriate form for subsequent reference. For this purpose, we need to introduce the following notations:

- $\omega_j(x) = \prod_{k=1}^{j-1}(x - c_k),\ j = 1, \ldots, r$, is the jth Newton polynomial defined by the considered abscissae;
- $x^j[c_1, \ldots, c_i]$ is the divided difference of the function x^j over the abscissae $c_1, \ldots, c_i$.

The following basic properties are also recalled, for sake of completeness:

P1: $\omega_j(c_i) = 0$, if $i < j$;

P2: $x^{j-1}[c_1, \ldots, c_i] = 0$, for $j < i$; $\quad x^{j-1}[c_1, \ldots, c_j] = 1$.

An easy consequence of the above properties is the following result.

Lemma 4.1 *The matrices*

$$L = (w_j(c_i))_{i,j=1,\ldots,r}\,, \qquad U = \left(x^{j-1}[c_1, \ldots, c_i]\right)_{i,j=1,\ldots,r}\,, \tag{4.25}$$

are lower and unit upper triangular, respectively.

Then, the following result follows.

Lemma 4.2 *Let V, L, U be defined according to (4.20) and (4.25). Then,*

$$V = LU. \tag{4.26}$$

Proof In fact, for all $i, j = 1, \ldots, r$, one has:

$$\mathbf{e}_i^T\, LU\mathbf{e}_j = \sum_{k=1}^{r} \omega_k(c_i) x^{j-1}[c_1, \ldots, c_k] = c_i^{j-1},$$

where the last equality is due to the fact that the corresponding left-hand side is the interpolating polynomial of the function x^{j-1}, over the abscissae $c_1, \ldots, c_r$, evaluated at c_i. □

Lemma 4.3 *The inverse of the matrix L in (4.25) is given by*

$$L^{-1} = (\ell_{ij}) \equiv \begin{cases} 0, & \text{if} \quad j > i, \\ \dfrac{1}{\prod_{k=1,k\neq j}^{i}(c_j - c_k)}, & \text{if} \quad j \leq i. \end{cases}$$

Proof By considering that

$$\ell_{ii} \equiv (\omega_i(c_i))^{-1}, \qquad i = 1, \ldots, r,$$

and that both L and L^{-1} are lower triangular, it is then sufficient to prove that

$$\mathbf{e}_i^T \, L^{-1} L \mathbf{e}_j = 0, \qquad \text{for} \quad i > j.$$

In such a case, by taking into account **P1**, one obtains:

$$\mathbf{e}_i^T \, L^{-1} L \mathbf{e}_j = \sum_{\nu=1}^{i} \frac{w_j(c_\nu)}{\prod_{k=1,k\neq\nu}^{i}(c_\nu - c_k)} = w_j[c_1, \ldots, c_i] = 0,$$

where the last equality follows from the fact that, for $j < i$, the polynomial w_j has degree less than or equal to $i - 2$. □

From Lemma 4.3, the following result follows.

Corollary 4.1 *Let $g(t)$ be a given function and let $g_i = g(t_0 + c_i h)$, $i = 1 \ldots, r$. Then,*

$$L^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_r \end{pmatrix} = \begin{pmatrix} h^0 g[t_0 + c_1 h] \\ \vdots \\ h^{r-1} g[t_0 + c_1 h, \ldots, t_0 + c_r h] \end{pmatrix}.$$

Proof From Lemma 4.3, for all $i = 1, \ldots, r$, one obtains that

$$\begin{aligned} \mathbf{e}_i^T \, L^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_r \end{pmatrix} &= \sum_{\nu=1}^{i} \frac{g_\nu}{\prod_{k=1,k\neq\nu}^{i}(c_\nu - c_k)} \\ &= h^{i-1} \sum_{\nu=1}^{i} \frac{g_\nu}{\prod_{k=1,k\neq\nu}^{i}(c_\nu - c_k) h} \\ &= h^{i-1} g[t_0 + c_1 h, \ldots, t_0 + c_i h]. \end{aligned}$$ □

Now, we are going to prove the result which will allow us to significantly simplify the procedure for the local error estimate, thus providing the "short cut" previously mentioned. Moreover, such result clearly quantifies the approximation to the truncation error provided by the left-hand side of equation (4.14).

Theorem 4.1 *Let $c_0 = 0$ and $g(t)$ be any function such that*

$$g(t_0 + c_i h) = f(t_0 + c_i h, y_i), \qquad i = 0, \ldots, r. \tag{4.27}$$

Then, (see (4.18)),

$$\begin{aligned} & I_r \otimes I_m \mathbf{y} - h\tilde{B} \otimes I_m \mathbf{f} - \mathbf{1} \otimes y_0 - h\tilde{\mathbf{b}} \otimes f_0 = \\ & = \quad -\frac{h^{r+1}}{r+1} \mathbf{w}_{r+1} \otimes g[t_0 + c_0 h, \ldots, t_0 + c_r h]. \end{aligned} \tag{4.28}$$

Remark 4.1 *By considering that the discrete solution is an $O(h^{r+1})$ approximation to the (local) solution at the grid points, and recalling that (see (2.1)) $y' = f(t, y)$, one easily realizes that, under suitable smoothness assumptions for f,*

$$g[t_0 + c_0 h, \ldots, t_0 + c_r h] = \frac{1}{r!} y^{(r+1)}(t_0) + O(h).$$

From (4.18), it then follows that (4.28) provides a first order approximation to the leading term at the right-hand side of equation (4.11).

Proof The numerical solution satisfies the discrete problem

$$I_r \otimes I_m \mathbf{y} - hB \otimes I_m \mathbf{f} - \mathbf{1} \otimes y_0 - h\mathbf{b} \otimes f_0 = \mathbf{0}. \tag{4.29}$$

Therefore, by subtracting (4.29) from the left-hand side of (4.28), and by setting

$$\tilde{\mathbf{f}} = \begin{pmatrix} f_0 \\ \mathbf{f} \end{pmatrix} \equiv \begin{pmatrix} f_0 \\ \vdots \\ f_r \end{pmatrix},$$

where $f_i = f(t_0 + c_i h, y_i) \equiv g(t_0 + c_i h),\ i = 0, \ldots, r$, from (4.15)–(4.24) we obtain:

$$\begin{aligned} & I_r \otimes I_m \mathbf{y} - h\tilde{B} \otimes I_m \mathbf{f} - \mathbf{1} \otimes y_0 - h\tilde{\mathbf{b}} \otimes f_0 = \\ & = \quad h([\mathbf{b} \,|\, B] - [\tilde{\mathbf{b}} \,|\, \tilde{B}]) \otimes I_m \tilde{\mathbf{f}} \\ & = \quad h(B - \tilde{B})[-\mathbf{1} \,|\, I_r] \otimes I_m \tilde{\mathbf{f}} \\ & = \quad -h\mathbf{w}_{r+1} \mathbf{e}_r^T G^{-1} V^{-1} D_r^{-1} [-\mathbf{1} \,|\, I_r] \otimes I_m \tilde{\mathbf{f}} \\ & = \quad -\frac{h}{r+1} \mathbf{w}_{r+1} \mathbf{e}_r^T V^{-1} D_r^{-1} [-\mathbf{1} \,|\, I_r] \otimes I_m \tilde{\mathbf{f}} \quad = \quad (*). \end{aligned}$$

From (4.25)-(4.26), property **P2**, Corollary 4.1, and considering that $c_0 = 0$, one then obtains:

$$
\begin{aligned}
(*) &= -\frac{h}{r+1}\mathbf{w}_{r+1}\mathbf{e}_r^T\, U^{-1}L^{-1}D_r^{-1}[-\mathbf{1}\,|\,I_r]\otimes I_m\tilde{\mathbf{f}} \\
&= -\frac{h}{r+1}\mathbf{w}_{r+1}\mathbf{e}_r^T\, L^{-1}D_r^{-1}[-\mathbf{1}\,|\,I_r]\otimes I_m\tilde{\mathbf{f}} \\
&= -\frac{h}{r+1}\mathbf{w}_{r+1}\mathbf{e}_r^T\, L^{-1}\begin{bmatrix} \frac{-1}{c_1} & \frac{1}{c_1} & & \\ \vdots & & \ddots & \\ \frac{-1}{c_r} & & & \frac{1}{c_r} \end{bmatrix}\otimes I_m\tilde{\mathbf{f}} \\
&= -\frac{h^2}{r+1}\mathbf{w}_{r+1}\mathbf{e}_r^T\, L^{-1}\begin{bmatrix} \frac{-1}{(c_1-c_0)h} & \frac{1}{(c_1-c_0)h} & & \\ \vdots & & \ddots & \\ \frac{-1}{(c_r-c_0)h} & & & \frac{1}{(c_r-c_0)h} \end{bmatrix}\otimes I_m\tilde{\mathbf{f}} \\
&= -\frac{h^2}{r+1}\mathbf{w}_{r+1}\mathbf{e}_r^T\, L^{-1}\otimes I_m\begin{pmatrix} g[t_0+c_0h, t_0+c_1h] \\ \vdots \\ g[t_0+c_0h, t_0+c_rh] \end{pmatrix} \\
&= -\frac{h^{r+1}}{r+1}\mathbf{w}_{r+1}\otimes g[t_0+c_0h,\ldots,t_0+c_rh].\ \square
\end{aligned}
$$

Since the vector $\mathbf{w}_{r+1}$ is known, see (4.18) and (3.50)-(3.53), from the previous theorem it follows that we can directly compute the divided difference at the right-hand side of equation (4.28), in order to obtain the estimate of $\boldsymbol{\tau}$ via deferred correction. This implies that the matrix $\tilde{B}$ is no longer required. In particular, when $c_i = i$, $i = 0,\ldots,r$, as it happens for the methods implemented in the code `BiM`, one obtains (see (4.18) and (4.27))

$$
\frac{h^{r+1}}{r+1}\mathbf{w}_{r+1}\otimes g[t_0+c_0h,\ldots,t_0+c_rh] = \mathbf{v}_{r+1}\otimes(h\Delta^r f_0)\,, \tag{4.30}
$$

where, here, Δ represents the (componentwise) difference operator. Moreover, see (4.4), the first order approximation (4.12) to the local error reduces to

$$
\mathbf{e} = \theta\,\boldsymbol{\tau} = -\mathbf{v}_{r+1}\otimes\left(h\Omega^{-1}\Delta^r f_0\right), \tag{4.31}
$$

so that it can be obtained at the cost of only one linear system solving with the factors of Ω. From the previous analysis, it follows that each block entry of the vector $\mathbf{e}$ is $O(h^{r+1})$, provided that the corresponding entry of the vector $\mathbf{v}_{r+1}$ is nonzero. Nevertheless, from Theorem 3.5 and (3.55), we observe that, since the last entry in $\mathbf{v}_{r+1}$ is 0, the last block entry in (4.31), say e_r, is 0 as well, whereas we need an $O(h^{p+1})$ approximation, if p is the order of the method. In order to obtain a corresponding suitable approximation also for e_r, we then consider the last block entry of the vector (see (4.4), (4.10)-(4.11))

$$\theta(I-\theta)^s(\gamma C^{-1}\mathbf{v}_{r+1}\otimes h\Delta^r f_0), \tag{4.32}$$

where $s=1$, when $r=3$, and $s=2$, otherwise. This entry turns out to be the one of largest norm and this feature will be useful for what we shall see in Section 4.3.1, when speaking about the handling of the "order reduction" phenomenon for stiff problems.

4.3 Stepsize and Order Variation

In this section we describe the strategies for the variation of both the stepsize of integration h and the order p of the method. Both strategies rely on the estimate of the local error previously discussed.

First of all, the norm used to measure the error is the same norm defined in (4.6). As a consequence, on one hand, from (4.4), (4.12), (4.31), and (4.32) one obtains that

$$\|\mathbf{e}\| = \max\left\{v_\infty^r\,|\Omega^{-1}\delta^{(r)}(f_0)|, |e_r|\right\} = O(h^{r+1}), \tag{4.33}$$

where $v_\infty^r \equiv \|\mathbf{v}_{r+1}\|_\infty$ and $\delta^{(r)}(f_0)\equiv h\Delta^r f_0$. On the other hand, the quantity

$$|e_r| = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{e_{rj}}{1+\texttt{ratol}|y_{0j}|}\right)^2} = O(h^{p+1}), \tag{4.34}$$

already computed to obtain (4.33), provides an estimate for $\|\mathbf{e}_{\rm up}\|$, namely the error corresponding to the use of the next higher-order method. This feature will be conveniently exploited when we shall speak about the order variation strategy. Before that, let us consider the problem of the stepsize variation in detail. If `rtol` and `atol` are the prescribed relative and absolute tolerances, the current solution is accepted provided that (see (4.33))

$$\|\mathbf{e}\| \le \texttt{atol}. \tag{4.35}$$

The new stepsize, to be used by the same method, is then obtained through extrapolation:

$$h_{\rm new} = h\left(\texttt{sftyerr}*\frac{\texttt{atol}}{\|\mathbf{e}\|}\right)^{\frac{1}{r+1}}, \tag{4.36}$$

where $\texttt{sftyerr}=\frac{1}{20}$ if (4.35) holds true and $\texttt{sftyerr}=\frac{1}{10}$ otherwise. Similarly, if $r<12$ the stepsize to be used by the next higher-order method would be

$$h_{\rm up} = h\left(\texttt{sftyup} * \frac{\texttt{atol}}{\|\mathbf{e}_{\rm up}\|}\right)^{\frac{1}{p+1}}, \tag{4.37}$$

where the approximation $\|\mathbf{e}_{\rm up}\| = |e_r|$ is used (see (4.34)) and, moreover, we have set $\texttt{sftyup} = \texttt{sftyerr}/2$. We shall use such an estimate for the stepsize of the higher order method when discussing the order variation strategy. Moreover, by denoting with $\hat{h}_{\rm new}$ the selected stepsize for the subsequent integration step and with $r_{\rm new}$ the blocksize of the corresponding method to be used, we set

$$\hat{h}_{\rm new} \leftarrow \min\{\max\{\hat{h}_{\rm new}, 0.12 * h\}, 10 * h, h_{\max}, (T - t_0)/r_{\rm new}\},$$

where, by default, $h_{\max} \equiv (T - \hat{t_0})/8$ being $\hat{t_0}$ the initial time of the IVP. In addition to this, if $0.1 * h \leq t_0 * \texttt{uround}$ (the machine precision), then the execution ends because the selected stepsize is too small. Finally, we also use the following heuristics: if `nfail` consecutive failures have occurred (either for the convergence of the iteration or for the accuracy) before the last successful step, then the stepsize is increased only after at least $\texttt{nfail} + 1$ consecutive successful steps occur.

Let us now consider the problem of the order variation. The aim is that of reducing the global computational cost for getting a discrete solution with a prescribed accuracy. For this purpose, we normalize the cost with respect to the width of the covered interval. By neglecting, for sake of simplicity, Jacobian and function evaluations, whose cost in general is strongly problem dependent, we then introduce the following *specific cost per step* function for the method with blocksize r:

$$c_{\rm tot}(\nu, r, m, h) = \frac{c_{\rm fact} + c_{\rm it} + c_{\rm err}}{rh}, \tag{4.38}$$

where $c_{\rm fact}$ is the cost for the factorization of the matrix Ω in (4.4), $c_{\rm it}$ is the number of flops required by ν iterations in (4.3), and $c_{\rm err}$ is the cost for computing the estimate (4.33) of the local error. In particular, in case of a full $m \times m$ Jacobian,

$$c_{\rm fact} \approx \frac{2}{3}\,m^3, \quad c_{\rm it} \equiv c_{\rm it}(r, \nu, m) \approx 4r\nu m^2, \quad c_{\rm err} \approx \begin{cases} 4m^2, & \text{if } r = 3, \\ 6m^2, & \text{otherwise.} \end{cases}$$

Corresponding formulae are used in case of a banded Jacobian.

Therefore, the next higher-order method, with blocksize $r_{\rm up}$ (see the second column in Table 4.1), requiring $\nu_{\rm up}$ iterations for satisfying the same stopping criterion, and using a stepsize $h_{\rm up}$ for getting the same accuracy, would be preferable in the subsequent step provided that

$$c_{\rm tot}(\nu_{\rm up}, r_{\rm up}, m, h_{\rm up}) < c_{\rm tot}(\nu_{\rm new}, r, m, h_{\rm new}), \tag{4.39}$$

where $h_{\rm new}$ and $\nu_{\rm new}$ are the stepsize and the number of expected iterations for the current-order method. Therefore, the problem is easily solved, once we have an estimate for the above quantities. We have already seen how to get estimates for $h_{\rm new}$ and $h_{\rm up}$ (see (4.36) and (4.37), respectively). It remains to obtain estimates for $\nu_{\rm new}$ and $\nu_{\rm up}$. We observe that, if the same stopping criterion has to be satisfied, then the following equalities should approximately hold,

$$\rho^{\nu} = (\rho_{\rm new})^{\nu_{\rm new}} = (\rho_{\rm up})^{\nu_{\rm up}}.$$

In the above equation, ρ is the spectral radius of the current iteration matrix (estimated by (4.8)), ν is the (known) number of iterations carried out to satisfy the convergence criterion (4.7), and $\rho_{\rm new}$, $\rho_{\rm up}$ are the spectral radii of the iteration matrices of the current-order method, by using the new stepsize $h_{\rm new}$, and of the next higher-order method, respectively. By taking into account that the stiff amplification factor of both methods is 0, and considering (3.18), we then obtain the following estimates,

$$\nu_{\rm new} = \nu \frac{\log \rho}{\log \rho(h_{\rm new}/h)}, \qquad \nu_{\rm up} = \nu \frac{\log \rho}{\log \rho(\tilde{\rho}_{\rm up}/\tilde{\rho})(h_{\rm up}/h)}, \tag{4.40}$$

where $\tilde{\rho}$ and $\tilde{\rho}_{\rm up}$ are the nonstiff amplification factors of the current and the next higher-order methods, respectively (see Table 4.1). Finally, in order to prevent erratic behaviour in some pathological cases, the previous strategy is applied provided that all the following three conditions are satisfied:

1. $0.8h \le h_{\rm new} \le 1.25h$;

2. at least $\max\{2, \texttt{nfail}\}$ successful consecutive steps have been carried out with the current-order method, when the previous `nfail` steps failed to satisfy the accuracy requirement (4.35);

3. the (estimated) spectral radius of the current iteration, say ρ, is "suitably small". The latter condition is assumed to be fulfilled, provided that $\rho < \rho_p$, where the parameter ρ_p is defined so that all methods do have a prescribed absolute cost to obtain convergence. In more detail, by setting

$$\rho_4 = 10^{-2} |\log_{10} \min\{10^{-1}, \texttt{atol}, \texttt{rtol}\}|, \tag{4.41}$$

we require that, for all allowed orders p, the quantity $c_{\rm it}(r_p, \nu_p, m)$ (see Table 4.1 and (4.38)) is constant, for the same stopping criterion,

where r_p and ν_p are the blocksize and the number of iterations required by the pth order method, $p = 4, 6, 8, 10, 12, 14$. This leads to the equalities,

$$r_p\,\nu_p = r_{p-2}\,\nu_{p-2}, \quad \rho_p^{\nu_p} = \rho_{p-2}^{\nu_{p-2}}, \qquad p = 6, 8, 10, 12, 14,$$

which provide the following recursion with starting value given by (4.41):

$$\rho_p = (\rho_{p-2})^{\frac{r_p}{r_{p-2}}}, \qquad p = 6, 8, 10, 12, 14. \tag{4.42}$$

We observe that the sequence $\{\rho_p\}$ is a decreasing one.

Actually, the last condition is relaxed when $\nu \leq 3$ and both the conditions of *stepsize stagnation* and *convergence stagnation*, as described in Section 4.3.1 below, are verified.

So far, we have dealt with the strategy for increasing the order of the method to be used at the subsequent step of numerical integration. However, it may be convenient to decrease the order of the method as well. Obviously, the criterion based on the minimization of the specific cost per step (4.38) could be, in principle, also used to decrease the order of the method, provided that an estimate for $h_{\rm low}$, namely the stepsize to be used by the next lower-order formula, is available. Its computation, based on a procedure similar to that required for evaluating $h_{\rm new}$, would require an additional linear system with the matrix Ω to be solved. Nevertheless, we decided not to systematically resort to such a criterion for decreasing the order, because there is numerical evidence that it is seldom effective. Instead, we chose to lower the order p to $p-2$ (when $r > 3$, see Table 4.1), in either one of the following two situations:

- a failure of the nonlinear iteration (4.3) occurs (in such a case, $h_{\rm new} = h/2$, as we have already said at the end of Section 4.1);
- all the following four conditions hold true:
  1. in the last step the current-order method has been successful;
  2. the nonlinear iteration (4.3) has required more than 3 iterations;
  3. the (estimated) spectral radius of the iteration matrix, ρ, satisfies $\rho > \rho_p$, where ρ_p is again defined according to (4.42), but with the initial condition, in place of (4.41),

$$\rho_4 = 0.5; \tag{4.43}$$

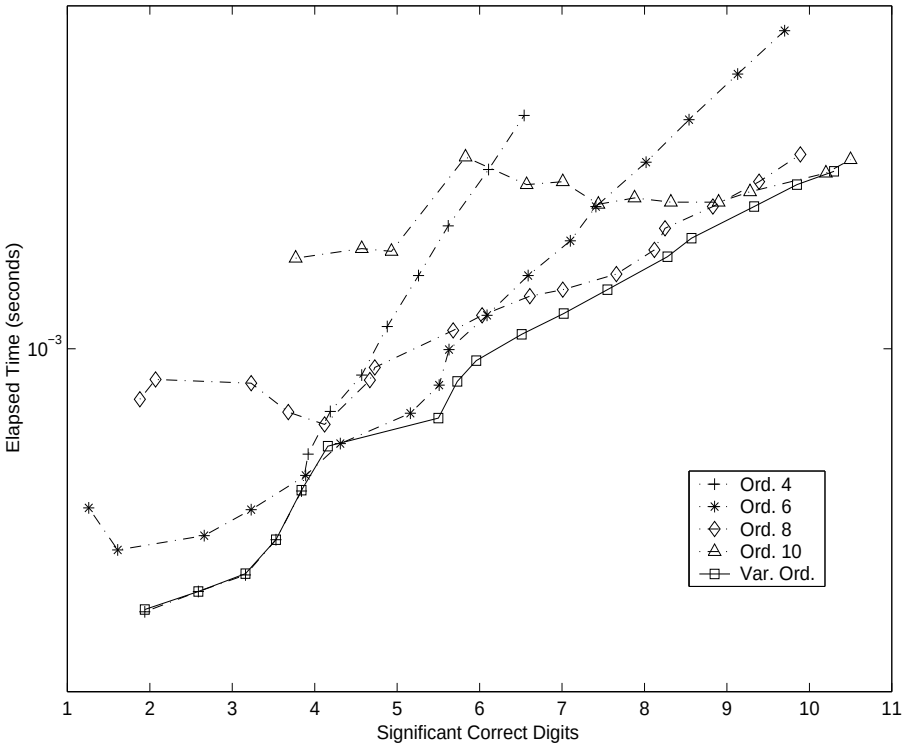


Figure 4.1: Variable versus fixed order implementation.

4. if condition (4.45) below is satisfied, then $h_{\text{low}} \geq h_{\text{new}}$.

In this case, the new stepsize is set equal to:

$$\begin{cases} h_{\text{low}}, & \text{if (4.45) and } h_{\text{low}} \geq h_{\text{new}} \text{ hold true,} \\ \min\{h_{\text{low}}, h_{\text{new}}\}, & \text{otherwise.} \end{cases}$$

In order to put into evidence the effectiveness of the above order variation strategy, in Figure 4.1 the results obtained for the Robertson problem (already introduced in Section 3.4) have been reported. In the figure we plot the elapsed time (in seconds) for the solution of the problem versus the number of significant correct digits (see (3.79)). As one can see, the plot of the variable order method is almost always below those of the fixed order ones, thus confirming the effectiveness of the order variation strategy.

4.3.1 Order reduction recovery

A particular handling is required in order to get rid of the so called *order reduction phenomenon* (see, for example, [59, chapter IV.15]). Such a phe-

nomenon occurs when, in the test equation (3.1), $h \to 0$ but $q = h\mu$ is large. In such a case, in fact, the expansion (4.11) of the truncation error becomes

$$\boldsymbol{\tau} = q^{r+1}\,\mathbf{v}_{r+1}\,y_0 + q^{r+2}\,\mathbf{v}_{r+2}\,y_0 + \ldots,$$

and the local error is given by $(I - qC)^{-1}\boldsymbol{\tau}$. However, the latter expression admits different expansions, depending on the "size" of q. In particular,

- when $|q|$ is small, then

$$(I - qC)^{-1}\boldsymbol{\tau} = q^{r+1}\,\mathbf{v}_{r+1}\,y_0 + q^{r+2}\,(\mathbf{v}_{r+2} + C\mathbf{v}_{r+1})\,y_0 + \ldots,$$

  and the principal term of each entry behaves like q^{r+1}, with the exception of the last one, which depends on higher order terms;

- when $|q|$ is large, then

$$(I - qC)^{-1}\boldsymbol{\tau} \approx -q^r\,C^{-1}\mathbf{v}_{r+1}\,y_0 + \ldots. \tag{4.44}$$

  In such a case, the principal term of each entry behaves like q^r, including the last one.

The conclusions in the latter case make evident the fact that $|e_r|$ (see (4.34)) is no more an estimate for $\|\mathbf{e}_{\rm up}\|$. On the other hand, when q is large, it happens that, see (4.33)-(4.34),

$$\|\mathbf{e}\| = |e_r|, \tag{4.45}$$

i.e., the norm of the last (block) entry of the vector defined in (4.32). Moreover, the latter vector turns out to be an approximation to the principal term of the expansion (4.44). In conclusion, when the order reduction phenomenon occurs, the strategy for the order variation previously described, which relies on the higher order accuracy of the last entry of the local error, may fail. Indeed, this actually happens for the well-known Prothero-Robinson problem (see [89]). In such a case, also the stepsizes stagnate. In the code `BiM`, the order reduction phenomenon is recognized when (4.45) holds true or all the following conditions are satisfied:

order stagnation: the order of the method has not been increased by the above mentioned strategy;

error stagnation: $|e_r|\,$`fateerr` $\geq \|\mathbf{e}\|$, where the parameter `fateerr` is chosen according to Table 4.1. When such a condition holds true, this means that the last entry of the local error is "not too small", with respect to the remaining ones. This is, indeed, usually the case, when it correctly estimates the error for the next higher-order method. The parameter `fateerr` is, at the moment, chosen in a heuristic way;

stepsize stagnation: the ratio between the new stepsize, $h_{\rm new}$, and the current one, h, belongs to the interval $[0.95, 1.05]$;

convergence stagnation: the ratio between the current estimated spectral radius, ρ (see (4.8)), and the one of the previous iteration, $\rho_{\rm old}$, belongs to the interval $[0.95, 1.05]$.

Once the error reduction phenomenon is recognized, it is possible to get rid of it, as explained in the sequel. The basic idea is to obtain an estimate for $\|\mathbf{e}_{\rm up}\|$ in a form similar to (4.33):

$$\|\mathbf{e}_{\rm up}\| \approx v_\infty^{r_{\rm up}} \, |\Omega^{-1}\delta^{(r_{\rm up})}(f)|. \tag{4.46}$$

Indeed, the quantity $v_\infty^{r_{\rm up}}$ is known. Concerning the second term, $\delta^{(r_{\rm up})}(f)$ can be approximated by suitable first (in the case $r = 3$) or second (in the case $r > 3$) differences of $\delta^{(r)}(f)$, since this function has already been computed at the previous blocks. Once the estimate (4.46) is available, the usual formula (4.37) can then be used, in order to predict $h_{\rm up}$.

An additional question needs to be considered, at this point, by observing that, when q is not small, then (3.18) is not valid. The latter approximated equality, in turn, was used in order to predict $\rho_{\rm new}$ and $\rho_{\rm up}$ from the knowledge of ρ, h, $h_{\rm new}$, $h_{\rm up}$ (see (4.40)). However, when q is large we know that, see (3.75),

$$\rho(q) \approx \frac{\tilde\rho^{(\infty)}}{|q|},$$

where the values of the parameter $\tilde\rho^{(\infty)}$ are listed in Table 4.1. The previous result allows us to derive the following estimates for $\nu_{\rm new}$ and $\nu_{\rm up}$, alternative to (4.40):

$$\nu_{\rm new} = \nu\frac{\log\rho}{\log\rho(h/h_{\rm new})}, \qquad \nu_{\rm up} = \nu\frac{\log\rho}{\log\rho(\tilde\rho_{\rm up}^{(\infty)}/\tilde\rho^{(\infty)})(h/h_{\rm up})}, \tag{4.47}$$

where $\tilde\rho^{(\infty)}$ is the parameter of the current-order method, and $\tilde\rho_{\rm up}^{(\infty)}$ is that of the next higher-order one.

Remark 4.2 *It must be stressed that in the estimates (4.47), the ratios $h/h_{\rm new}$ and $h/h_{\rm up}$ are exactly reversed, with respect to those used in (4.40). This is due to the use of the approximation (3.75) in place of (3.18).*

The estimates (4.47) are then used in the check (4.39), in order to decide whether to increase the order of the method to be used in the subsequent step, when the order reduction phenomenon is diagnosed. Finally, we mention that, for robustness, when (4.45) holds true, the order is not increased when the following two conditions are both fulfilled:

- $h_{\rm up} \geq h$;
- the estimated spectral radius for the higher order method,

$$\rho_{\rm up} = \rho \frac{\tilde{\rho}^{(\infty)}_{\rm up}}{\tilde{\rho}^{(\infty)}} \frac{h}{h_{\rm up}}, \tag{4.48}$$

  is larger than the corresponding maximum allowed value, as defined by (4.42)-(4.43).

Indeed, the first condition ensures that the approximation (4.48), derived from (3.75), is appropriate also for the next higher-order method.

4.4 Jacobian evaluation and LU factorization

In Section 4.1 we have already observed that the overall computational cost for the solution of the discrete problem generated by a blended implicit method approximately amounts to:

- the evaluation of J_0, the Jacobian matrix at (t_0, y_0),
- the factorization of the $m \times m$ matrix Ω (see (4.4)),
- $r\nu$ function evaluations,
- $2r\nu$ system solvings with the factors of the matrix Ω,

if ν iterations are required to obtain convergence. Obviously, the relative computational cost of the first two entries, with respect to the overall computational cost, depends on the continuous problem and on ν. In particular, their relative cost increases when ν decreases. Therefore, when the blended iteration (4.3) converges rapidly, the overall computational cost of the iteration can be reduced significantly by means of one of the following approximations:

$$J_0 \approx J_{\rm old}, \qquad \text{and/or} \qquad \Omega \approx \Omega_{\rm old}, \tag{4.49}$$

where $J_{\rm old}$ and $\Omega_{\rm old}$ are the analogues of J_0 and Ω at the previous block of points. It is clear that (see (4.4)) in both cases a perturbation is introduced in the matrix θ and, therefore, the spectral radius of the corresponding iteration matrix turns out to be affected. In the following two sections, we shall study this aspect by means of a linear analysis, which relies on the particular structure of the discrete problem.

4.4.1 The blended iteration with approximate Jacobian

Let us consider the application of the method, corresponding to the blended iteration (4.3), to the test problem:

$$y' = \mu(t)y, \qquad y(t_0) = y_0 \in \mathbb{R}, \qquad Re(\mu(t)) < 0,$$

and let us denote with μ the value of $\mu(t)$ at the initial point of the current sub-interval of integration. Then, we can write

$$\mu = \mu_{\rm old}(1+\delta), \tag{4.50}$$

where $\mu_{\rm old}$ is the corresponding value of μ at the previous block of points and $\delta \in \mathbb{C}$ is a suitable parameter. The approximate blended iteration, corresponding to the use of the previous Jacobian, is therefore given by

$$\begin{aligned} \mathbf{y}^{(i+1)} &= \mathbf{y}^{(i)} - \hat{\theta}\left[\hat{\theta}\left((I-\gamma C^{-1})\mathbf{y}^{(i)} - h(C-\gamma I)\mathbf{f}^{(i)}\right)\right. \\ &\quad \left. +\gamma\left(C^{-1}\mathbf{y}^{(i)} - h\mathbf{f}^{(i)}\right) - \hat{\boldsymbol{\eta}}\right], \qquad i = 0, 1, \ldots, \end{aligned} \tag{4.51}$$

where

$$\hat{\theta} \equiv (1-\gamma\hat{q})^{-1}I, \qquad \hat{q} \equiv h\mu_{\rm old}, \tag{4.52}$$

and (see 3.10),

$$\hat{\boldsymbol{\eta}} \equiv \hat{\theta}(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2) + \boldsymbol{\eta}_2.$$

We shall consider the additional first order approximation $\mathbf{f}^{(i)} \approx \mu\mathbf{y}^{(i)}$ so that the iteration (4.51) can be rewritten as

$$\begin{aligned} \mathbf{y}^{(i+1)} &= \mathbf{y}^{(i)} - \hat{\theta}\left[\left(\hat{\theta}\left(I-\gamma C^{-1} - q(C-\gamma I)\right) + \gamma\left(C^{-1} - qI\right)\right)\mathbf{y}^{(i)} - \hat{\boldsymbol{\eta}}\right] \\ &= \mathbf{y}^{(i)} - \hat{\theta}\left[\left(\hat{\theta}\left(I-\gamma C^{-1} - \hat{q}(1+\delta)(C-\gamma I)\right)\right.\right. \\ &\quad \left.\left. +\gamma\left(C^{-1} - \hat{q}(1+\delta)I\right)\right)\mathbf{y}^{(i)} - \hat{\boldsymbol{\eta}}\right], \qquad i = 0, 1, \ldots, \end{aligned} \tag{4.53}$$

where (see (4.50) and (4.52)) $q \equiv h\mu = \hat{q}(1+\delta)$. The spectral radius of the corresponding iteration matrix depends, therefore, on both $\hat{q}$ and δ: let it be $\hat{\rho}(\hat{q}, \delta)$. The following result holds true.

Theorem 4.2 *If $|\delta| < \bar{\delta}$ with $\bar{\delta}$ sufficiently small, then the spectral radius $\hat{\rho}(\hat{q}, \delta)$ of the amplification matrix, say $Z(\hat{q}, \delta)$, of the iteration (4.53) is such that*

1. *when $\hat{q} \approx 0$*

$$\hat{\rho}(\hat{q}, \delta) \approx |\hat{q}|\ \tilde{\rho}(\delta), \tag{4.54}$$

where

$$\tilde{\rho}(\delta) = \begin{cases} \left| \frac{(\lambda_1-\gamma)^2}{\lambda_1} + \delta\lambda_1 \right|, & \text{when} \quad \mathrm{Im}(\delta) \geq 0, \\ \left| \frac{(\bar{\lambda}_1-\gamma)^2}{\bar{\lambda}_1} + \delta\bar{\lambda}_1 \right|, & \text{when} \quad \mathrm{Im}(\delta) < 0, \end{cases} \tag{4.55}$$

and, see (3.63)-(3.65), λ_1 is the eigenvalue of C with minimum modulus and positive imaginary part;

2. *when $\hat{q} \to \infty$*

$$\hat{\rho}^{(\infty)}(\delta) \equiv \lim_{\hat{q}\to\infty} \hat{\rho}(\hat{q},\delta) = |\delta|. \tag{4.56}$$

Proof The amplification matrix corresponding to (4.53) is given by (see (4.52))

$$\begin{aligned} Z(\hat{q},\delta) &= I - \hat{\theta}^2 \left(I - \gamma C^{-1} - \hat{q}(1+\delta)(C-\gamma I) + \gamma\hat{\theta}^{-1}(C^{-1} - \hat{q}(1+\delta)I) \right) \\ &= \frac{\hat{q}}{(1-\gamma\hat{q})^2} C^{-1} \left((C-\gamma I)^2 + \delta C(C - \gamma^2\hat{q}I) \right). \end{aligned} \tag{4.57}$$

Therefore, since $|\delta|$ is assumed to be bounded, when $\hat{q} \approx 0$ one has

$$Z(\hat{q},\delta) \approx \hat{q}\, C^{-1} \left((C-\gamma I)^2 + \delta C^2 \right),$$

so that

$$\hat{\rho}(\hat{q},\delta) \approx |\hat{q}| \max_{\lambda\in\sigma(C)} \left| \frac{(\lambda-\gamma)^2}{\lambda} + \delta\lambda \right| \equiv |\hat{q}|\, \tilde{\rho}(\delta).$$

We observe that, when $\mu = \mu_{\mathrm{old}}$ or, equivalently, see (4.50), when $\delta = 0$, $\tilde{\rho}(0)$ coincides with the nonstiff amplification factor corresponding to the "exact" blended iteration (see (3.61)), i.e.

$$\tilde{\rho}(0) = \tilde{\rho}. \tag{4.58}$$

From the result in Lemma 3.1, it then follows that, for all $\delta \in \mathbb{C}$ with $|\delta|$ suitably small, $\tilde{\rho}(\delta)$ is obtained in correspondence of λ_1 or of the complex conjugate $\bar{\lambda}_1$. In particular, by considering that $\gamma = |\lambda_1|$ and $\mathrm{Im}(\lambda_1) > 0$, one verifies that

$$\left| \frac{(\lambda_1-\gamma)^2}{\lambda_1} + \delta\lambda_1 \right| \geq \left| \frac{(\bar{\lambda}_1-\gamma)^2}{\bar{\lambda}_1} + \delta\bar{\lambda}_1 \right| \qquad \Leftrightarrow \qquad \mathrm{Im}(\delta) \geq 0.$$

This completes the first part of the proof. On the other hand, when $|\hat{q}| \to \infty$, from (4.57) and the hypothesis on $|\delta|$ one obtains

$$Z(\hat{q},\delta) \to -\delta\, I,$$

from which (4.56) easily follows. □

The previous theorem immediately implies that the blended iteration is no more L-convergent. Nevertheless, one is still able, by estimating $|\delta|$, to control the convergence properties of such iteration when $|\hat{q}| \gg 1$. On the other hand, when $\hat{q} \approx 0$ the following result holds true.

Theorem 4.3 *If $\hat{q} \approx 0$, $\alpha > 0$ is a suitably small fixed parameter, and*

$$|\delta| \leq \frac{\tilde{\rho}(0)\alpha}{(1+\alpha)\tilde{\rho}(0)+\gamma}, \tag{4.59}$$

then

$$\hat{\rho}(\hat{q},\delta) \lesssim \rho(q)(1+\alpha)$$

where $\rho(q)$ is the spectral radius of the iteration matrix with exact Jacobian.

Proof We observe that, since $|\delta|$ is bounded,

$$\hat{q} \approx 0 \qquad \Rightarrow \qquad q = \hat{q}(1+\delta) \approx 0$$

and, therefore, see (3.18) and (4.58)Sk81,Sk86,,

$$\rho(q) \approx \tilde{\rho}(0)|q| = \tilde{\rho}(0)|\hat{q}||1+\delta|.$$

Moreover, when α is suitably small the term on the right-hand side of (4.59) is sufficiently small so that the first result of Theorem 4.2 applies. From (3.61), (4.50), (4.54)-(4.59), and by recalling that $\gamma = |\lambda_1|$, it then follows that

$$\begin{aligned} \hat{\rho}(\hat{q},\delta) \approx |\hat{q}|\,\tilde{\rho}(\delta) &\leq |\hat{q}|(\tilde{\rho}(0)+|\delta|\gamma) \leq |\hat{q}|\,\tilde{\rho}(0)\,(1-|\delta|)(1+\alpha) \\ &\leq |\hat{q}||1+\delta|\,\tilde{\rho}(0)\,(1+\alpha) \approx \rho(q)(1+\alpha). \end{aligned}$$ □

An immediate consequence of the previous two theorems is that an estimate of $|\delta|$ is needed in order to control the perturbation on the spectral radius of the iteration matrix. From (4.50) we obtain $\delta = (\mu - \mu_{\rm old})/\mu_{\rm old}$. Consequently, estimates of $|\mu - \mu_{\rm old}|$ and of $|\mu_{\rm old}|$ are needed. In general, when we are solving problem (1.2), we will need to estimate $\delta = \|J_0 - J_{\rm old}\|/\|J_{\rm old}\|$. By considering a suitable vector $\boldsymbol{\chi}$ such that $||\boldsymbol{\chi}||_\infty = 1$, we then evaluate the vector $g = f(t_0, y_0 + s\cdot\boldsymbol{\chi}) - f_0$, with $s > 0$ a suitably small parameter, thus obtaining the following estimates:

$$\|J_0\|_\infty \approx \frac{1}{s}\|g\|_\infty, \qquad \|J_0 - J_{\rm old}\|_\infty \approx \frac{1}{s}\|g - g_{\rm old}\|_\infty.$$

We observe that, for the linear autonomous equation $y' = J\,y$, one obtains $\|g - g_{\rm old}\|_\infty = 0$, so that the re-evaluation of the Jacobian is not needed, in this case, as one would expect.

Concerning the choice of the parameter α (see (4.59)) made in the code `BiM`, if p is the order of the method with blocksize r_p (see Table 4.1) then the corresponding parameter, say α_p, is chosen as follows:

$$\alpha_4 = 5 \cdot 10^{-2}, \qquad \alpha_p = (\alpha_{p-2})^{\frac{r_p}{r_{p-2}}}, \qquad p = 6, 8, 10, 12, 14.$$

The previous criterion is applied only when, at the previous block of points, (4.45) does not hold true and the blended iteration has been sufficiently "fast" convergent. In particular, by denoting with $\rho_{\rm old}$ and $\nu_{\rm old}$ the spectral radius of the iteration matrix and the number iterations at the previous block of points, respectively, in the code `BiM` a fast convergence is assumed when

$$\rho_{\rm old} < 5 \cdot 10^{-2} \qquad \text{or} \qquad \nu_{\rm old} < 4. \tag{4.60}$$

On the other hand, when (4.45) is satisfied, we assume $|q| \gg 1$ and the Jacobian is not re-evaluated provided that (4.60) holds true and

$$|\delta| \leq \bar{\delta}^{(\infty)},$$

where the value of $\bar{\delta}^{(\infty)}$ depends on the order of the method, as specified in Table 4.2.

The previous analysis, requiring an additional function evaluation to get the estimate of δ, is actually applied provided that $m > 5$ (i.e., the size of the continuous problem is not very small). Moreover, an additional classical control, used in many codes to decide whether the Jacobian should be not evaluated, is also used in the code `BiM`. In more detail, the Jacobian is not evaluated when the blended iteration for the previous block of points turns out to be "very fast" convergent. This is recognized when the following condition is satisfied:

$$\rho_{\rm old} < \rho^J \qquad \text{or} \qquad \nu_{\rm old} < 3,$$

where ρ^J depends on the order of the method, according to the values listed in Table 4.2.

4.4.2 The blended iteration with approximate factorization

We now study the case where the following approximation is considered

$$\Omega \approx \Omega_{\rm old}, \tag{4.61}$$

Table 4.2: Parameters of the methods used in the code `BiM`.

p	r	$\bar{\delta}^{(\infty)}$	ρ^J	x_1	x_2	$d^{\min}$	$d^{\max}$
4	3	$5 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	-1.4487	2.3593	0.90	1.10
6	4	$4 \cdot 10^{-2}$	$4 \cdot 10^{-3}$	-1.4983	3.1163	0.91	1.09
8	6	$3 \cdot 10^{-2}$	$3 \cdot 10^{-3}$	-1.4662	3.5197	0.92	1.08
10	8	$2 \cdot 10^{-2}$	$2 \cdot 10^{-3}$	-1.4290	3.7538	0.93	1.07
12	10	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	-1.3964	3.9104	0.94	1.06
14	12	$9 \cdot 10^{-3}$	$9 \cdot 10^{-4}$	-1.3689	4.0240	0.95	1.05

(see (4.3)-(4.4)), in order not to evaluate the new factorization. First of all, it must be stressed that the previous approximation is allowed only when the Jacobian has not been evaluated since, otherwise, such evaluation would result to be useless. Consequently, we assume that only the stepsize has changed, from the previous iteration. We shall, therefore, resort to a linear analysis of convergence, by applying the method to the test problem (3.1). In such a case, the blended iteration (4.3), with the approximation (4.61), becomes

$$\begin{aligned} \mathbf{y}^{(i+1)} &= \mathbf{y}^{(i)} - \theta_{\rm old} \left[\left(\theta_{\rm old} \left(I - \gamma C^{-1} - q(C - \gamma I) \right) \right. \right. \\ &\quad \left. \left. + \gamma \left(C^{-1} - qI \right) \right) \mathbf{y}^{(i)} - \bar{\boldsymbol{\eta}} \right] \\ &= \mathbf{y}^{(i)} - \theta_{\rm old} \left[\left(\theta_{\rm old} \left(I - \gamma C^{-1} - q_{\rm old}\, d\, (C - \gamma I) \right) \right. \right. \\ &\quad \left. \left. + \gamma \left(C^{-1} - q_{\rm old}\, dI \right) \right) \mathbf{y}^{(i)} - \bar{\boldsymbol{\eta}} \right], \qquad i = 0, 1, \ldots, \end{aligned} \tag{4.62}$$

where $h_{\rm old}$ is the stepsize used for the previous block of points, $q_{\rm old} = h_{\rm old}\mu$,

$$\theta_{\rm old} = (1 - \gamma q_{\rm old})^{-1} I, \qquad q \equiv h\mu = \left(\frac{h}{h_{\rm old}} \right) q_{\rm old} \equiv d\, q_{\rm old}, \tag{4.63}$$

and (see 3.10),

$$\bar{\boldsymbol{\eta}} \equiv \theta_{\rm old}(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2) + \boldsymbol{\eta}_2.$$

Therefore, the spectral radius, say $\bar{\rho}(q_{\rm old}, d)$, of the corresponding iteration matrix will now depend on both $q_{\rm old}$ and d. The following theorem holds true.

Theorem 4.4 *If $|d - 1|$ is sufficiently small, then the spectral radius of the iteration matrix of (4.62) is such that*

1. *when $q_{\rm old} \approx 0$,*

$$\bar{\rho}(q_{\rm old}, d) \approx |q_{\rm old}|\ \tilde{\rho}(d - 1), \tag{4.64}$$

where $\tilde{\rho}(\cdot)$ is defined according to (4.55);

2. when $q_{\rm old} \to \infty$,

$$\bar{\rho}^{(\infty)}(d) \equiv \lim_{q_{\rm old}\to\infty} \bar{\rho}(q_{\rm old}, d) = |d-1|. \tag{4.65}$$

Proof We observe that the iteration (4.62) formally coincides with the iteration (4.52)-(4.53) with the substitutions $\hat{q} \leftarrow q_{\rm old}$ and $\delta \leftarrow d-1$. Consequently, from Theorem 4.2, one immediately obtains $\bar{\rho}(q_{\rm old}, d) = \hat{\rho}(q_{\rm old}, d-1)$, and, hence, the thesis follows. □

From the previous theorem, one immediately obtains that $d \in (0,2)$ is a necessary requirement for a satisfactory behaviour of the iteration for stiff problems. More precisely, when (4.45) holds true, so that we may assume $|q_{\rm old}| \gg 1$, re-factorization is avoided provided that, see Table 4.2,

$$|d-1| \leq \bar{\delta}^{(\infty)}.$$

Let now suppose $q_{\rm old} \approx 0$. The following analysis is devoted to provide an estimate of the number, say $\bar{\nu}$, of iterations in (4.62), depending on the number of iterations ν that would have been required without the approximation (4.61). The latter number can be estimated from the iteration parameters as discussed in Section 4.3 (see (4.40)). In order to derive the criterion used in the code `BiM`, we shall look for values of d (see (4.63)) such that

$$\bar{\nu} \leq \beta\nu, \qquad \beta = 1 + m(6r\nu)^{-1}, \tag{4.66}$$

where r is the blocksize of the blended implicit method and m is the size of the continuous problem. Indeed, for such value of the parameter β, one verifies that the cost of the linear algebra involved in the blended iteration with the approximation (4.61) is less than or equal to the cost of the exact iteration plus the cost to factor Ω (evidently, for sake of simplicity, the cost of function and Jacobian evaluations has been neglected). Moreover, we observe that if the stepsize has not been changed, i.e. $h = h_{\rm old}$, than, see (4.63), $d = 1$ and $q = q_{\rm old}$. In this case, from Theorem 4.4 and (4.58), one immediately obtains

$$\bar{\rho}(q, 1) = \rho(q),$$

where $\rho(q)$ is the spectral radius of the "exact" blended iteration. By assuming that the same stopping criterion has to be satisfied, we then obtain $\bar{\rho}(q_{\rm old}, d)^{\bar{\nu}} = \bar{\rho}(q, 1)^{\nu}$ and, therefore,

$$\bar{\nu} = \nu \frac{\log \bar{\rho}(q, 1)}{\log \bar{\rho}(q_{\rm old}, d)}.$$

Consequently, the inequality in (4.66) can be written as

$$\frac{\bar{\rho}(q_{\rm old}, d)^{\beta}}{\bar{\rho}(q, 1)} \leq 1. \tag{4.67}$$

We observe that (see (4.63)), since d is bounded, then $q_{\text{old}} \approx 0$ implies $q \approx 0$ as well. Therefore (see (4.55)), by setting ρ_{old} the spectral radius of the iteration matrix at the previous integration step, one obtains,

$$\begin{aligned} \bar{\rho}(q_{\text{old}}, d) &\approx |q_{\text{old}}|\tilde{\rho}(d-1) \approx \left(\frac{\rho_{\text{old}}}{\tilde{\rho}(0)}\right)\tilde{\rho}(d-1), \\ \bar{\rho}(q, 1) &\approx |q|\,\tilde{\rho}(0) \approx \rho_{\text{old}}\, d. \end{aligned} \tag{4.68}$$

From (4.67)-(4.68), it follows then that d must satisfy

$$\frac{\tilde{\rho}(d-1)^{\beta}}{d} \leq \rho_{\text{old}} \left(\frac{\tilde{\rho}(0)}{\rho_{\text{old}}}\right)^{\beta}. \tag{4.69}$$

Moreover, since $d-1$ is real, from (4.55) one obtains that (see (4.2))

$$\tilde{\rho}(d-1) \equiv \left|\frac{(\lambda_1-\gamma)^2}{\lambda_1} + (d-1)\lambda_1\right| = \gamma(d^2 + 2x_1 d + x_2)^{\frac{1}{2}}, \tag{4.70}$$

where

$$x_1 = 2\rho^*(\rho^* - 1) - 1, \qquad x_2 = 1 + 4\rho^*.$$

The values of x_1 and x_2 for the methods implemented in the code `BiM` are listed in Table 4.2. From (4.69) and (4.70), it follows that the stepsize ratio d must satisfy

$$\frac{(d^2 + 2x_1 d + x_2)^{\frac{\beta}{2}}}{d} \leq \rho_{\text{old}} \left(\frac{\tilde{\rho}(0)}{\gamma\rho_{\text{old}}}\right)^{\beta}. \tag{4.71}$$

Only one of the following two cases may then occur:

1. $d \geq 1$;
2. $d < 1$.

In the first case, i.e. when the stepsize has been increased, from Table 4.2 it is possible to verify that the inequality (4.71) is satisfied for $\beta = 1$ and $d \in [1, 2)$. Clearly, from (4.66) one obtains that this will hold true for all $\beta \geq 1$. Consequently, (see (4.63)) in the code `BiM` we don't re-factorize, when the stepsize has been increased, unless $d > d^{\max}$ (see Table 4.2), where the last inequality is aimed to guarantee fast convergence for stiff problems and the first result in Theorem 4.4 to hold true.

In the second case, i.e. when the stepsize has been decreased, we can assume $1 > d \geq d^{\min}$, for a fixed $d^{\min} > 0$ (see Table 4.2, for the values used

in the code `BiM`). In such a case, one derives that a sufficient condition for (4.71) to be satisfied is given by

$$d^2 + 2x_1 d + x_3 \leq 0, \tag{4.72}$$

where

$$x_3 = x_2 - (d^{\min}\rho_{\text{old}})^{\frac{2}{\beta}} \left(\tilde{\rho}(0)/(\gamma\rho_{\text{old}})\right)^2.$$

Consequently, in the code `BiM`, re-factorization is avoided, when the stepsize is reduced, unless (4.72) turns out to be not satisfied or $d < d^{\min}$. We observe that, because of (4.65), we have required $|d^{\min} - 1| = |d^{\max} - 1|$.

Chapter 5

Numerical Experiments

During the development of the code `BiM` several numerical experiments on difficult stiff test problems, taken from the CWI testset [79] (now available at the University of Bari [73]) and from the Geneva testset [62], have been performed and, in the following sections, the most significant results are reported. In addition, in order to put into evidence the effectiveness of the proposed approach, such results are compared with those provided by some of the most efficient codes currently available for the numerical solution of stiff IVPs for ODEs:

- `DASSL` (June 1991) implementing Backward Differentiation Formulae of orders from 1 through 5 (L. R. Petzold, [11]);
- `GAM` (November 1999) based on the Generalized Adams Methods of orders 3, 5, 7, 9 (F. Iavernaro and F. Mazzia, [71]);
- `MEBDFDAE` (November 1998) based on the Modified Extended Backward Differentiation Formulae of orders ranging from 1 to 7 (J. Cash, [40]);
- `RADAU5` (January 2002) implementing the Radau IIa implicit Runge-Kutta method of order 5 (E. Hairer and G. Wanner, [59]);
- `RADAU` (January 2002) which is a variable-order version of `RADAU5` implementing the Radau IIa implicit Runge-Kutta methods of orders 5, 9 and 13 (E. Hairer and G. Wanner, [59]);

All executions have been carried out on a dedicated AMD Duron 1.3GHz computer, under Linux, and by using, for each code, the same compiler option `-O3` for optimization. Numerical experiments have been performed by using different values for the input parameters consisting of: the stepsize h_0 to be used for the first step (not needed for `DASSL`) and the prescribed absolute (`atol`) and relative (`rtol`) tolerances for the numerical solution. In the following sections, for each problem, we report:

- a brief introduction describing the origin of the problem and the corresponding mathematical formulation. The reader interested in further details may find them in the cited references;
- the run characteristics of some tests performed with the problem. They consist of the following statistics describing the numerical integration: *steps*, providing the total number of steps needed by the solver (including the rejected steps due to error test failures and/or convergence test failures); *accept*, giving the number of accepted steps; *f-eval* and *j-eval* representing, respectively, the total number of function and jacobian evaluations, and *LU-dec* for the total number of LU-decompositions (not available for `DASSL`). Concerning the latter one, we remark that the values reported in correspondence of the codes `BiM`, `GAM` and `MEBDFDAE` refers to the factorizations of matrices with the same dimension m of the continuous problem. The `RADAU` and `RADAU5` codes, instead, count (at most) 1 factorization per step. We recall that such codes require, at each step, the factorization of 1 real $m \times m$ matrix and $(r-1)/2$ $m \times m$ complex ones, where r is the blocksize of the method (see Section 2.2.1 and [60]). A comparison based on the number of LU-decomposition must, therefore, take care of this fact.
  In addition, for each run, we report the elapsed time (in seconds) needed for the integration and the precision of the numerical solution y with respect to a reference one, say y_{true}, at the end of the integration interval. The latter is measured both in terms of the significant correct digits (*scd*), already defined in Section 3.4, and of the mixed-error significant correct digits (*mescd*), defined as

$$\text{mescd} \equiv -\log_{10}\left(\|(y - y_{\text{true}})./(\texttt{artol} + |y_{\text{true}}|)\|_\infty\right),$$

  where $\texttt{artol} \equiv \frac{\texttt{atol}}{\texttt{rtol}}(1 \ldots 1)^T \in \mathbb{R}^m$ and ./ represents the componentwise ratio operator.
- the Work-Precision Diagrams (WPDs) plotting the "work", measured in terms of the elapsed-time required for the integration, versus the "precision" measured in terms of both *scd* and *mescd*.

5.1 The elastic Beam problem

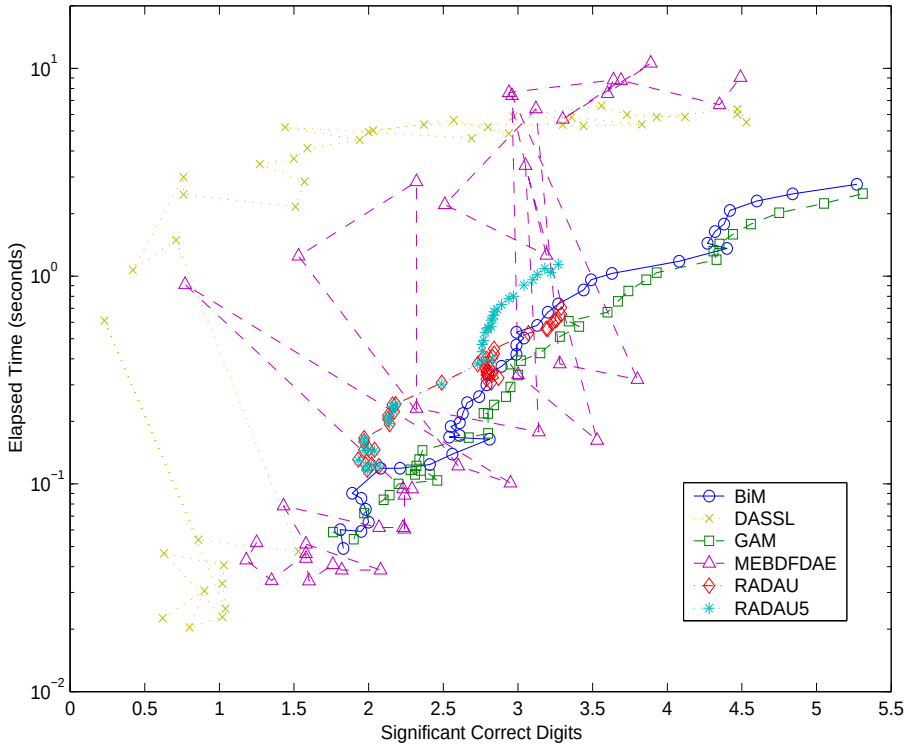
The problem originates from mechanics and describes the motion of a thin elastic beam of length 1 which is supposed inextensible. Moreover, it is assumed that the beam is clamped at one end and a force F acts at the free end. It was originally described by a partial differential equation subject to boundary conditions. The semi-discretization in space of this equation leads to a stiff system of n nonlinear second-order differential equations which is rewritten to first order form thus providing a stiff system of nonlinear ODEs of size $2n$. The eigenvalues of the corresponding Jacobian are purely imaginary and vary between -6400i and 6400i. A complete description of the problem can be found in [59].

Numerical experiments on this problem have been done for $n = 40$ (leading to a system of 80 ODEs). Moreover, the equation has been integrated for $0 \leq t \leq 5$. Table 5.1 and Figure 5.1 present, respectively, the corresponding run characteristics and the work-precision diagrams. For the latter ones we used: $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(2+m/8)}$, $m = 0, \ldots, 40$.

We remark the high regularity of the WPDs corresponding to the codes `BiM` and `GAM`. The widely chaotic behaviour of the code `MEBDFDAE` and the high inefficiency of the code `DASSL` are mainly due to the lack of A-stability of the higher order formulae on which such codes are based.

Table 5.1: Run characteristics for the Elastic Beam problem (h_0 = atol = rtol).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BiM	10^{-2}	1.83	2.18	14	14	289	12	14	$4.88 \cdot 10^{-2}$
	10^{-4}	2.63	3.45	63	63	1224	58	61	$2.18 \cdot 10^{-1}$
	10^{-6}	4.08	4.73	332	332	7038	301	312	$1.18 \cdot 10^{0}$
DASSL	10^{-2}	0.62	1.45	63	60	101	7		$2.26 \cdot 10^{-2}$
	10^{-4}	1.57	1.98	28473	28269	30714	276		$2.86 \cdot 10^{0}$
	10^{-6}	3.36	4.20	53079	52532	58352	650		$5.84 \cdot 10^{0}$
GAM	10^{-2}	1.76	2.03	16	15	485	14	16	$5.89 \cdot 10^{-2}$
	10^{-4}	2.80	3.65	51	49	1793	47	51	$2.16 \cdot 10^{-1}$
	10^{-6}	3.93	4.92	244	242	8699	237	244	$1.04 \cdot 10^{0}$
MEBDFDAE	10^{-2}	1.25	1.52	57	55	740	8	8	$2.59 \cdot 10^{-2}$
	10^{-4}	2.23	2.49	274	270	2514	26	26	$9.45 \cdot 10^{-2}$
	10^{-6}	3.19	4.02	4622	4620	30577	303	303	$1.25 \cdot 10^{0}$
RADAU	10^{-2}	1.99	2.59	23	20	176	16	23	$1.18 \cdot 10^{-1}$
	10^{-4}	2.49	3.57	62	55	406	43	61	$3.13 \cdot 10^{-1}$
	10^{-6}	2.84	3.73	58	58	847	41	55	$4.53 \cdot 10^{-1}$
RADAU5	10^{-2}	1.99	2.59	23	20	176	16	23	$1.17 \cdot 10^{-1}$
	10^{-4}	2.49	3.57	62	55	406	43	60	$3.02 \cdot 10^{-1}$
	10^{-6}	2.89	3.77	162	148	1114	95	139	$7.31 \cdot 10^{-1}$



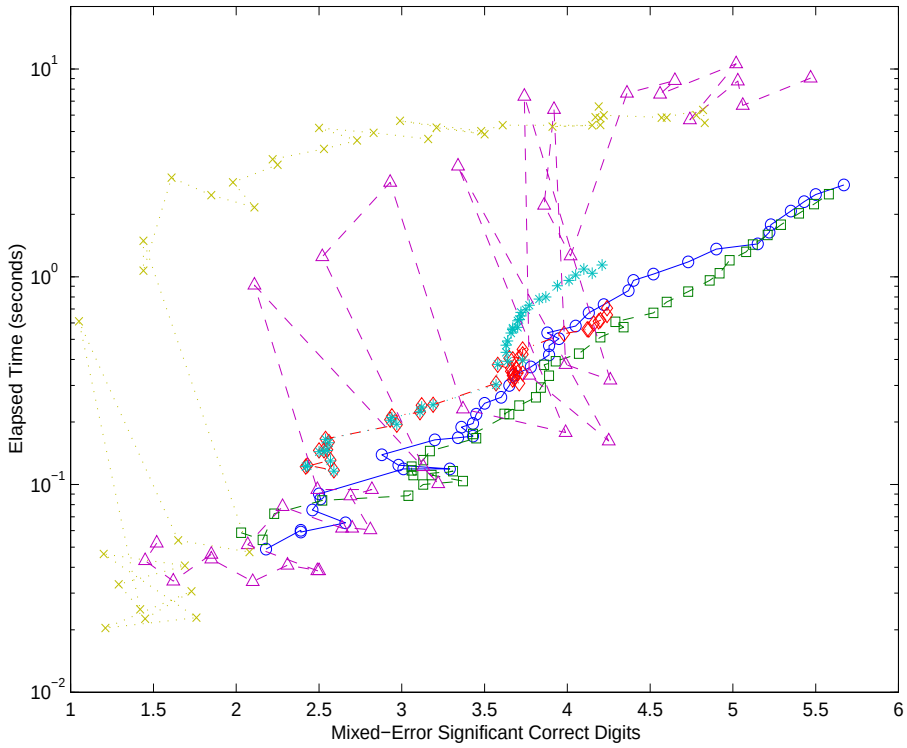


Figure 5.1: Work-Precision Diagrams for the elastic Beam problem.

5.2 The Brusselator with 1D diffusion problem

The problem arises from chemical kinetics. Its mathematical formulation is a reaction-diffusion partial differential equation. In particular, the following one-spatial variable formulation of the problem has been considered [59]:

$$\begin{cases} \frac{\partial u}{\partial t} = A + u^2 v - (B+1)u + \alpha \frac{\partial^2 u}{\partial x^2}, \\ \frac{\partial v}{\partial t} = Bu - u^2 v + \alpha \frac{\partial^2 v}{\partial x^2}, \end{cases}$$

with $0 \leq x \leq 1$, $0 \leq t \leq 10$, $A = 1$, $B = 3$, $\alpha = 1/50$ and boundary conditions

$$\begin{array}{rcll} u(0,t) & = & u(1,t) = 1, & v(0,t) = v(1,t) = 3, \\ u(x,0) & = & 1 + \sin(2\pi x), & v(x,0) = 3. \end{array}$$

The equation is transformed into a large stiff system of ODEs by means of the method of lines applied to the diffusion terms. In particular, a grid of 500 points has been considered for the space interval, thus leading to an IVP for a system of 1000 ODEs. By considering the following ordering for the components of the solution

$$y \equiv (\ u_1\ v_1\ u_2\ v_2\ \ldots\ )^T,$$

where u_i and v_i represent the approximations at the i-th spatial grid point, the Jacobian of the resulting system turns out to be banded with upper and lower bandwidth equal to 2.

Table 5.2 and Figure 5.2 present, respectively, the run characteristics and the work-precision diagrams of the numerical experiments on this problem. For the diagrams we used: $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(2+m/4)}$, $m = 0, \ldots, 44$. We observe that since only the components with indexes $7k + 1$, $k = 0, \ldots, 142$ are provided for the reference solution, the reported *scd* values refer only to them. Moreover, the *mescd* values have not been computed since this measure of the precision is of interest only when it refers to all the components of the numerical solution (see [73]).

Table 5.2: Run characteristics for the Brusselator 1D problem ($h_0 = \text{atol} = \text{rtol}$).

Solver	rtol	scd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-5}	6.36	33	32	663	28	33	$2.86 \cdot 10^{-1}$
	10^{-8}	9.64	50	50	1268	38	49	$5.20 \cdot 10^{-1}$
	10^{-11}	12.77	74	73	2501	55	73	$1.11 \cdot 10^{0}$
DASSL	10^{-5}	4.13	133	131	161	18		$1.04 \cdot 10^{-1}$
	10^{-8}	6.79	474	473	550	24		$3.60 \cdot 10^{-1}$
	10^{-11}	9.68	1442	1440	2014	49		$1.16 \cdot 10^{0}$
GAM	10^{-5}	5.41	26	24	847	21	26	$4.08 \cdot 10^{-1}$
	10^{-8}	8.22	37	36	1392	27	36	$7.34 \cdot 10^{-1}$
	10^{-11}	11.41	74	72	3055	58	74	$1.62 \cdot 10^{0}$
MEBDFDAE	10^{-5}	5.83	121	120	182	19	19	$2.79 \cdot 10^{-1}$
	10^{-8}	7.68	263	261	380	31	31	$6.98 \cdot 10^{-1}$
	10^{-11}	11.06	614	614	861	59	59	$1.66 \cdot 10^{0}$
RADAU	10^{-5}	5.54	46	44	320	38	46	$1.85 \cdot 10^{-1}$
	10^{-8}	9.04	43	40	656	29	43	$3.32 \cdot 10^{-1}$
	10^{-11}	11.59	49	46	1169	28	49	$5.60 \cdot 10^{-1}$
RADAU5	10^{-5}	5.54	46	44	320	38	46	$1.81 \cdot 10^{-1}$
	10^{-8}	8.15	124	123	846	92	107	$4.56 \cdot 10^{-1}$
	10^{-11}	10.66	381	381	2637	58	169	$1.18 \cdot 10^{0}$

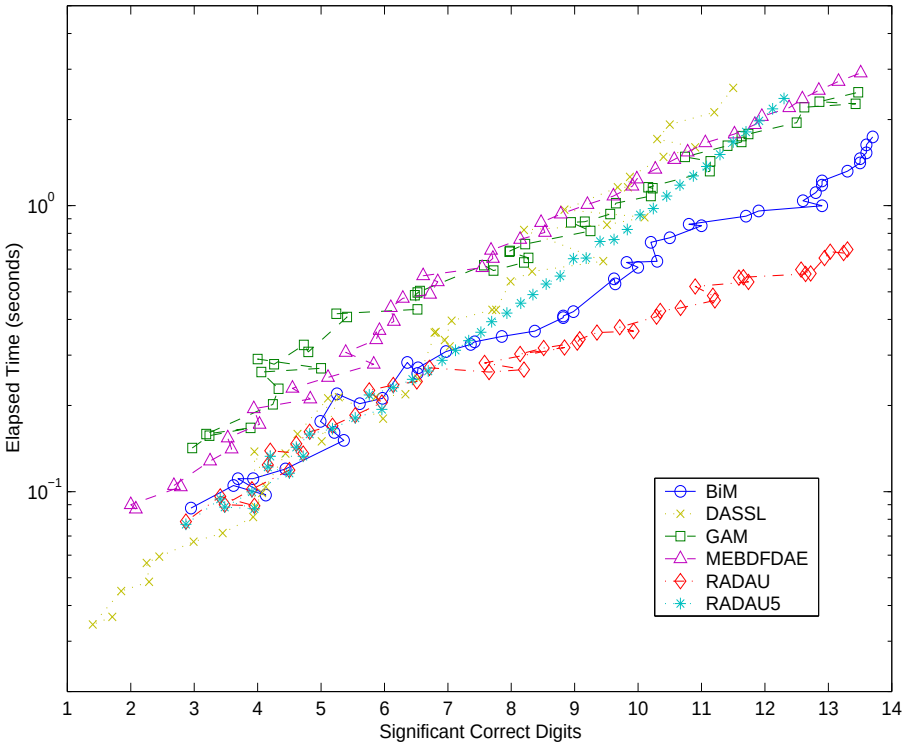


Figure 5.2: Work-Precision Diagrams for the Brusselator 1D problem.

5.3 The Emep problem

The problem is the chemistry part of the EMEP MSC-W ozone chemistry model which is in development at the Norwegian Meteorological Institute of Oslo, [73, 95, 96, 97]. About 140 reactions with a total of 66 species are involved in the model. The time interval $[t_0, T] = [3600 \cdot 4,\ 3600 \cdot (-4+24 \cdot 5)]$ covers 112 hours of simulation (the time is measured in seconds). Moreover, some of the involved species undergo a discontinuity at sunrise and sunset corresponding to to $t = 3600\,(\pm 4 + 24\,i)$ with $i = 1, 2, 3, 4$.

The equation has been solved by subdividing $[t_0, T]$ into 9 adjacent subintervals determined by the previous discontinuities. Table 5.3 and Figure 5.3 contain, respectively, the run characteristics and the work-precision diagrams for the Emep Problem. Since components y_{36} and y_{38} are relatively very small and considered physically unimportant, they are not included in the computation of the *scd* values. For the WPDs we used: `rtol` $= 10^{-(2+m/4)}$, $m = 0, \ldots, 36$; `atol` $= 1$ and $h_0 = 10^{-7}$.

We observe that, even thought the codes `DASSL` and `MEBDFDAE` turn out to be the most efficient ones in solving this problem, the code `BiM` is able to provide very regular results (see in particular the WPD with the *mescd* on the abscissae in Figure 5.3). As a matter of fact, this is not the case for the codes `GAM`, `RADAU` and `RADAU5`.

Table 5.3: Run characteristics for the Emep problem (atol $= 1, h_0 = 10^{-7}$).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-3}	2.13	2.13	368	360	5635	240	364	$3.65 \cdot 10^{-1}$
	10^{-6}	4.63	4.65	727	669	16960	549	713	$1.01 \cdot 10^{0}$
	10^{-9}	7.01	7.48	978	859	29064	647	930	$1.64 \cdot 10^{0}$
DASSL	10^{-3}	2.40	2.40	1149	1093	2171	189		$1.62 \cdot 10^{-1}$
	10^{-6}	4.83	4.83	4145	3965	6981	459		$4.99 \cdot 10^{-1}$
	10^{-9}	7.40	7.68	9022	8770	12811	708		$9.11 \cdot 10^{-1}$
GAM	10^{-3}	3.46	3.46	316	282	11148	210	316	$5.15 \cdot 10^{-1}$
	10^{-6}	5.97	5.98	444	407	22184	324	432	$1.02 \cdot 10^{0}$
	10^{-9}	7.25	7.69	758	648	35939	485	697	$1.65 \cdot 10^{0}$
MEBDFDAE	10^{-3}	2.35	2.35	1020	960	2247	172	172	$1.81 \cdot 10^{-1}$
	10^{-6}	5.18	5.18	2887	2728	5343	441	441	$4.74 \cdot 10^{-1}$
	10^{-9}	7.80	8.24	4962	4731	8107	713	713	$7.74 \cdot 10^{-1}$
RADAU	10^{-3}	2.46	2.46	436	382	3837	277	436	$6.89 \cdot 10^{-1}$
	10^{-6}	3.60	3.62	463	390	10241	281	463	$2.03 \cdot 10^{0}$
	10^{-9}	5.47	5.94	651	547	13929	408	650	$2.81 \cdot 10^{0}$
RADAU5	10^{-3}	2.46	2.46	436	382	3837	277	436	$6.80 \cdot 10^{-1}$
	10^{-6}	4.43	4.45	965	905	8026	760	930	$1.47 \cdot 10^{0}$
	10^{-9}	5.57	6.04	1867	1756	13882	1462	1724	$2.68 \cdot 10^{0}$

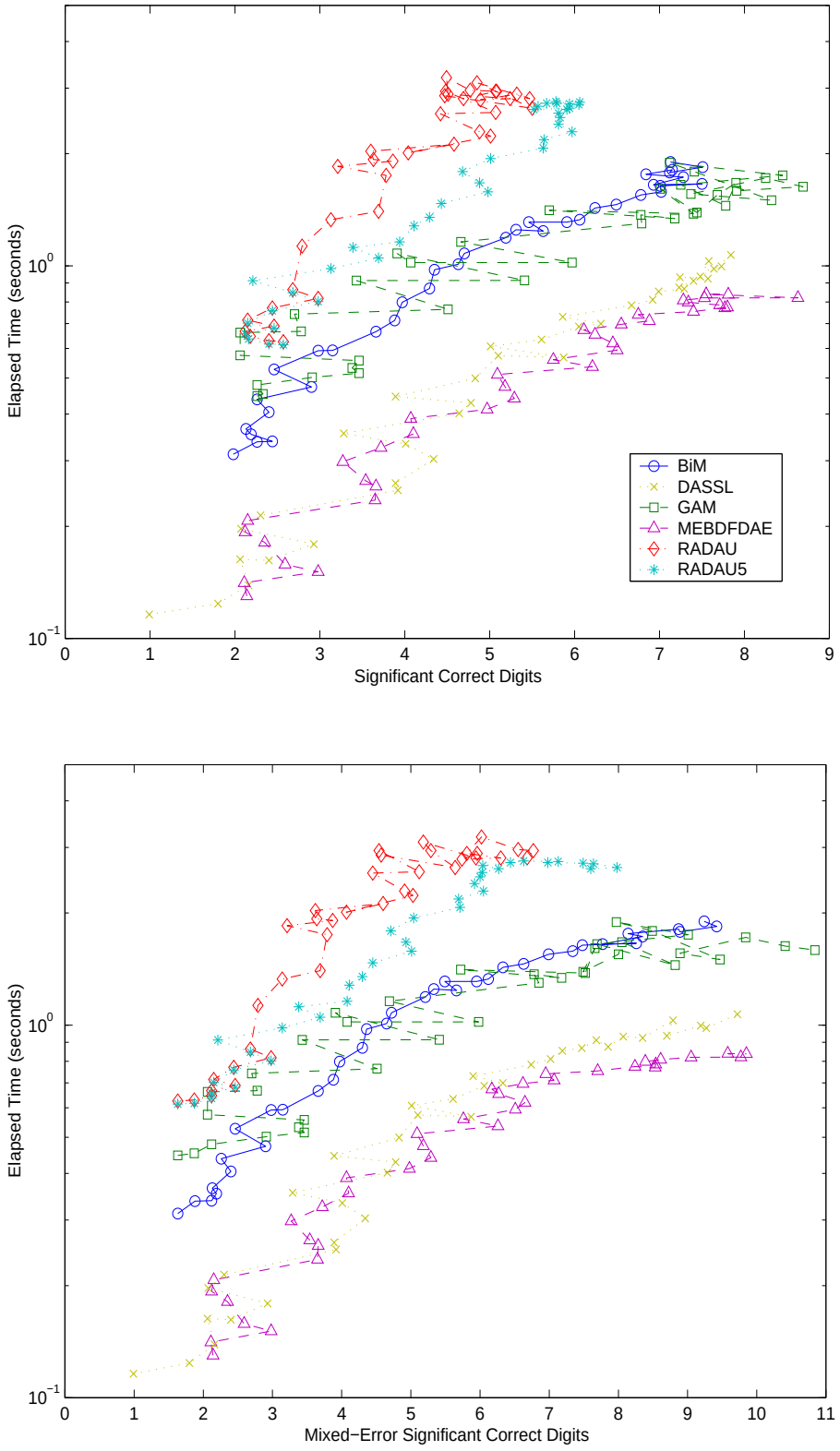


Figure 5.3: Work-Precision Diagrams for the Emep problem.

5.4 The Medical Akzo Nobel problem

The Medical Akzo Nobel research laboratories formulated this problem in the study of the penetration of radio-labeled antibodies into a tissue that has been infected by a tumor, [67]. This study was carried out for diagnostic as well as therapeutic purposes.

The mathematical formulation of the model leads to a reaction diffusion system of size 2 in one spatial dimension, (see [73] for further details). The problem is then transformed into a stiff IVP for a system of $2N$ ODEs by means of the method of lines. The Jacobian of such system is banded with upper and lower bandwidth equal to 2.

Numerical experiments were done in the case $N = 200$. Table 5.4 and Figure 5.4 show the run characteristics and the work-precision diagrams respectively. For the latter ones, we used `atol` = `rtol` = $10^{-(2+m/4)}$, $m = 0, \ldots, 28$, $h_0 = 10^{-5}$ `rtol`.

We remark the competitiveness of the results provided by the code `BiM`. In addition to this, when compared to the other variable-order codes, the WPD corresponding to the code `BiM` turns out to be the most regular.

Table 5.4: Run characteristics for the Medical Akzo Nobel problem (atol = rtol, $h_0 = 10^{-5} \cdot$ rtol).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-3}	3.65	3.66	73	73	972	62	73	$8.78 \cdot 10^{-2}$
	10^{-6}	7.23	7.29	152	152	2967	131	152	$3.07 \cdot 10^{-1}$
	10^{-9}	9.82	9.83	216	216	5998	194	214	$6.36 \cdot 10^{-1}$
DASSL	10^{-3}	2.34	2.35	254	239	395	50		$7.16 \cdot 10^{-2}$
	10^{-6}	4.64	4.71	898	873	1272	86		$2.39 \cdot 10^{-1}$
	10^{-9}	7.61	7.65	2363	2336	2906	120		$5.72 \cdot 10^{-1}$
GAM	10^{-3}	3.89	3.91	61	61	1538	53	61	$1.14 \cdot 10^{-1}$
	10^{-6}	7.17	7.18	99	99	4034	82	99	$3.50 \cdot 10^{-1}$
	10^{-9}	9.54	9.55	138	137	8516	114	138	$7.95 \cdot 10^{-1}$
MEBDFDAE	10^{-3}	3.36	3.44	241	230	420	53	53	$9.00 \cdot 10^{-2}$
	10^{-6}	6.38	6.44	686	667	1005	95	95	$2.57 \cdot 10^{-1}$
	10^{-9}	8.61	8.67	1342	1312	1911	147	147	$5.40 \cdot 10^{-1}$
RADAU	10^{-3}	3.62	3.68	71	70	598	43	71	$5.67 \cdot 10^{-2}$
	10^{-6}	6.59	6.65	85	85	1527	49	85	$2.35 \cdot 10^{-1}$
	10^{-9}	9.11	9.17	142	142	2490	85	141	$3.91 \cdot 10^{-1}$
RADAU5	10^{-3}	3.62	3.68	71	70	598	43	71	$5.49 \cdot 10^{-2}$
	10^{-6}	5.49	5.50	182	182	1370	124	169	$1.29 \cdot 10^{-1}$
	10^{-9}	8.31	8.46	522	522	3384	336	401	$3.23 \cdot 10^{-1}$

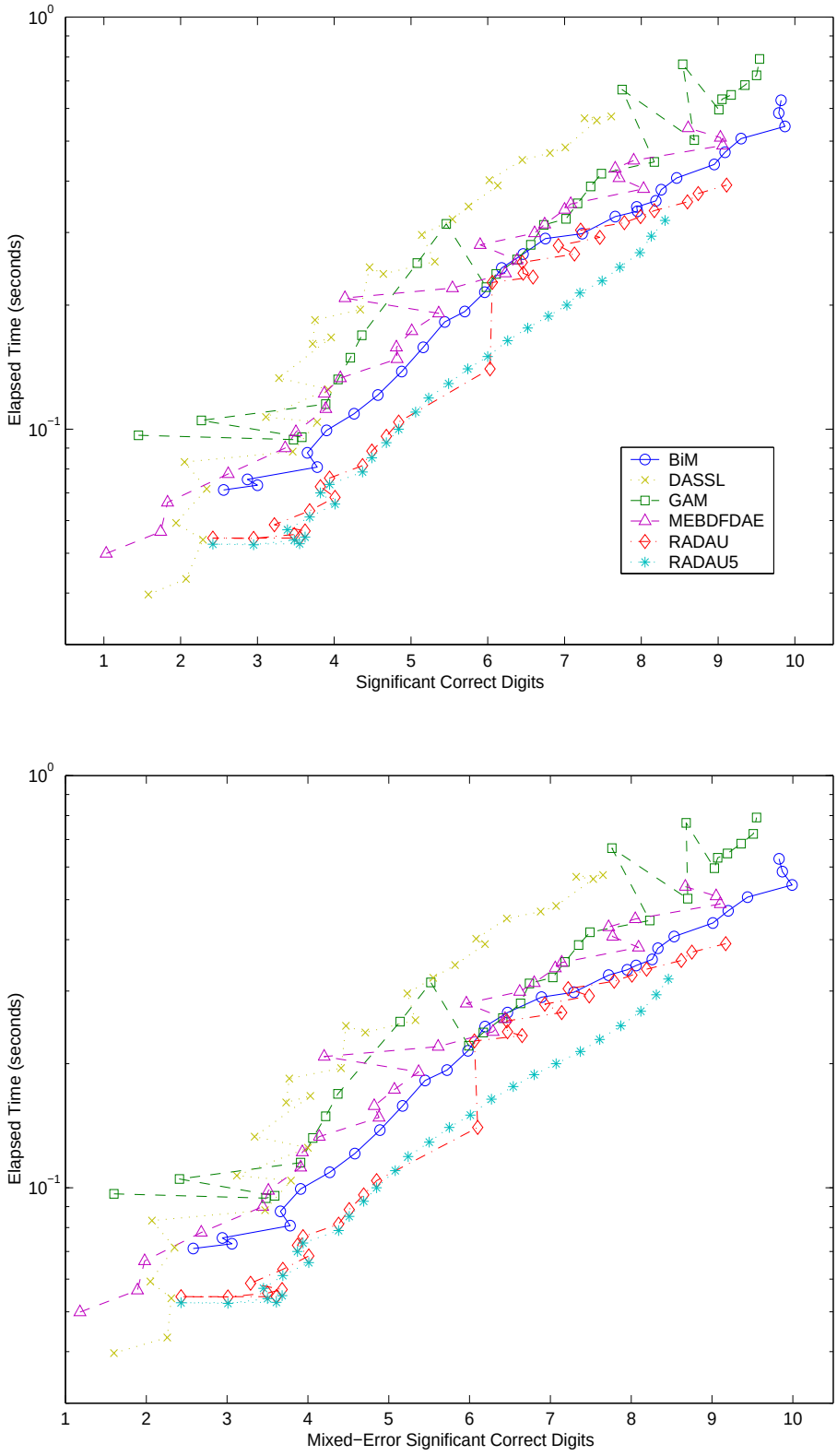


Figure 5.4: Work-Precision Diagrams for the Medical Akzo Nobel problem.

5.5 The Plate problem

The plate problem is a linear non-autonomous problem with constant coefficient matrix arising from the description of the movement of a rectangular plate under the load of a car passing across it, [59]. The mathematical formulation of the problem is:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} + \omega \frac{\partial u}{\partial t} + \sigma \Delta \, \Delta u = f(x,y,t), & (x,y) \in \Omega, \\ u|_{\partial\Omega} = 0, \qquad \Delta u|_{\partial\Omega} = 0, \\ u(x,y,0) = 0, \quad \frac{\partial}{\partial t} u(x,y,0) = 0. \end{cases}$$

The domain $\Omega \equiv [0,2] \times [0,4/3]$, representing the plate, is discretized on a grid of 8×5 interior points thus leading to an IVP for a second-order system of 40 ODEs. This is then transformed into a system of 80 first-order ODEs.

Numerical experiments for this problem were done for $\omega = 1000$, $\sigma = 100$ and integration interval $[t_0, T] = [0, 7]$. Table 5.5 and Figure 5.5 contain, respectively, the run characteristics and the corresponding work-precision diagrams. The input parameters used for the diagrams are the following: $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(2+m/4)}$, $m = 0, \ldots, 44$.

As one can see from the values listed in Table 5.5, the implemented strategy concerning the evaluation of the Jacobian recognize the problem to be linear with a constant coefficient matrix and, consequently, such evaluation is almost always avoided. Moreover, this is a problem for which the order reduction phenomenon occurs and the reported results prove the effectiveness of the corresponding recovery implemented in the code `BiM` (see Section 4.3.1).

Table 5.5: Run characteristics for the Plate problem ($h_0 = \text{atol} = \text{rtol}$).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-5}	5.41	7.41	21	20	522	3	19	$3.99 \cdot 10^{-2}$
	10^{-8}	7.40	9.51	38	37	1315	2	31	$9.37 \cdot 10^{-2}$
	10^{-11}	10.12	12.19	61	60	2728	2	49	$1.90 \cdot 10^{-1}$
DASSL	10^{-5}	2.81	4.95	115	112	181	15		$1.76 \cdot 10^{-2}$
	10^{-8}	5.62	7.98	524	520	710	26		$5.78 \cdot 10^{-2}$
	10^{-11}	8.14	10.08	3424	3413	4877	44		$3.32 \cdot 10^{-1}$
GAM	10^{-5}	3.50	5.64	22	20	655	17	22	$4.17 \cdot 10^{-2}$
	10^{-8}	6.26	8.40	38	35	1561	29	38	$9.62 \cdot 10^{-2}$
	10^{-11}	9.27	11.41	68	66	3641	59	68	$2.09 \cdot 10^{-1}$
MEBDFDAE	10^{-5}	3.35	5.29	96	91	152	9	9	$1.77 \cdot 10^{-2}$
	10^{-8}	7.14	9.08	206	202	299	23	23	$3.94 \cdot 10^{-2}$
	10^{-11}	10.22	12.16	445	442	636	35	35	$7.80 \cdot 10^{-2}$
RADAU	10^{-5}	3.18	5.43	21	19	107	3	18	$6.07 \cdot 10^{-2}$
	10^{-8}	4.42	6.56	30	29	181	2	25	$9.47 \cdot 10^{-2}$
	10^{-11}	6.81	8.91	47	44	341	4	37	$1.51 \cdot 10^{-1}$
RADAU5	10^{-5}	3.20	5.34	27	25	117	3	21	$6.09 \cdot 10^{-2}$
	10^{-8}	5.07	7.18	87	85	394	3	32	$1.15 \cdot 10^{-1}$
	10^{-11}	6.46	8.50	292	289	1438	4	75	$3.34 \cdot 10^{-1}$

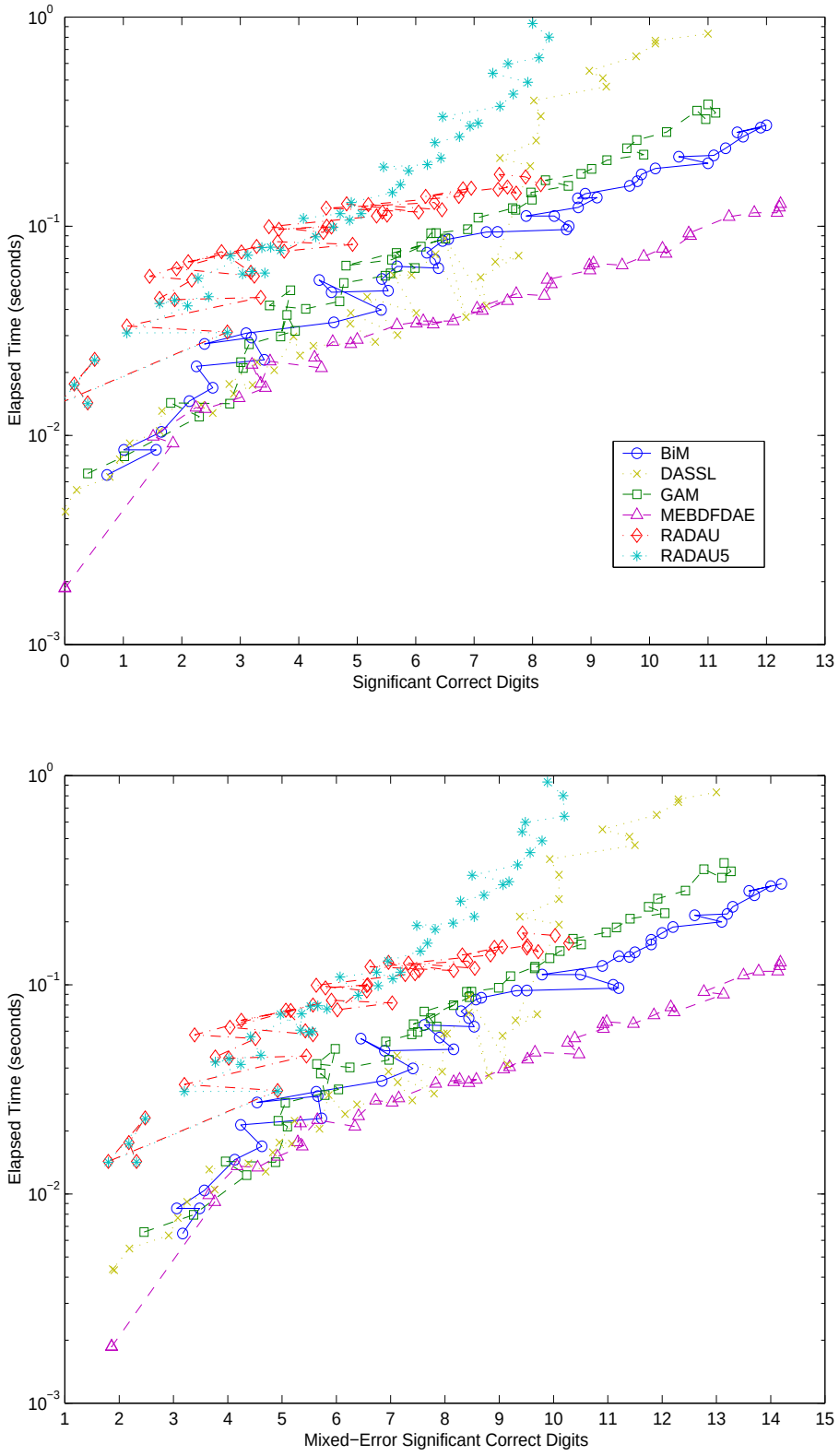


Figure 5.5: Work-Precision Diagrams for the Plate problem.

5.6 The Pollution problem

The problem is a chemical model consisting of 25 reactions and 20 reacting compounds. It represents the chemical reaction part of the air pollution model developed at The Dutch National Institute of Public Health and Environmental Protection (RIVM) and it is described by Verwer in [103].

The mathematical formulation of such model produce a stiff IVP for a system of 20 nonlinear ODEs, [73]. The time interval $[0, 60]$ is sufficient to adequately represent the behaviour of the reactants.

Numerical experiments for this problem have been done with the following set of input parameters $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(2+m/2)}$, $m = 0, \ldots, 22$. The codes RADAU and RADAU5 fail to solve the problem for $m = 0$ since the used stepsize became too small. The run characteristics and the work-precision diagrams are reported in Table 5.6 and Figure 5.6 respectively.

Table 5.6: Run characteristics for the Pollution problem ($h_0 = \text{atol} = \text{rtol}$).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-4}	4.49	6.25	14	14	198	14	14	$1.35 \cdot 10^{-3}$
	10^{-7}	5.81	9.24	24	24	571	21	24	$3.82 \cdot 10^{-3}$
	10^{-10}	9.32	12.53	43	43	1241	29	43	$8.25 \cdot 10^{-3}$
DASSL	10^{-4}	1.96	3.89	35	34	55	13		$8.43 \cdot 10^{-4}$
	10^{-7}	4.13	5.94	135	135	190	22		$2.67 \cdot 10^{-3}$
	10^{-10}	5.93	9.92	381	378	497	37		$6.81 \cdot 10^{-3}$
GAM	10^{-4}	3.53	5.58	13	12	284	9	13	$1.49 \cdot 10^{-3}$
	10^{-7}	6.64	8.70	25	24	743	15	24	$4.11 \cdot 10^{-3}$
	10^{-10}	5.79	12.91	36	36	1463	26	36	$8.37 \cdot 10^{-3}$
MEBDFDAE	10^{-4}	3.15	5.18	37	37	57	10	10	$8.73 \cdot 10^{-4}$
	10^{-7}	4.74	6.72	123	123	184	19	19	$2.72 \cdot 10^{-3}$
	10^{-10}	6.98	10.75	247	247	352	34	34	$5.45 \cdot 10^{-3}$
RADAU	10^{-4}	1.23	3.05	22	18	156	15	21	$1.70 \cdot 10^{-3}$
	10^{-7}	3.78	5.59	32	29	227	21	32	$2.48 \cdot 10^{-3}$
	10^{-10}	7.75	8.77	35	35	449	21	35	$4.09 \cdot 10^{-3}$
RADAU5	10^{-4}	1.23	3.05	22	18	156	15	21	$1.68 \cdot 10^{-3}$
	10^{-7}	3.78	5.59	32	29	227	21	32	$2.44 \cdot 10^{-3}$
	10^{-10}	7.39	8.78	65	65	458	31	46	$4.10 \cdot 10^{-3}$

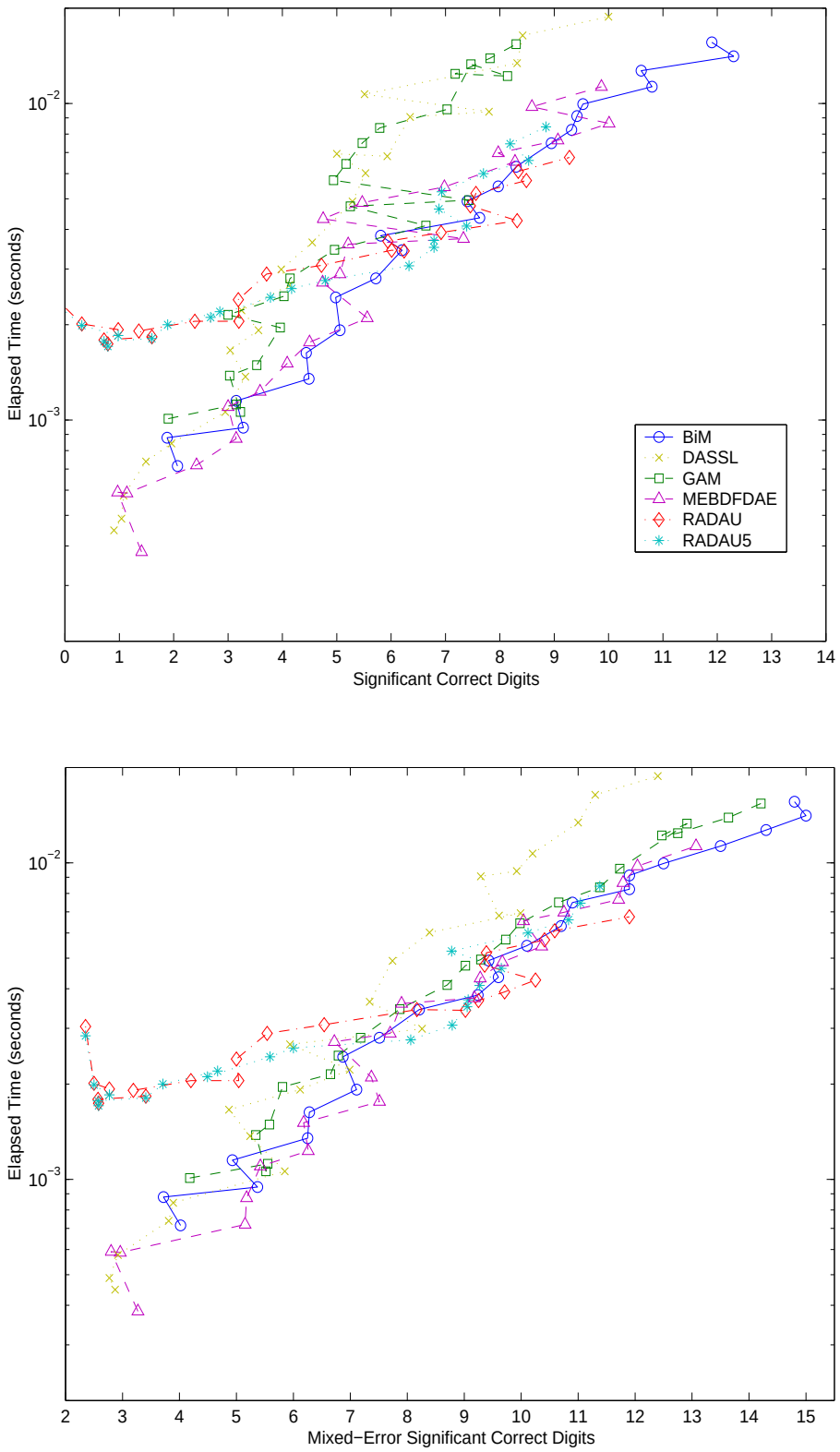


Figure 5.6: Work-Precision Diagrams for the Pollution problem.

5.7 The Ring Modulator problem

The problem originates from electrical circuit analysis and describes the behaviour of the so-called "ring modulator". The latter is an electrical circuit which produce a mixed signal starting from two input signals: one with low-frequency and the second with high frequency, [70, 73]. The application of the Kirchoff Current and Voltage Laws to each closed loop present in the circuit yields an IVP for a system of 15 nonlinear ODEs.

The type and difficulty of the problem depends on the value of the capacity C_s in the circuit. The numerical results here presented refers to $C_s = 2 \times 10^{-12}$ farad, for which the resulting problem is a stiff differential equation. In Table 5.7 and in Figure 5.7 the run characteristics and the work precision diagrams are shown. The input parameters used for the diagrams are the following $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(4+m/4)}$, $m = 0, \ldots, 32$. Failed runs due to overflow occurs when the Radau code is used to solve the problem with input tolerances corresponding to $m = 0 - 11,\ 15 - 17$. We remark that, with respect to the source code available at [73], the control aimed to prevent overflow has been omitted.

Table 5.7: Run characteristics for the Ring Modulator problem ($h_0 = \text{atol} = \text{rtol}$).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-4}	2.22	2.91	18406	17998	420799	16831	18258	$2.41 \cdot 10^{0}$
	10^{-7}	6.17	7.11	25741	25091	816709	25077	25733	$4.56 \cdot 10^{0}$
	10^{-10}	8.83	9.52	29380	28609	1422679	28591	29376	$8.13 \cdot 10^{0}$
DASSL	10^{-4}	0.46	1.15	85466	82972	115884	3510		$1.26 \cdot 10^{0}$
	10^{-7}	2.52	3.21	248615	244982	322234	7720		$3.62 \cdot 10^{0}$
	10^{-10}	4.93	5.62	749570	743521	1071129	17106		$1.11 \cdot 10^{1}$
GAM	10^{-4}	1.73	2.41	13482	11731	475787	11532	13468	$2.41 \cdot 10^{0}$
	10^{-7}	5.32	6.01	19443	18041	914241	17194	19310	$4.74 \cdot 10^{0}$
	10^{-10}	7.96	8.65	34488	33218	1763773	30011	33581	$9.09 \cdot 10^{0}$
MEBDFDAE	10^{-4}	1.78	2.46	65732	65404	99268	6419	6419	$1.17 \cdot 10^{0}$
	10^{-7}	4.64	5.33	155991	155293	217989	13796	13796	$2.67 \cdot 10^{0}$
	10^{-10}	7.28	7.97	348393	347390	464821	25611	25611	$5.82 \cdot 10^{0}$
RADAU	10^{-10}	7.83	8.52	19617	16807	454097	7572	17076	$2.90 \cdot 10^{0}$
RADAU5	10^{-4}	1.45	2.14	36373	28683	176940	8923	32269	$1.49 \cdot 10^{0}$
	10^{-7}	3.81	4.49	102504	93116	545239	12302	54807	$3.58 \cdot 10^{0}$
	10^{-10}	6.12	6.81	288746	279396	1704967	13033	142688	$1.04 \cdot 10^{1}$

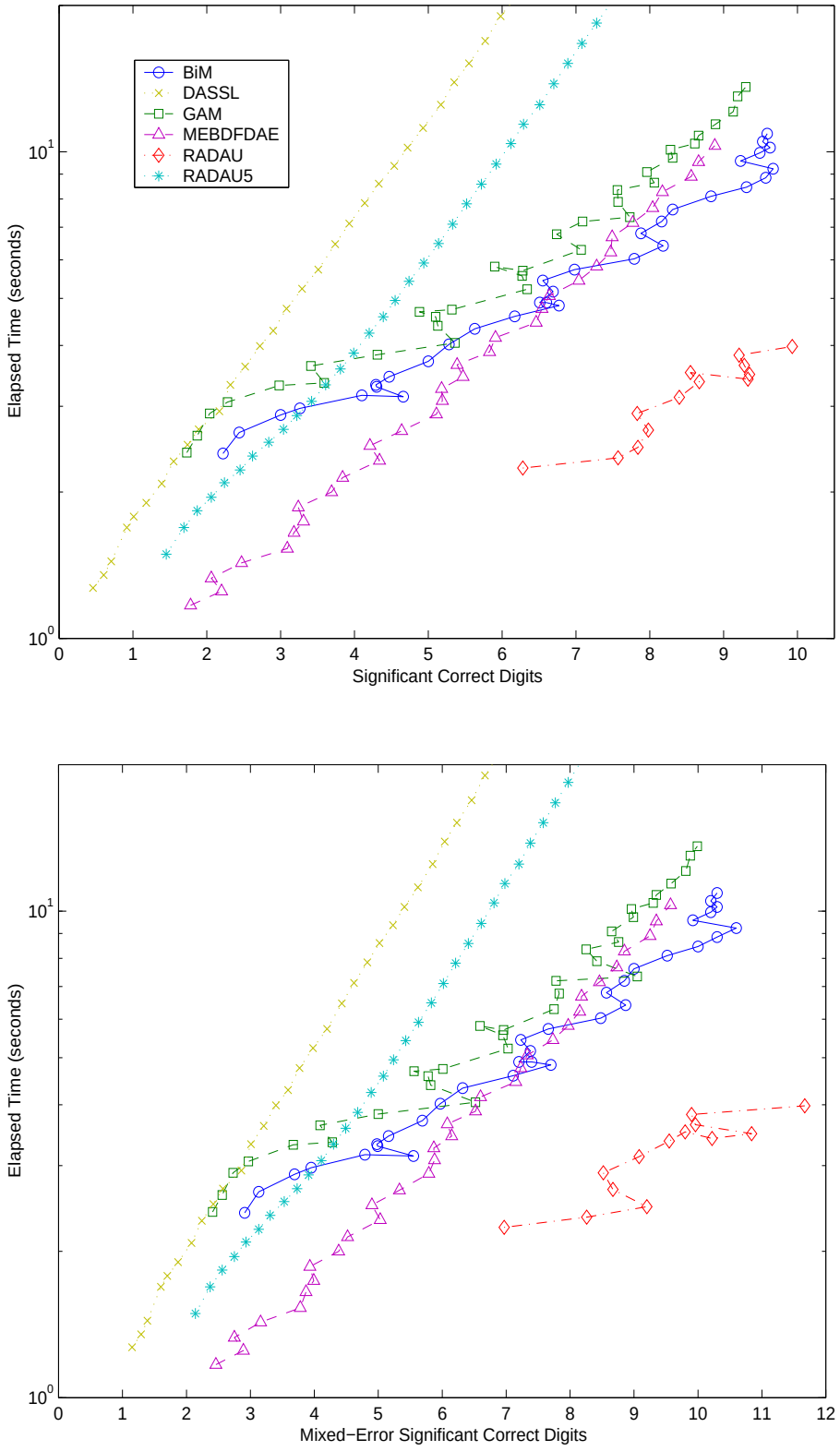


Figure 5.7: Work-Precision Diagrams for the Ring Modulator problem.

5.8 The Robertson problem

The problem describes the kinetics of an autocatalytic reaction given in 1966 by Robertson, [90]. The model involves three chemical species and the corresponding mathematical formulation is:

$$\begin{cases} y_1' = -0.04\, y_1 \;+\; 10^4\, y_2\, y_3, \\ y_2' = \;\; 0.04\, y_1 \;-\; 10^4\, y_2\, y_3 \;-\; 3\cdot 10^7 y_2^2, \\ y_3' = \qquad\qquad\qquad\quad 3\cdot 10^7 y_2^2, \end{cases}$$

where $t \in [0, T]$ and initial value $y_0 = (1, 0, 0)^T$.

Numerical experiments for this problem have been done for $T = 4 \cdot 10^6$. Table 5.9 and Figure 5.8 show respectively the run characteristics and the corresponding work-precision diagrams. For the diagrams we used $h_0 =$ `atol` $=$ `rtol` $= 10^{-(2+m/4)}$, $m = 0, \ldots, 44$. In Table 5.8, we list the failed runs occurred during the experiments.

We observe that, when high accuracy is required for the numerical solution, the codes `BiM` and `RADAU` are the most efficient ones.

Table 5.8: Failed runs for the Robertson problem.

Solver	m	reason
DASSL	1,2	error test failed repeatedly
MEBDFDAE	3,4,5	$h_{\min}$ reduced by a factor of 10^{10}
RADAU and RADAU5	0-8	stepsize too small

Table 5.9: Run characteristics for the Robertson problem ($h_0 =$ atol $=$ rtol).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-5}	5.50	8.79	59	59	1038	59	59	$7.28 \cdot 10^{-4}$
	10^{-8}	8.28	11.57	58	57	2213	53	58	$1.53 \cdot 10^{-3}$
	10^{-11}	11.39	14.48	93	92	3960	86	93	$2.77 \cdot 10^{-3}$
DASSL	10^{-5}	2.13	5.99	226	219	341	40		$8.72 \cdot 10^{-4}$
	10^{-8}	4.56	8.49	776	752	1116	75		$2.86 \cdot 10^{-3}$
	10^{-11}	7.29	10.94	1855	1817	2526	113		$6.34 \cdot 10^{-3}$
GAM	10^{-5}	4.92	8.21	51	43	1726	39	49	$1.09 \cdot 10^{-3}$
	10^{-8}	6.66	10.36	55	55	2989	45	55	$2.03 \cdot 10^{-3}$
	10^{-11}	9.65	13.03	101	101	5719	89	101	$3.89 \cdot 10^{-3}$
MEBDFDAE	10^{-5}	4.11	7.40	213	212	305	39	39	$6.68 \cdot 10^{-4}$
	10^{-8}	7.35	10.65	500	496	747	63	63	$1.62 \cdot 10^{-3}$
	10^{-11}	9.37	12.66	991	988	1446	114	114	$3.19 \cdot 10^{-3}$
RADAU	10^{-5}	3.93	7.22	61	59	488	56	61	$4.57 \cdot 10^{-4}$
	10^{-8}	6.83	10.12	147	145	1057	139	147	$9.88 \cdot 10^{-4}$
	10^{-11}	8.88	12.16	104	103	1952	91	104	$1.42 \cdot 10^{-3}$
RADAU5	10^{-5}	3.93	7.22	61	59	488	56	61	$3.94 \cdot 10^{-4}$
	10^{-8}	6.83	10.12	147	145	1057	139	147	$8.82 \cdot 10^{-4}$
	10^{-11}	8.49	11.78	416	415	2914	217	229	$2.12 \cdot 10^{-3}$

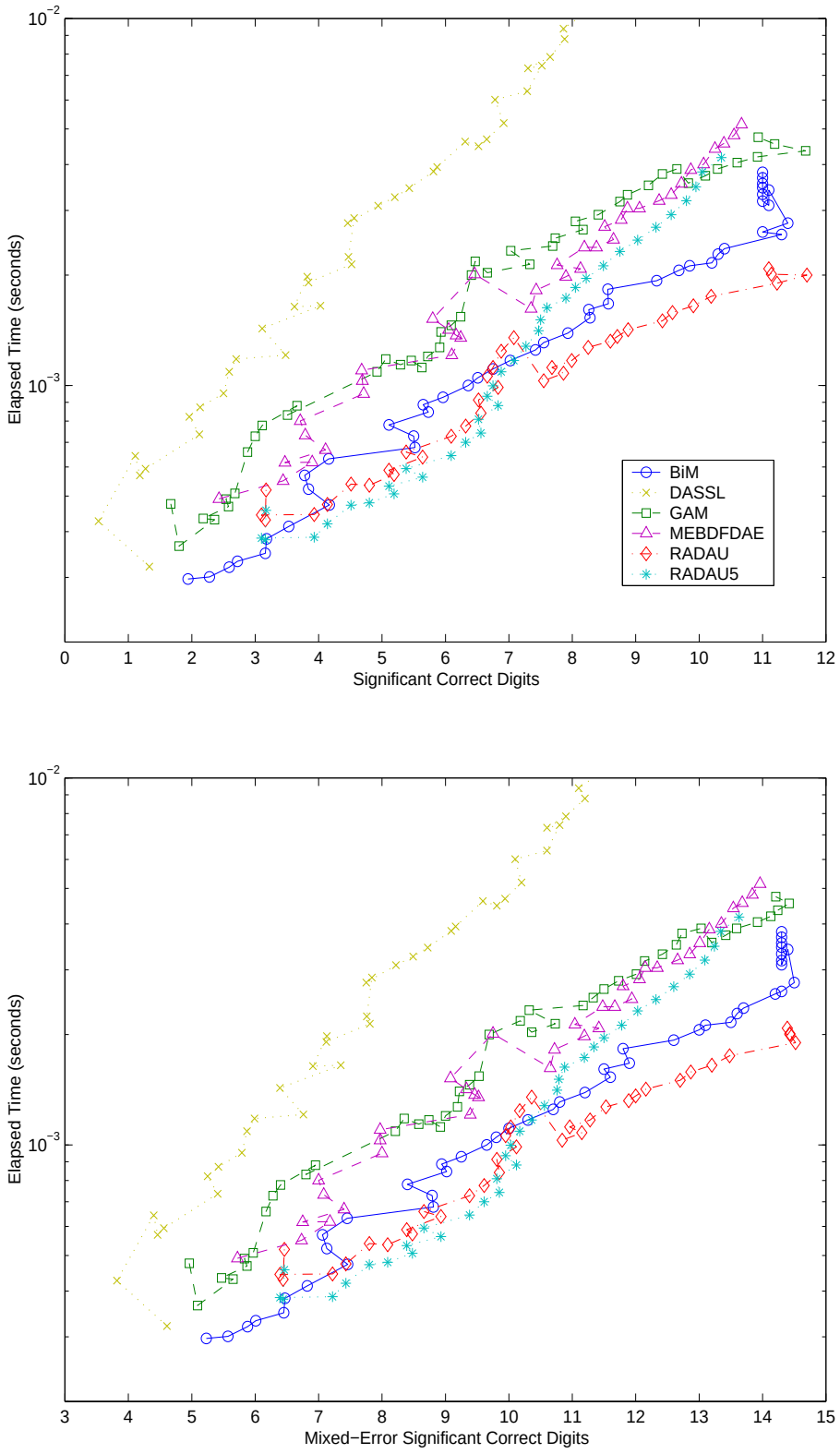


Figure 5.8: Work-Precision Diagrams for the Robertson problem.

5.9 The van der Pol problem

The van der Pol problem originates from electronics and describes the behaviour of a nonlinear vacuum tube circuit, [59]. The standard mathematical formulation of the problem is:

$$z'' + \mu(z^2 - 1)z' + z = 0, \qquad \mu > 0.$$

This equation has two periodic solutions: the constant solution $z(t) \equiv 0$, that is unstable, and the nontrivial periodic solution (corresponding to the initial conditions $z(0) = 2$, $z'(0) = 0$), which it is an attractive limit cycle, since all the other nontrivial solutions approach it, as $t \to \infty$.

Numerical experiments on this problem have been done by performing the classical transformation into a first-order system of 2 ODEs, and considering the initial value $(2, 0)^T$, Finally, we consider the value $\mu = 1000$ and the integration interval $[0, \mu]$. In Table 5.10 and Figure 5.9, the run characteristics and the work-precision diagrams are shown. For the latter ones, we used $h_0 = \texttt{atol} = \texttt{rtol} = 10^{-(2+m/4)}$, $m = 0, \ldots, 44$.

Table 5.10: Run characteristics for the van der Pol problem (h_0 = atol = rtol).

Solver	rtol	scd	mescd	steps	accept	f-eval	j-eval	LU-dec	CPU
BIM	10^{-5}	6.15	6.40	79	69	1848	66	79	$9.32 \cdot 10^{-4}$
	10^{-8}	8.97	9.66	123	117	3940	108	123	$1.93 \cdot 10^{-3}$
	10^{-11}	11.96	13.71	157	157	6397	144	157	$3.12 \cdot 10^{-3}$
DASSL	10^{-5}	4.10	4.49	354	335	574	64		$1.06 \cdot 10^{-3}$
	10^{-8}	6.09	6.54	973	959	1537	74		$2.93 \cdot 10^{-3}$
	10^{-11}	8.89	9.34	3275	3251	4861	116		$9.33 \cdot 10^{-3}$
GAM	10^{-5}	6.15	6.34	66	50	2751	42	66	$1.25 \cdot 10^{-3}$
	10^{-8}	7.73	7.94	101	87	5988	62	101	$2.73 \cdot 10^{-3}$
	10^{-11}	10.35	10.75	126	118	7743	63	117	$3.54 \cdot 10^{-3}$
MEBDFDAE	10^{-5}	3.77	4.21	336	313	562	48	48	$8.67 \cdot 10^{-4}$
	10^{-8}	7.07	7.47	668	647	1090	74	74	$1.74 \cdot 10^{-3}$
	10^{-11}	9.99	10.58	1560	1544	2337	160	160	$3.98 \cdot 10^{-3}$
RADAU	10^{-5}	4.33	5.88	127	113	1116	93	125	$7.49 \cdot 10^{-4}$
	10^{-8}	6.47	7.92	137	134	1877	106	133	$1.12 \cdot 10^{-3}$
	10^{-11}	10.95	11.14	143	135	3403	98	138	$1.77 \cdot 10^{-3}$
RADAU5	10^{-5}	5.22	6.04	146	131	1133	93	134	$6.57 \cdot 10^{-4}$
	10^{-8}	7.48	7.92	373	368	2813	181	306	$1.63 \cdot 10^{-3}$
	10^{-11}	9.46	9.95	1147	1146	8394	243	854	$4.70 \cdot 10^{-3}$

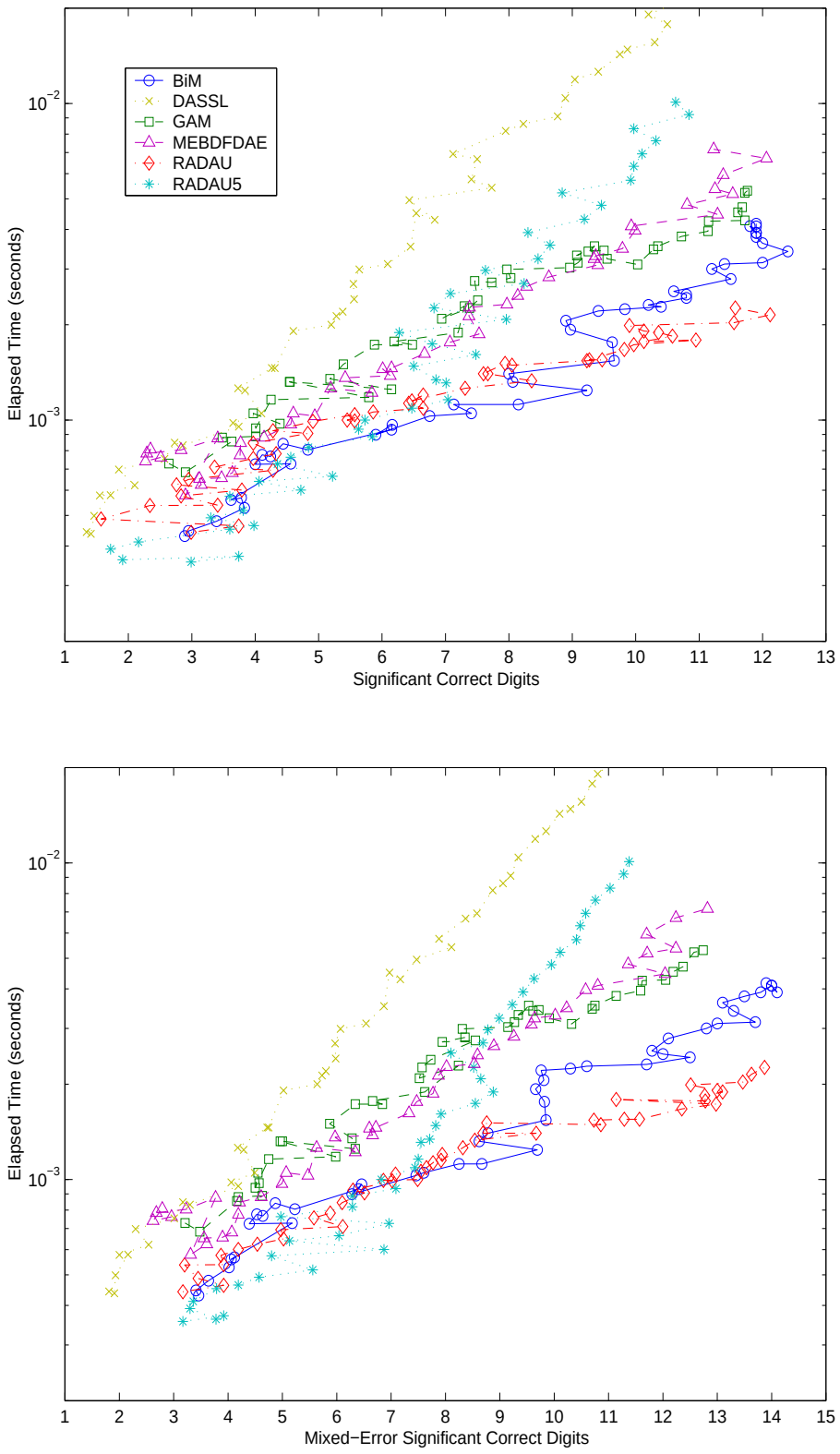


Figure 5.9: Work-Precision Diagrams for the van der Pol problem.

5.10 Final Remarks

The previous results prove that the code `BiM` turns out to be a robust and reliable one. We think that such peculiarities are mainly due to the results obtained with the linear analysis of convergence for the blended iteration which have allowed to construct a code with a very little heuristics inside. Moreover , we think that, in evaluating the usability of a numerical software, the capability of providing regular and robust results has to be taken into full account.

In terms of efficiency, the new code turns out to be competitive with respect to some of the best codes currently available. In particular, we remark that the code `BiM` always well compare with respect to the code `GAM`. This comparison turns out to be of particular interest because of the following considerations:

- both codes make use of a nonlinear splitting for the solution of the discrete problem generated by the implemented block implicit methods;
- the nonlinear iteration used in `BiM` requires the solution of twice linear systems per iteration with respect to the one used in `GAM`, for methods with the same blocksize r.

As a consequence, the obtained results prove the high efficiency of the proposed blended implementation in terms of convergence properties of the corresponding nonlinear iteration.

The code `BiM` is currently available at the WEB site:

`http://www.math.unifi.it/~brugnano/BiM/`

The page contains the Fortran77 source files of the code. Moreover, the results obtained in several numerical experiments, among which the ones here reported, are also available on that page. In addition, for each test problem, a corresponding Fortran77 source code is available . The latter contains the routines for the function and jacobian evaluations, the definition of the initial value and of the integration interval and, finally, the reference solution with respect to which the precision of the numerical solution has been computed.

5.11 Future Research

Several directions for future researches concerning blended implicit methods can be foreseen. Among them, we quote the following ones:

- The research for the implementation on parallel computer of the code `BiM` . As already observed, the diagonal splitting used in the code `BiM` determines a perfect degree of parallelism of the blended iteration, for what concerns the system solvings and the function evaluations. An implementation on parallel computer of such methods seems, therefore, to be promising. Obviously, a necessary requirement for the effectiveness of the parallel code, at least for small/medium size problems, is a "reasonable" balance between the peak performance of the processor elements of the parallel computer and the cost for the interprocessor communications. When large size problems have to be solved, instead, the use of an algorithm for a parallel decomposition is mandatory. The previous considerations refers to a general-purpose IVP parallel solver. However, when the continuous problem is of large size and has a sparse Jacobian matrix (as it happens, for example, for the ODEs arising from the application of the method of lines to reaction-diffusion PDEs in more than 1 dimension), the use of iterative methods for linear systems, in place of direct ones, may be more convenient. In solving this kind of problems, a parallel version of the code `BiM` seems to have great potentialities;

- The extension of the code `BiM` for the solution of linearly implicit DAEs,
$$M\,y'(t) = f(t, y), \tag{5.1}$$
with constant mass matrix M and index lower or equal to 3, is a further important argument of future research. In this context, the choice of the weight function θ in (4.4) has to be adapted. Then, a linear analysis of convergence of the obtained iteration is required. Moreover, the problem of the local error estimates needed for the variation of both the stepsize and the order of the method has to be investigated;

- Finally, the search for different Blended Implicit Methods, with respect to the ones implemented in the code `BiM`, represents an interesting subject of further investigation. As an example, the use of basic block implicit methods with non uniformly distributed internal abscissae may result in an improvement of the conditioning of the coefficient matrix C of the method and, consequently, of the discrete problem.

Bibliography

[1] L. Aceto, D. Trigiante. The Matrices of Pascal and other Greats, *Amer. Math. Monthly* **108** (2001) 232-245.

[2] J. C. Adams, F.Bashforth. *An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops*, Cambridge University Press, Cambridge, 1883.

[3] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff ODEs, *SIAM J. Numer. Anal.*, **14** (1977) 1006-1021.

[4] P. Amodio, L. Brugnano. A note on the efficient implementation of implicit methods for ODEs, *J. Comput. Appl. Math.,* **87** (1997) 1-9.

[5] U. M. Asher, L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, 1998.

[6] O. Axelsson. A class of A-stable methods, *BIT*, **9** (1969), 185-199.

[7] O. Axelsson. A note on a class of strongly A-stable methods, *BIT*, **12** (1972), 1-4.

[8] A. Bellen, M. Zennaro. *Numerical Methods for Delay Differential Equations*, Oxford Science Publication, 2003.

[9] C. Bendtsen. Highly stable parallel Runge-Kutta methods, *Appl. Numer. Math.,* **21** (1996) 1–8.

[10] C. Bendtsen. A parallel stiff ODE solver based on MIRKs, *Advances in Comput. Math.,* **7** (1997) 27–36.

[11] K. E. Brenan, S. L. Campbell, L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations.* Classics in Applied Mathematics, vol. 14, SIAM, Philadelphia, 1996. Code available at:
`http://www.netlib.org/ode/ddassl.f`

[12] P. N. Brown, G. D. Byrne, A. C. Hindmarsh. VODE: a variable-coefficient ODE solver, *SIAM J. Sci. Stat. Comput.*, **10** (1989), 1038-1051.

[13] L. Brugnano. Boundary Value Methods for the Numerical Approximation of Ordinary Differential Equations, *Lecture Notes in Comput. Sci.,* **1196** (1997) 78–89.

[14] L. Brugnano. Blended block BVMs (B_3VMs): A family of economical implicit methods for ODEs, *J. Comput. Appl. Math.,* **116** (2000) 41-62.

[15] L. Brugnano, C. Magherini. Blended implementation of Block Implicit Methods for ODEs, *Appl. Numer. Math.,* **42** (2002) 29-45.

[16] L. Brugnano, C. Magherini. The BiM code for the numerical solution of ODEs, *J. Comput. Appl. Math.,* **164-165** (2004) 145-158. Code available at: `http://www.math.unifi.it/~brugnano/BiM/`

[17] L. Brugnano, C. Magherini. Some linear algebra issues concerning the implementation of Blended Implicit Methods, *Numer. Lin. Alg. Appl.,* (in press).

[18] L. Brugnano, C. Magherini. Economical Error Estimates for Block Implicit Methods for ODEs via Deferred Correction, (submitted).

[19] L. Brugnano, D. Trigiante. On the characterization od stiffness for ODEs. *Dynamics of Continuous, Discrete and Impulsive Systems*, **2** (1996) 317-335.

[20] L. Brugnano, D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Taylor & Francis, London, 1998.

[21] L. Brugnano, D. Trigiante. Block implicit methods for ODEs, in: D.Trigiante (Ed.), Recent trends in Numerical Analysis, Nova Science Publ. Inc., New York, 2001, pp. 81-105.

[22] K. Burrage. A special family of Runge-Kutta methods for solving stiff differential equations, *BIT,* **18** (1978) 22-41.

[23] K. Burrage. High order algebraically stable Runge-Kutta methods, *BIT,* **18** (1978) 373-383.

[24] K. Burrage. Order properties of implicit multivalue methods, *IMA J. Numer. Anal.,* **8** (1988), 385-400.

[25] K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations*, Clarendon Press, Oxford, 1995.

[26] P. M. Burrage. *Runge-Kutta methods for Stochastic Differential Equations*, PhD Thesis, Department of Mathematics, University of Queensland, Australia, 1999.

[27] J. C. Butcher. Coefficients for the study of Runge-Kutta integration processes, *J. Austral. Math. Soc.,* **3** (1963), 185-201.

[28] J. C. Butcher. Implicit Runge-Kutta processes, *Math. Comp.,* **18** (1964), 50-64.

[29] J. C. Butcher. Integration processes based on Radau quadrature formulae. *Math. Comp.*, **18** (1964), 233-244.

[30] J. C. Butcher. On the attainable order of Runge-Kutta methods, *Math. Comp.,* **19** (1965), 408-417.

[31] J. C. Butcher. A modified multistep method for the numerical integration of ordinary differential equations, *J. Assoc. Comput. Mach.*, **12** (1965), 124-135.

[32] J. C. Butcher. On the convergence of numerical solutions to ordinary differential equations, *Math.Comp.,* **20** (1966), 1-10.

[33] J. C. Butcher. On the implementation of implicit Runge-Kutta methods, *BIT,* **16** (1976), 237-240.

[34] J. C. Butcher. The non-existence of ten stage eighth order explicit Runge-Kutta methods, *BIT,* **25** (1985), 521-540.

[35] J. C. Butcher. *The numerical analysis of ordinary differential equations*, John Wiley, Chichester, 1987.

[36] G. D. Byrne, R. J. Lambert. Pseudo Runge-Kutta methods involving two-points. *J. Assoc. Comp. Mach.,* **13** (1966), 114-123.

[37] J. R. Cash. On the integration of stiff systems of ODEs using extended backward differentiation formulae, *Numer. Math.,* **34,2** (1980), 235-246.

[38] J. R. Cash. The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae, *Comput. Math. Appl.,* **9,5** (1983), 645-657.

[39] J. R. Cash. Iterated Deferred Correction Algorithms for Two-Point BVPs, *WSSIA* **2** (1993) 113–125.

[40] J.R. Cash, S. Considine. An MEBDF code for stiff Initial Value Problems, *ACM Trans. Math. Software* **18,2** (1992) 142-158. Code available at: `http://www.ma.ic.ac.uk/~jcash/IVP_software/readme.html`

[41] J. R. Cash, M. H. Wright. A Deferred Correction Algorithm for Nonlinear Two-Point Boundary Value Problems: Implementation and Numerical Evaluation, *SIAM J. Sci. Statist. Comput.* **12** (1991) 971–989.

[42] F. H. Chipman. A-stable Runge-Kutta processes. *BIT,* **11**, (1971) 384-388.

[43] C. F. Curtiss, J. O. Hirshfelder. Integration of stiff equations, *Proc. Nat. Acad. Sci.,* **38** (1952), 235-243.

[44] G. Dahlquist. Convergence and stability in the numerical integration of ordinary differential equations, *Math. Scand.,* **4** (1956), 33-53.

[45] G. Dahlquist. *Stability and error bounds in the numerical integration of ordinary differential equations*, Trans. Roy. Inst. Technol., Stockholm, Sweden, Nr. 130, 1959.

[46] G. Dahlquist. A special stability problem for linear multistep methods, *BIT,* **3** (1963), 27-43.

[47] K. Dekker, J. G. Verwer. *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North-Holland, Amsterdam-New-York-Oxford, 1984.

[48] B. L. Ehle. High order A-stable methods for the numerical solution of systems of DEs. *BIT,* **8** (1968) 276-278.

[49] B. L. Ehle. *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems.* Research Report CSRR 2010, Dept. AACS, Univ. of Waterloo, Ontario, Canada, 1969.

[50] B. L. Ehle. A-stable methods and Padé approximation to the exponential, *SIAM J. Math. Anal.,* **4** (1973) 671-680.

[51] W. H. Enright. Second derivative multistep methods for stiff ordinary differential equations, *SIAM J. Numer. Anal.,* **11** (1974) 127-136.

[52] I. Galligani. Splitting methods for solving large systems of linear ordinary differential equations on a vector computer, *WSSIAA*, **2** (1993) 165-176.

[53] I. Galligani, V. Ruggiero. Solving large systems of linear ordinary differential equations on a vector computer, *Parallel Comput.*, **9** (1989) 359-365.

[54] C. W. Gear. Hybrid methods for initial value problems in ordinary differential equations. *SIAM J. Numer. Anal.,* **2** (1965), 69-86.

[55] S. Gill. A process for the step-by-step integration of differential equations in an automatic digital computing machine, *Proc. Cambridge Philos. Soc*, **47** (1951) 96-108.

[56] W. B. Gragg, H. J. Stetter. Generalized multistep predictor-corrector methods, *J. Assoc. Comp. Mach.*, **11** (1964), 188-209.

[57] E. Hairer, S. P. Nørsett, G. Wanner. Order stars and stability theorems, *BIT,* **18** (1978) 475-489.

[58] E. Hairer, S. P .Nørsett, G. Wanner. *Solving ordinary differential equations I: Nonstiff Problems,* 2^{nd} *ed.*, Springer Series in Computat. Mathematics, vol.8, Springer Verlag, Berlin, 1993.

[59] E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems,* 2^{nd} *ed.*, Springer Series in Computat. Mathematics, vol. 14, Springer-Verlag, Berlin, 1996. Codes available at:
`http://www.unige.ch/math/folks/hairer/software.html`

[60] E. Hairer, G. Wanner. Stiff differential equations solved by Radau methods, *J. Comp. Appl. Math.,* **111** (1999) 93-111.

[61] E. Hairer, C. Lubich, G. Wanner. *Geometric Numerical Integration: structure-preserving algorithms for Ordinary Differential Equations*,Springer Series in Computat. Mathematics, vol. 31, Springer-Verlag, 2002.

[62] `http://www.unige.ch/math/folks/hairer/testset/testset.html`

[63] P. Henrici. *Discrete variable methods in ordinary differential equations*, John Wiley, New York, 1962.

[64] K. Heun. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen, *Z. Math. Phis.,* **45** (1900), 23-38.

[65] A. C. Hindmarsh. *ODEPACK, a systematized collection od ODE solvers,* in Scientific Computing, R. S. Stepleman et al. (Eds.), North Holland, Amsterdam, (1983) 55-64.

[66] W. Hoffmann, J. J. B. de Swart. Approximating Runge-Kutta matrices by trangular matrices, *Preprint NM-R9517*, CWI, Amsterdam, 1995.

[67] R. van der Hout, 1994. Private communication.

[68] P. J. van der Houwen, J. J. B. de Swart. Triangularly implicit iteration methods for ODE-IVP solvers, *SIAM J. Sci. Comput.,* **18** (1997) 41-55.

[69] P. J. van der Houwen, J. J. B. de Swart. Parallel linear system solvers for Runge-Kutta methods, *Adv. Comput. Math.,* **7,1-2** (1997) 157-181.

[70] W. Kampowski, P. Rentrop, W. Schmidt. Classification and numerical simulation of electric circuits. *Surveys on Mathematics for Industry*, 2(1):23-65, 1992.

[71] F. Iavernaro, F. Mazzia. Solving Ordinary Differential Equations by Generalized Adams Methods: Properties and Implementation Techniques. *Appl. Numer. Math.,* **28** (1998) 107–126. Code GAM available at: `http://www.dm.uniba.it/mazzia/ode/readme.html`

[72] F. Iavernaro, F. Mazzia. Block-Boundary Value Methods for the solution of Ordinary Differential Equation. *SIAM J. Sci. Comput.,* **21 (1)** (1999) 323-339.

[73] F. Iavernaro, F. Mazzia. *Test set for Initial Value Problem solvers*, 2002. Available at `http://pitagora.dm.uniba.it/~testset/`

[74] E. Isaacson, H. B. Keller. *Analysis of numerical methods,* Wiley, New York, 1966.

[75] W. Kutta. Beitrag zur näherungsweisen integration totaler differentialgleichungen, *Z. Math. Phis.,* **46** (1901) 435-453.

[76] J. D. Lambert. *Numerical methods for ordinary differential systems.* John Wiley & Sons, New York, 1991.

[77] M. Lentini, V. Pereyra. A Variable Order Finite Difference Method for Nonlinear Multipoint Boundary Value Problems, *Math. Comp.* **28** (1974) 981–1024.

[78] B. Lindberg. Error Estimation and Iterative Improvement for Discretization Algorithms, *BIT* **20** (1980) 486–500.

[79] W. M. Lioen, J. J .B. de Swart, W. A. van der Veen. Test set for IVP solvers, Report NM-R96150, CWI, Department of Mathematics, Amsterdam. 1996.

[80] R. H. Merson. An operational method for the study of integration processes, *Proc. of the Symposium on Data Processing*, Weapons Research Establishment, Salisbury, South Australia, 1957.

[81] W. E. Milne. Numerical integration of ordinary differential equations, *Amer. Math. Monthly,* **33** (1926) 455-460.

[82] F. R. Moulton. *New methods in exterior ballistics*, University of Chicago, 1926.

[83] S. P. Nørsett. Semi-explicit Runge-Kutta methods, Technical Report No. 6/74, Dept. of Math., Univ. of Trondheim, Norway, 1974.

[84] S. P. Nørsett. Runge-Kutta methods with a multiple real eigenvalue only, *BIT,* **16** (1976) 388-393.

[85] E. J. Nyström. Über die numerische integration von differentialgleichungen, *Acta Soc. Sci. Fennicae,* **50** (1925) 1-54.

[86] J. M. Ortega, W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables,* Academic Press, New York, 1970.

[87] V. Pereira. On Improving an Approximate Solution of a Functional Equation by Deferred Corrections, *Numer. Math.* **8** (1966) 376–391.

[88] V. Pereira. Iterated Deferred Corrections for Nonlinear Operator Equations, *Numer. Math.* **10** (1967) 316–323.

[89] A. Prothero, A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equation, *Math. Comput.,* **28** (1974) 145-162.

[90] H. H. Robertson. The solution of a set of reaction rate equations, pages 178-182. Academ Press, 1966.

[91] C. Runge. Über die numerische auflösung von differentialgleichungen, *Math. Ann.,* **46** (1895) 167-178.

[92] E. B. Saff, R. S. Varga. On zeros and poles of Padé approximants to e^z, *III, Numer. Math.,* **30** (1978) 241-266.

[93] J. M. Sanz-Serna, M. P. Calvo. *Numerical Hamiltonian Problems*, Chapman&Hall, London, 1994.

[94] L. F. Shampine. *Numerical solution of ordinary differential equations*, Chapman & Hall, New York, 1994.

[95] D. Simpson, Y. Andersson-Skold, and M. E. Jenkin. Updating the chemical scheme for the EMEP MSC-W model: Current status. Report EMEP MSC-W Note 2/93, The Norwegian Meteorological Institute, Oslo, 1993.

[96] D. Simpson. Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989. Model results and comparisons with observations. *Atmospheric Environment*, 27A:921-943, 1993.

[97] D. Simpson, J. G. Verwer. Explicit methods for stiff ODEs from athmospheric chemistry. Report NM-R9409, CWI, Amsterdam, 1994.

[98] R. D. Skeel. A Theoretical Framework for Proving Accuracy Results for Deferred Corrections, *SIAM J. Numer. Anal.* **19** (1981) 171–196.

[99] R. D. Skeel. Thirteen Ways to Estimate Global Error, *Numer. Math.* **48** (1986) 1–20.

[100] R. D. Skeel, A. K. Kong. Blended linear multistep methods, *ACM Trans. Math. Software,* **3** (1977) 326-345.

[101] H. J. Stetter. The Defect Correction Principle and Discretization Methods, *Numer. Math.* **29** (1978) 425–443.

[102] H. J. Stetter. Global Error Estimation in ODE-Solvers, *Lecture Notes in Mathematics* **630** (1978) 245–258.

[103] J. G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM J. Sci. Comput.,* **15**(**5**) (1994) 1243-1259.

[104] H. A. Watts, L. F. Shampine. *A*-stable block one-step methods, *BIT,* **12** (1972) 252-266.

[105] P. Zadunaisky. On the Estimation of Errors Propagated in the Numerical Solution of Ordinary Differential Equations, *Numer. Math.* **27** (1976) 21–39.