CWI Quarterly Volume 11 (1) 1998, pp. 7 – 32

The Potential of Parallel Multi-Value Methods for the Simulation of Large Real-Life Problems

P. Chartier
INRIA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France

Potential advantages and drawbacks of multi-value methods are discussed in detail. This presentation leads in a natural way to the definition and the construction of Diagonally-Implicit Multi-Stage Integration Methods (DIMSIMs). In particular, it is shown that DIMSIMs should be soon efficiently implemented in a parallel environment.

1. Introduction

Ordinary differential equations (ODEs) arise in a wide variety of situations, ranging from chemical kinetics (chemical reaction systems), to mechanics (equations of motions), to electrical circuits and to problems of fluid flows. The length of the interval, the number of equations as well as the sparsity of the problem vary considerably from one field to the other. When either the interval is enormously long, the size of the system enormously large or when the problem has to be solved in real-time, the use of parallel computers may well provide the answer.

However, as noticed by many authors [2, 16, 20], usual numerical methods for the solution of ODEs are inherently sequential, and this observation has prevented until recently the development of parallel methods. In our search for new numerical methods for the initial value problem

$$\left\{ \begin{array}{lcl} y'(x) & = & f(y(x)) \in \mathbb{R}^m, \\ y(x_0) & = & y_0 \in \mathbb{R}^m, \end{array} \right. \tag{1.1}$$

we may distinguish, along the lines of [20, 35], several techniques open to some parallelism :

1. dividing the computational cost of the right-hand side of the problem by using different processors for different components,

2. decoupling the system into independent sub-systems with fewer coordinates via an iterative process : all sub-systems can then be integrated simultaneously provided there is a sufficient number of processors [38],

3. designing new methods with some intrinsic parallelism, such as block Runge-Kutta methods [28, 29], Parallel Diagonally Iterated Runge-Kutta methods (PDIRKs) [21, 22, 31, 34, 36, 37], Multiply Implicit Runge-Kutta methods (MIRKs) [30], General linear methods [18, 19, 32] or DIMSIMs [4, 5, 7, 9, 10, 11, 12, 13, 14, 15].

It is out of the scope of this paper to discuss these various techniques and methods. For a up-to-date review of parallel methods, one may refer to the book on parallel methods for ODEs by BURRAGE [2]. Here we would rather concentrate on a particular technique, usually referred to as "parallelism across the method". More precisely, our aim is to identify the various methods within the class of multi-value methods which have some potential for parallelism. It is the belief of the author that there is much more to gain for *implicit* methods, which are considerably more costly than explicit ones and however necessary for stiff systems and the focus of this paper will reflect this opinion.

In Section 2, we will briefly present multi-value methods and give some ideas of their computational costs and implementation features. DIMSIM methods then appear as a *convenient* class of methods and we will exhibit their distinctive characteristics. Recent results in the construction of DIMSIMs will be reported in Section 3 and some designing options will be examined into some details. Finally, we will address in Section 4 the implementation aspects of a fifth-order implicit DIMSIM and assess its performances on several test problems in Section 5.

2. FROM MULTI-VALUE METHODS TO DIMSIMS

2.1. Multi-value methods

Multi-value methods can be regarded as a natural generalization of both r-step linear multi-step methods and s-stage Runge-Kutta methods. However, they are much more than the *concatenation* of those two classes, and they include a wide range of new schemes.

Formalism A general multi-value method is defined by

- an *exact value function* $z(x,h)$ taking its values in $\mathbb{R}^{r\times m}$ and furnishing the *interpretation* of the method, -each of the r components of $z(x,h)$ represents a m-dimensional function related in some prescribed way to the exact solution $y(x)$-,
- a *starting procedure* $\mathcal{S}$ providing the initial approximation $y^{[0]}$ at x_0 :

$$\begin{array}{rccl} \mathcal{S} & : & \mathbb{R}^m & \rightarrow & \mathbb{R}^{r\times m}, \\ & & y_0 & \rightarrow & y^{[0]} = \mathcal{S}(y_0) = z(x_0,h_0) + \mathcal{O}(h_0^{p+1}), \end{array}$$

– a *set of equations* giving a new approximation $y^{[n+1]}$ to $z(x_{n+1}, h_{n+1})$ from an approximation $y^{[n]}$ to $z(x_n, h_n)$. In their fixed step-size version ($h_n = h$), these relations take the following form

$$\begin{array}{rcl} Y_i^{[n+1]} & = & h\sum_{j=1}^{s} a_{ij} f(Y_j^{[n+1]}) \quad + \quad \sum_{j=1}^{r} u_{ij} y_j^{[n]}, \quad i = 1, \cdots, s, \\ y_i^{[n+1]} & = & h\sum_{j=1}^{s} b_{ij} f(Y_j^{[n+1]}) \quad + \quad \sum_{j=1}^{r} v_{ij} y_j^{[n]}, \quad i = 1, \cdots, r, \end{array}$$

or in an equivalent but more compact form,

$$\begin{array}{rcl} Y^{[n+1]} & = & h(A \otimes I_m) F(Y^{[n+1]}) \quad + \quad (U \otimes I_m) y^{[n]}, \\ y^{[n+1]} & = & h(B \otimes I_m) F(Y^{[n+1]}) \quad + \quad (V \otimes I_m) y^{[n]}, \end{array}$$

where I_m is the m-dimensional identity matrix, $A = (a_{ij})$, $U = (u_{ij})$, $B = (b_{ij})$, $V = (v_{ij})$ and where

$$Y^{[n+1]} = \begin{bmatrix} Y_1^{[n+1]} \\ Y_2^{[n+1]} \\ \vdots \\ Y_s^{[n+1]} \end{bmatrix} \text{ and } F(Y^{[n+1]}) = \begin{bmatrix} f(Y_1^{[n+1]}) \\ f(Y_2^{[n+1]}) \\ \vdots \\ f(Y_s^{[n+1]}) \end{bmatrix}. \tag{2.2}$$

The four matrices A, U, B and V fully characterize the multi-value method and are usually gathered for this reason in a tableau

$$\mathcal{M} = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right]. \tag{2.3}$$

By $\mathcal{M}$ we will denote both the matrix above and the map corresponding to the multi-value method.

The function $z(x, h)$ has a simple form for known methods such as Runge-Kutta or multi-step methods. For a Runge-Kutta method, one has

$$z(x, h) = y(x),$$

while for a linear r-step method

$$z(x, h) = \begin{bmatrix} y(x - (r-1)h) \\ y(x - (r-2)h) \\ \vdots \\ y(x) \end{bmatrix}.$$

In general however, the *exact value* function can be much more complicated, and in order to get a precise idea of its various forms some acquaintance with B-series is required.

2.2. The concept of B-series

B-series is a crucial concept in the definition of multi-value methods and in the analysis of their order. It has been introduced by HAIRER and WANNER [26] and called B-series in honor of Butcher for his determinant contribution to the theory of trees [8]. Let us now briefly recall a few definitions related to trees :

DEFINITION 2.1. *The set of trees T is recursively defined by*

1. *$\emptyset$ belongs to T,*
2. *$t = [t_1, \dots, t_n]$ belongs to T iff $t_1, \dots, t_n$ belong to T.*

$t = [t_1, \dots, t_n]$ is the tree formed by joining the trees t_1, ..., t_n to a common vertex which becomes the root of t.

DEFINITION 2.2. *The functions ρ (order) and α (number of labellings) are recursively defined on T by*

1. *$\rho(\emptyset) = 0$, $\alpha(\emptyset) = 1$.*
2. *If $t = [\underbrace{t_1, \dots, t_1}_{\mu_1}, \underbrace{t_2, \dots, t_2}_{\mu_2}, \dots, \underbrace{t_n, \dots, t_n}_{\mu_n}]$, where $t_1, \dots, t_n$ are all distinct, then*
   $\rho(t) = 1 + \sum_{i=1}^{n} \mu_i \rho(t_i)$, $\alpha(t) = (\rho(t) - 1)! \prod_{i=1}^{n} \frac{1}{\mu_i!} \left(\frac{\alpha(t_i)}{\rho(t_i)!} \right)^{\mu_i}$.

DEFINITION 2.3. *The function $F(t)(y)$ is recursively defined on T by*

1. *$F(\emptyset)(y) = y$,*
2. *$F([t_1, \dots, t_n]) = \frac{\partial^n f}{\partial y^n} \left(F(t_1)(y), \dots, F(t_n)(y) \right)$*

These various definitions are illustrated for all trees of order less than or equal to 4 in Table 1 :

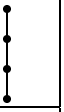
t	$\emptyset$								
$\rho(t)$	0	1	2	3	3	4	4	4	4
$\alpha(t)$	1	1	1	1	1	1	3	1	1
$F(t)$	Id	f	$f_y f$	$f_{yy}(f, f)$	$f_y f_y f$	$f_{yyy}(f, f, f)$	$f_{yy}(f, f_y f)$	$f_y f_{yy}(f, f)$	$f_y f_y f_y f$

TABLE 1. Various functions defined on T

If $a : T \to \mathbb{R}^n$ is an arbitrary map of $(\mathbb{R}^n)^T$, a B-series is the formal series

$$B(a, y) = \sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) \left(a(t) \otimes F(t)(y) \right). \tag{2.4}$$

On the one hand, it can be checked that the exact solution $y(x+h)$ is a B-series with coefficients identically equal to 1, that is to say

$$\begin{aligned} y(x+h) &= \sum_{k=0}^{\infty} \frac{h^k}{k!} y^{(k)}(x), \\ &= \sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) F(t)(y(x)), \end{aligned}$$

where the convergence of the series is ensured for instance for analytical y and sufficiently small h. On the other hand, Butcher has shown that the numerical solution after one step of a Runge-Kutta method starting from $y(x)$ is a B-series whose coefficients can be expressed in terms of the coefficients of the numerical method. Conversely, any B-series may be *interpreted* as the numerical solution after one step of a Runge-Kutta method (with a possibly infinite number of stages). As a consequence, we will only consider *exact value* functions which can be represented by B-series of the form

$$z(x,h) = \sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) \left(\zeta(t) \otimes F(t)(y(x)) \right),$$

where $\zeta(t) = [\zeta_1(t), \zeta_2(t), \ldots, \zeta_r(t)]^T$ takes its values in $\mathbb{R}^r$. This means that it will be possible to obtain *starting procedures* exact up to any arbitrary order p by constructing a set of r Runge-Kutta methods whose B-series coincide with ζ_1, ζ_2, ..., ζ_r up to p^{th}-order terms. For instance, the *exact value* function of an r-steps method has the following expansion

$$z(x,h) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \otimes y(x) + \sum_{t \in T, t \neq \emptyset} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) \begin{bmatrix} (1-r)^{\rho(t)} \\ (2-r)^{\rho(t)} \\ \vdots \\ (-1)^{\rho(t)} \\ 0 \end{bmatrix} \otimes F(t)(y(x)),$$

and a possible *starting procedure* $\mathcal{S}$ for it is given by

$$\begin{bmatrix} \Phi^{1-r} \\ \Phi^{2-r} \\ \vdots \\ Id \end{bmatrix}$$

where Φ is the numerical flow corresponding to any Runge-Kutta method of sufficiently large order.

If a is a map from T to $\mathbb{R}$ such that $a(\emptyset) = 1$ and b from T to $\mathbb{R}^n$, Hairer and Wanner have shown that the *composition* $B(b, B(a,y))$ of the two B-series $B(a,y)$ and $B(b,y)$ is again a B-series of coefficients ab. Although this is

a fundamental result for B-series, it is out of the scope of this introductory section to give a complete description of the composition law. We will rather develop in Table 2 the product $(ab)(t)$ for all trees of order less or equal to 4. This will be enough to get some insight. For a precise definition of ab we refer the reader to [2] or [25].

t	$(ab)(t)$
$\emptyset$	$b(\emptyset)$
•	$a(\bullet)b(\emptyset) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 2a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 3a(\bullet)^2b(\bullet) + 3a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 3a(\bullet)b(\bullet) + 3a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 4a(\bullet)^3b(\bullet) + 6a(\bullet)^2b(\bullet) + 4a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 4a(\bullet)a(\bullet)b(\bullet) + (4a(\bullet)^2 + 2a(\bullet))b(\bullet) + \frac{4}{3}a(\bullet)b(\bullet) + \frac{8}{3}a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 4a(\bullet)b(\bullet) + 6a(\bullet)^2b(\bullet) + 4a(\bullet)b(\bullet) + b(\bullet)$
	$a(\bullet)b(\emptyset) + 4a(\bullet)b(\bullet) + 6a(\bullet)b(\bullet) + 4a(\bullet)b(\bullet) + b(\bullet)$

TABLE 2. Multiplication table

Multi-value methods as one-step methods According to a recent result from STOFFER [33] *strictly stable* multi-value methods are essentially conjugate to one-step methods. By *strictly stable* methods we mean methods such that V has all its eigenvalues within the unit disc except 1, which is assumed to be simple. Note that this condition is fulfilled for most methods of practical interest.

More precisely, Theorem 2.3. of [33] implies that there exists an *exact value* function $z^*(x,h) = B(\zeta^*, y)$, $\zeta^* \in (\mathbb{R}^r)^T$, possibly different from $z(x,h)$, and a one-step method $\Phi(y) = B(\varphi, y)$, $\varphi \in \mathbb{R}^T$ such that the diagram

$$\begin{array}{ccc} \mathbb{R}^{r\times m} & \xrightarrow{\mathcal{M}} & \mathbb{R}^{r\times m} \\ z^* \uparrow & & \uparrow z^* \\ \mathbb{R}^m & \xrightarrow{\Phi} & \mathbb{R}^m \end{array}$$

commutes. Moreover, Φ has the same order as the multi-value method $\mathcal{M}$. In terms of B-series, this is equivalent to

$$\begin{aligned} B(\zeta^*, B(\varphi, y)) &= (B \otimes I_m)B(\Upsilon', y) + (V \otimes I_m)B(\zeta^*, y), \\ B(\Upsilon, y) &= (A \otimes I_m)B(\Upsilon', y) + (U \otimes I_m)B(\zeta^*, y), \\ B(\Upsilon'_i, y) &= hf(B(\Upsilon_i, y)),\ i = 1, \cdots, s. \end{aligned}$$

Translated in terms of trees, this gives

$$\forall\, t \in T,\ (\varphi\zeta^*)(t) = B\Upsilon'(t) + V\zeta^*(t), \tag{2.5}$$

$$\Upsilon(t) = A\Upsilon'(t) + U\zeta^*(t), \tag{2.6}$$

where

$$\Upsilon'(\emptyset) = 0, \tag{2.7}$$

$$\Upsilon'(\bullet) = [1, \cdots, 1]^T := e, \tag{2.8}$$

$$\Upsilon'(t) = \rho(t) \prod_{i=1}^{n} \Upsilon(t_i),\ \text{if } t = [t_1, \cdots, t_n]. \tag{2.9}$$

Now, using the assumption of *strict stability* and relation (2.5) for $t = \emptyset$, V can be written as

$$V = \zeta^*(\emptyset)v^T + \widetilde{V},$$

with $\widetilde{V}\zeta^*(\emptyset) = 0$, $v^T\widetilde{V} = 0^T$ and $v^T\zeta^*(\emptyset) = 1$. Let us now show how the coefficients $\varphi(t)$, $\zeta^*(t)$ and $\Upsilon(t)$ can be constructed recursively under the assumption that $v^T\zeta(t) = \psi(t)$, where ψ is an arbitrary map from T to $\mathbb{R}$. For $t = \emptyset$, relation (2.5) is satisfied by definition of v and $\widetilde{V}$. As for equation (2.6) it comes

$$\Upsilon(\emptyset) = e = U\zeta^*(\emptyset),$$

which is nothing else but the preconsistency condition, ensuring that $f(B(\Upsilon_i, y))$ makes sense for any h. Assume that $\varphi(t)$, $\zeta^*(t)$ and $\Upsilon(t)$ are known for all trees t of order less or equal to $p-1 \geq 0$ and consider a tree t of order p. As it can be noticed from Table 2 and more generally from the definition of the composition law, the product $(\varphi\zeta^*)(t)$ is composed of

$$(\varphi\zeta^*)(t) = \zeta^*(t) + \varphi(t)\zeta^*(\emptyset) + R(t),$$

where $R(t)$ is an expression involving only trees of order less than $\rho(t)$ and greater than 0. Hence, we write equation (2.5) as

$$(I - \widetilde{V})\zeta^*(t) + (\varphi(t) - \psi(t))\zeta^*(\emptyset) = B\Upsilon'(t) - R(t),$$

and it follows from a left multiplication by v^T that

$$\underbrace{v^T(I - \widetilde{V})\zeta^*(t)}_{\psi(t)} + \underbrace{v^T\zeta^*(\emptyset)}_{1}(\varphi(t) - \psi(t)) = v^T\left[B\Upsilon'(t) - R(t)\right],$$

i.e.

$$\varphi(t) = v^T \left[B\Upsilon'(t) - R(t) \right].$$

On the other hand,

$$\begin{aligned} \zeta^*(t) &= (I - \widetilde{V})^{-1} \left[B\Upsilon'(t) - R(t) - (\varphi(t) - \psi(t))\zeta^*(\emptyset) \right], \\ &= (I - \widetilde{V})^{-1} \left[B\Upsilon'(t) - R(t) \right] + (\psi(t) - \varphi(t))\zeta^*(\emptyset) \end{aligned}$$

and it can be checked that $v^T \zeta^*(t) = \psi(t)$. Equation (2.9) now defines $\Upsilon(t)$ and the construction can be continued inductively. Note that apart for $t = \emptyset$ ($\psi(\emptyset) = 1$) $\psi(t)$ can be chosen during the process of construction; ψ should thus be regarded as a degree of freedom.

For instance, φ and ζ^* are given for trees of order less or equal to 2 in Table 3 with the notations $W = (I - \widetilde{V})^{-1}$, $\gamma = v^T Be$ and $c = Ae + U\zeta^*(\bullet)$:

t	$\varphi(t)$	$\zeta^*(t)$	$\Upsilon(t)$
$\emptyset$	1	$\zeta^*(\emptyset)$	e
$\bullet$	γ	$WBe + (\psi(\bullet) - \gamma)\zeta^*(\emptyset)$	c
$\vert$	$2v^T B(A+UWB)e - 2\gamma^2$	$2WBc - 2\gamma W^2 Be + (\psi(\vert) - 2v^T Bc + 2\gamma^2)\zeta^*(\emptyset)$	$2Ac + U\zeta^*(\vert)$

TABLE 3. The *exact value* function and its corresponding one-step method.

Implementation costs For implicit methods, the equation

$$Y^{[n+1]} = h(A \otimes I_m)F(Y^{[n+1]}) + (U \otimes I_m)y^{[n]}$$

is implicit and has to be solved by Newton's method. If J denotes the Jacobian matrix of f at a point close to $y(x_n)$, the *simplified Newton iteration*, which is the only one considered in practice, takes the form

$$(I_s \otimes I_m - hA \otimes J)\left(Y^{(k+1)} - Y^{(k)}\right) =$$

$$-Y^{(k)} + h(A \otimes I_m)F(Y^{(k)}) + (U \otimes I_m)y^{[n]}.$$

This large linear system of size $s \times m$ represents the main computational cost of the method. It indeed requires the LU-decomposition of $(I_s \otimes I_m - hA \otimes J)$. For implicit linear multi-step methods, A has only one non-zero element, situated on the diagonal of A. As a consequence, the system under consideration is mostly explicit, the remaining implicitness concerning only a m-dimensional vector. As far as the non-linear system to be solved at each step is concerned, multi-step methods have a minimal cost per step. For implicit Runge-Kutta methods, i.e methods with $r = 1$, the matrix A is full and the cost of the resolution often considered as prohibitive for large systems. Several techniques have

been proposed for lowering this cost to a level comparable to what is observed for multi-step methods. They all make use of a similarity transformation of the matrix A into a simpler form

$$A = T^{-1}\Lambda T,$$

where Λ has a *nice* structure. For instance, for Singly Implicit Runge-Kutta (SIRK) methods [1] or Effective Singly Implicit Runge-Kutta (ESIRK) methods [6] Λ is of the form

$$\begin{bmatrix} \lambda & 0 & \cdots & & 0 \\ 1 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & \lambda \end{bmatrix},$$

and leads to the resolution of s consecutive sub-systems of dimension m with the same matrix $I_m - h\lambda J$. For Radau and Gauss methods [8, 27], Λ is block-diagonal and allows for some savings as well. One of the great advantages of multi-value methods is that A can actually be made diagonal, allowing for the possibility to decouple the non-linear system into s sub-systems

$$Y_i^{[n+1]} = h\lambda_i f(Y_i^{[n+1]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n]}, \ i = 1, \ldots, s,$$

which can be solved independently. With respect to the core of computations, these methods can thus be regarded as *parallel* methods, with a computational cost similar to multi-step methods.

2.3. Why are multi-value methods not more widespread?

Until recently and only since the apparition of DIMSIMs, very few multi-value methods have been used. There are several reasons for that, related to the obstacles encountered either in their construction or in their implementation.

Order conditions When one wants to design a multi-value method, one is confronted with the necessity to solve *order conditions*. These conditions can be here obtained from the analysis conducted in Section 2.2. For $\mathcal{M}$ to be of order p, Φ has to be of order p, i.e. its coefficients $\varphi(t)$ have to coincide with those of $y(x+h)$ for all trees of order less than or equal to p. These conditions on φ are known to be of the form

$$\forall t \in T, \rho(t) \leq p, \varphi(t) = 1. \tag{2.10}$$

$\mathcal{M}$ is of order p iff there exists a map $\zeta^* \in (\mathbb{R}^n)^T$ and a map $\varphi \in \mathbb{R}^T$ satisfying (2.10) such that

$$\begin{aligned} V\zeta^*(\emptyset) &= \zeta^*(\emptyset), \\ U\zeta^*(\emptyset) &= e, \end{aligned}$$

and such that relations (2.5, 2.6) are satisfied for all trees of order less or equal to p. For instance, we get for $p = 2$

$$\begin{aligned} V\zeta^*(\emptyset) &= \zeta^*(\emptyset), \\ U\zeta^*(\emptyset) &= e, \\ v^T Be &= 1, \\ v^T(BAe + BU(I - \widetilde{V})^{-1}Be) &= \frac{3}{2}. \end{aligned}$$

Our purpose here is not to construct a second order method, but to show the high complexity of the *order conditions*. For order three, these conditions are already hard to write down, unless clever simplifying assumptions are made (see [2] for a description of such assumptions). We will see in the next subsection how DIMSIMs overcome this difficulty by using a simplifying assumption on the stage values $Y^{[n+1]}$ and on the *exact value* function.

Starting procedure The *starting procedure* of a general multi-value method can be obtained once again from the analysis of Section 2.2. If the matrices A, U, B and V are given, it is possible to recover this *starting procedure* $z(x, h)$ by looking for its B-series. However, one then needs to construct effectively a set of r numerical methods with the prescribed B-series. This is a quite heavy task in the general case, if $z(x, h)$ has no particular pattern. Consider for instance a third-order method $\mathcal{M}$ with $r \geq 2$ such that

$$\begin{aligned} z_1(x,h) &= y(x), \\ z_2(x,h) &= \frac{h^2}{2}F(\bullet\!\!\!|)(y(x)) + \frac{h^3}{6}\left(F(\bullet\!\!\!\vee)(y(x)) + 2F(\bullet\!\!\!|\!\!\!|)(y(x))\right). \end{aligned}$$

One has to construct a Runge-Kutta method (A, b, c) such that

$$b^T e = 0,\ b^T c = \frac{1}{2},\ b^T c^2 = \frac{1}{3},\ b^T Ac = \frac{2}{6}.$$

For instance, the following method

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline 0 & -\frac{3}{4} & 0 & \frac{3}{4} \end{array}$$

furnishes a third-order approximation to $z_2(x,h)$. However, no less than 3 stages are necessary in this case, and the construction of such *starting procedures* would soon become tedious for higher orders without some assumptions on the structure of $z(x,h)$.

Change of step-size Changing the step-size can be quite tricky for a multi-value method having an *exact value* function with no special pattern. For instance, for $z(x,h)$ as in previous paragraph, an approximation of the term $f_{yy}(y(x))(f(y(x)),f(y(x)))+2f_y(y(x))f_y(y(x))f(y(x))$ is needed for any change of step-size. The term $f_y(y(x))f(y(x))$ is easy to estimate since it is equal to $y''(x)$. This is not the case of $f_{yy}(f,f)+2f_yf_yf$ which differs from

$$y^{(3)}(x) = F(\;\bullet\!\!\diagup\!\!\diagdown\!\!\bullet\;)(y(x)) + F(\;\bullet\!\!\!\!\bullet\;)(y(x)).$$

As a consequence, $f_{yy}(y(x))(f(y(x)),f(y(x)))$ and $f_y(y(x))f_y(y(x))f(y(x))$ have to be approximated individually. In general, special formulae have to be constructed to this aim. For our example, one can restart the method from the approximation of $z_1(x,h)=y(x)$ obtained at the current step with the Runge-Kutta method designed in previous paragraph. Although possible in principle, all this procedure introduces a lot of intermediate computations difficult to handle both from the point of view of implementation and from the point of view of stability analysis.

2.4. DIMSIMs

DIMSIMs were introduced by J. C. Butcher in 1993 [9] as a mean to make multi-value methods easy to construct and easy to implement. DIMSIM have a lot of features which make them very attractive for practical purposes.

Interpretation of DIMSIMs The key idea to DIMSIMs is to consider *exact value* functions with a simple form : if p is the order of the method under consideration, $z(x,h)$ is assumed to be a linear combination of full derivatives $y^{(k)}(x)$ with weights $\alpha_0 \in \mathbb{R}^r$, $\alpha_1 \in \mathbb{R}^r$, ..., $\alpha_p \in \mathbb{R}^r$,

$$z(x,h) = \sum_{k=0}^{p} h^k \alpha_k \otimes y^{(k)}(x).$$

The B-series $B(\zeta, y(x))$ representing $z(x,h)$ is such that whenever $0 \le \rho(u) = \rho(v) = k \le p$,

$$\zeta(u) = \zeta(v) = k!\,\alpha_k. \tag{2.11}$$

As a direct consequence the composition $(\varphi\zeta)(t)$ for $\varphi \equiv 1$ is considerably simplified for $\rho(t) \le p$, since we now have

$$\forall\, t \in T,\ \rho(t) \le p, \qquad (\varphi\zeta)(t) = \sum_{k=0}^{\rho(t)} \left(\begin{array}{c} \rho(t) \\ k \end{array} \right) k!\,\alpha_k.$$

This much simplified formula will help in getting order conditions in a nice form.

Different groups of DIMSIMS DIMSIMs have been divided into 4 different classes depending on the structure of the matrix A, which is nevertheless always assumed to be of the form :

$$\begin{bmatrix} \lambda & 0 & \cdots & 0 \\ \times & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \times & \cdots & \times & \lambda \end{bmatrix}.$$

The pattern of A has been chosen in order to minimize the computational cost per step (see Table 4). Whenever all the coefficients below the diagonal are

Type	λ	$a_{i,j}, j < i$	Parallelism	Class of problems
1	0	Arbitrary	No	Non-stiff
2	$\neq 0$	Arbitrary	No	Stiff
3	0	0	Yes	Non-stiff
4	$\neq 0$	0	Yes	Stiff

TABLE 4. The 4 classes of DIMSIMs.

zero, the method offers some intrinsic parallelism, as explained in Section 2.2. In the sequel, we will concentrate on type 4 methods, for which A is diagonal with constant diagonal and on methods that are an immediate generalization in the sense that diagonal elements of A may be different.

In an orthogonal way, DIMSIMs may as well be divided into two groups, according to whether their stage-order q is equal to their order p or to $p-1$. A multi-value method is said to be of stage-order q if its stage values $Y_i^{[n+1]}$ approximate the exact solution at some points $x_n + c_i h$ up to order q. Due to the following result of Burrage and Moss [3], this assumption is also of great significance and is satisfied for all DIMSIMs.

THEOREM 2.4. *Consider a multi-value method such that (2.11) holds. The simplifying assumptions*

$$\frac{c^j}{j!} = \frac{1}{(j-1)!} A c^{j-1} + U \alpha_j, \ j = 0, \ldots, p-1,$$

where the product of vectors is meant to be component-wise, together with the order conditions

$$\begin{aligned} \alpha_0 &= V \alpha_0, \\ \sum_{k=0}^{j} \frac{\alpha_k}{(j-k)!} &= \frac{1}{(j-1)!} B c^{j-1} + V \alpha_j, \ j = 1, \ldots, p, \end{aligned}$$

imply that the method is of order p.

Compared to the order conditions derived in Section 2.3, the conditions obtained here are extremely simple. Order conditions given in the sequel will all be based on this theorem.

Implementation of DIMSIMs Let us now demonstrate some of the practical advantages of DIMSIMs :

- *starting procedure* : due to the particular form of the *exact value* function, getting started with a DIMSIM is not difficult. Amongst various possibilities, one is to perform one step of a Runge-Kutta method with stage-order greater or equal to p (for instance a SIRK method). The derivatives of stage values obtained can then be combined in an appropriate way to get the desired approximation to $y^{[0]}$.
- *change of step-size* : suppose the DIMSIM is of stage-order q equal to its order. Then, it has been shown in [7] that the method can be reformulated so as to approximate the Nordsieck vector. Changing the step-size is then just a matter of scaling. For methods with $q = p - 1$, a similar procedure is a-fortiori possible.
- *dense output* : getting a highly accurate dense output is easily achieved by combining the stage-values for example. If the method is expressed in its Nordsieck form, this is even easier :

$$y(x_n + th) \approx \left([1, t, \frac{t^2}{2}, \ldots, \frac{t^p}{p!}]^T \otimes I_m \right) y^{[n]}.$$

3. Construction of parallel implicit DIMSIMs

3.1. A general framework

From now on, the focus will be put on DIMSIMs of type 4. This is motivated by the fact that most large systems involve some stiffness and thus necessitate an implicit method, which in turn is costly and can greatly benefit from parallelism.

p^{th}-order DIMSIMs with stage-order p In order to go a little further than Theorem 2.4, we need to make some assumptions on r and s. Since it is more convenient to work with square matrices, J.C. Butcher has derived the following theorem :

Theorem 3.1. *Let $s = r = p$ and suppose that the DIMSIM is of the form*

$$\begin{bmatrix} A & I \\ B & V \end{bmatrix},$$

with $Ve = e$. It is of order p and stage-order p iff

$$B = B_0 - AB_1 - VB_2 + VA,$$

where the (i,j) elements of B_0, B_1 and B_2 are respectively given by

$$\frac{\int_0^{1+c_i} \Phi_j(x)dx}{\Phi_j(c_j)}, \frac{\Phi_j(1+c_i)}{\Phi_j(c_j)} \text{ and } \frac{\int_0^{c_i} \Phi_j(x)dx}{\Phi_j(c_j)},$$

$\Phi_j(x)$ being the polynomial $\prod_{k\neq j}(x-c_k)$.

Using this theorem is extremely easy. Suppose we choose for instance $c = [0,1]^T$, than we have [9]

$$B_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 2 \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \text{ and } B_2 = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Taking

$$V = \begin{bmatrix} v & 1-v \\ v & 1-v \end{bmatrix} \text{and } A = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix},$$

we get

$$B = \begin{bmatrix} 1/2\,v + v\lambda & -\lambda + 1/2\,v + (1-v)\,\lambda \\ \lambda - 1/2 + 1/2\,v + v\lambda & 3/2 - 2\,\lambda + 1/2\,v + (1-v)\,\lambda \end{bmatrix}.$$

Linear stability The domain of stability of a numerical method is a fundamental characteristic. The larger its intersection with the left-half plane, the better. For the linear test problem

$$\begin{cases} y'(x) &= \mu y(x), \\ y(x_0) &= y_0, \end{cases}$$

the DIMSIM step to step formulae become

$$\begin{aligned} y^{[n+1]} &= (h\mu)BY^{[n+1]} + Vy^{[n]}, \\ Y^{[n+1]} &= (h\mu)AY^{[n+1]} + Uy^{[n]}, \end{aligned}$$

that is to say

$$y^{[n+1]} = M(h\mu)y^{[n]},$$

where $M(z)$ is the *stability matrix*

$$M(z) := V + zB(I_s - zA)^{-1}U.$$

Similarly to any multi-step or Runge-Kutta method, a DIMSIM is said to be A-stable iff the *stability domain*

$$S := \{z \in \mathbb{C}, \exists\, C \in \mathbb{R}_+^*, \forall n \in \mathbb{N}, \|M(z)\|^n \leq C\}$$

contains $\mathbb{C}^-$ (the left-half plane). We see that a point z of the complex plane belongs to S iff the *characteristic polynomial* of $M(z)$ has all its roots within the unit disc and is such that the roots on the unit circle correspond to non-defective eigenvalues of $M(z)$. Using the maximum principle for the spectral radius of $M(z)$, we can formulate the following theorem from [17], based on the Routh-Hurwitz criterion [25] :

THEOREM 3.2. *Let $\pi(t,z) := \det(M(z) - tI_r)$ denote the characteristic polynomial of $M(z)$ and let ε_j be the coefficient of ω^j, $j = 0, \ldots, 2r$, in*

$$W(\omega, y) = (\omega - 1)^{2r} \pi\left(\frac{\omega+1}{\omega-1}, iy\right) \pi\left(\frac{\omega+1}{\omega-1}, -iy\right).$$

Moreover, let $\mathcal{H}$ be the $2r \times 2r$ matrix

$$\mathcal{H} = \begin{bmatrix} \varepsilon_{2r-1} & \varepsilon_{2r-3} & \varepsilon_{2r-5} & \cdots & \varepsilon_{1-2r} \\ \varepsilon_{2r} & \varepsilon_{2r-2} & \varepsilon_{2r-4} & \cdots & \varepsilon_{2-2r} \\ 0 & \varepsilon_{2r-1} & \varepsilon_{2r-3} & \cdots & \varepsilon_{3-2r} \\ 0 & \varepsilon_{2r} & \varepsilon_{2r-2} & \cdots & \varepsilon_{4-2r} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \varepsilon_{2r} & \cdots & \varepsilon_0 \end{bmatrix},$$

where $\varepsilon_j = 0$ for $j < 0$, and $\Delta(y)$ its principal minor of order $(2r-1)$. Then the following set of conditions is sufficient for A-stability :

$$\begin{cases} \exists y^* \in \mathbb{R}_+^*, \forall\, \omega \in \mathbb{C}, W(\omega, y^*) = 0 \Rightarrow \mathrm{Re}(\omega) < 0, \\ \forall\, y \in \mathbb{R}_+^*, \varepsilon_0(y) \neq 0, \\ \forall\, y \in \mathbb{R}_+^*, \varepsilon_{2r}(y) \neq 0, \\ \forall\, y \in \mathbb{R}_+^*, \Delta(y) \neq 0. \end{cases} \tag{3.12}$$

For the method obtained in previous section, we have

$$M(z) = \begin{bmatrix} -1/2\,\frac{v(2+z)}{-1+z\lambda} & 1/2\,\frac{-2+2\,z\lambda+2\,v-zv}{-1+z\lambda} \\ -1/2\,\frac{2\,v+2\,z\lambda-z+zv}{-1+z\lambda} & 1/2\,\frac{-2+4\,z\lambda+2\,v-3\,z-zv}{-1+z\lambda} \end{bmatrix},$$

and

$$\pi(t,z) = t^2 - 1/2\,\frac{(4\,z\lambda - 2\,zv - 3\,z - 2)\,t}{-1+z\lambda} +$$

$$1/2\,\frac{z\left(-z\lambda - 2\,zv\lambda + 2\,v - 2\,\lambda + 2\,zv + 2\,z\lambda^2 + 1\right)}{(-1+z\lambda)^2}.$$

If we further impose that the matrix $M(z)$ has vanishing eigenvalues at infinity, v and λ must satisfy the equations

$$\begin{cases} 2\,v - 2\,v\lambda - \lambda + 2\,\lambda^2 &= 0, \\ -4\,\lambda^2 + 2\,v\lambda + 3\,\lambda &= 0, \end{cases}$$

and we finally arrive at the following two methods

$$\mathcal{M}_{\pm} = \left[\begin{array}{cc|cc} 3/2 \pm 1/2\sqrt{3} & 0 & 1 & 0 \\ 0 & 3/2 \pm 1/2\sqrt{3} & 0 & 1 \\ \hline 9/2 \pm 11/4\sqrt{3} & -3 \mp 7/4\sqrt{3} & 3/2 \pm \sqrt{3} & -1/2 \mp \sqrt{3} \\ 11/2 \pm \frac{13}{4}\sqrt{3} & -3 \mp 9/4\sqrt{3} & 3/2 \pm \sqrt{3} & -1/2 \mp \sqrt{3} \end{array}\right].$$

And we get

$$\begin{aligned} \varepsilon_0(y) &= -1/4\left(-7 \mp 4\sqrt{3}\right)\left(9\,y^4 + 100\,y^2 \mp 56\,y^2\sqrt{3} + 112 \mp 64\sqrt{3}\right), \\ \varepsilon_4(y) &= -1/4\left(-7 \mp 4\sqrt{3}\right)\left(9\,y^2 + 28 \mp 16\sqrt{3}\right)y^2, \\ \Delta(y) &= -\left(-1351 \mp 780\sqrt{3}\right)\left(81\,y^4 + 252\,y^2 \mp 126\,y^2\sqrt{3} + 364 \mp 208\sqrt{3}\right) \\ &\quad \times \left(9\,y^4 + 4\,y^2 \pm 2\,y^2\sqrt{3} + 28 \mp 16\sqrt{3}\right)y^4, \end{aligned}$$

so that $\mathcal{M}_-$ and $\mathcal{M}_+$ are A-stable.

3.2. L-stable methods

In some situations, it may be desirable to damp the very stiff components of the solution. This has led to the definition of L-stability : a Runge-Kutta method is said to be L-stable if its stability function vanishes at infinity. For multi-value methods, one can either require that $\rho(M(\infty))$ vanishes or that $M(\infty)$ vanishes. In the next two paragraphs, we will construct methods of both kinds.

Methods with zero spectral radius at infinity Consider a r-step s-stage DIMSIM with $r = s$ of the form

$$\left[\begin{array}{c|c} \lambda I_r & I_r \\ \hline B & ev^T \end{array}\right],$$

with $v^T e = 1$ and $B = B_0 - \lambda B_1 - ev^T B_2 + \lambda ev^T$. With these special choices, the stability matrix becomes

$$M(z) = V + \frac{z}{1-\lambda z}B,$$

with characteristic polynomial of the form

$$\begin{aligned} \pi(t,z) &= \det(ev^T + \frac{z}{1-\lambda z}B - tI_r), \\ &= \det(e_r e_r^T + \frac{z}{1-\lambda z}\widetilde{B} - tI_r), \\ &= \det(\frac{z}{1-\lambda z}\widetilde{B} - tI_r) + \det(\frac{z}{1-\lambda z}\widetilde{B}_{r-1} - tI_{r-1}), \\ &= t^r - \frac{(a_0 - b_1 z)}{(1-\lambda z)}t^{r-1} - z\frac{(a_1 - b_2 z)}{(1-\lambda z)^2}t^{r-2} - \ldots - z^{r-1}\frac{(a_{r-1} - b_r z)}{(1-\lambda z)^r}, \end{aligned}$$

where $\widetilde{B}$ is a matrix obtained from B by a similarity transformation and $\widetilde{B}_{r-1}$ is its $(r-1)^{th}$ main minor. The method being of order $p = r$, we have

$$\pi(e^z, z) = \mathcal{O}(z^{r+1}),$$

which we can write as

$$e^z(1-\lambda z) = \widehat{F}\left(z\frac{e^z}{1-\lambda z}\right) + \mathcal{O}(z^{r+1}),$$

where

$$\widehat{F}(Z) = \frac{\sum_{i=0}^{r-1} a_i Z^i}{1 + \sum_{i=1}^{r} b_i Z^i}.$$

$\widehat{F}$ is thus a rational approximation of a function F defined by

$$F\left(z\frac{e^z}{1-\lambda z}\right) = e^z(1-\lambda z).$$

In [4], J.C. Butcher has shown that

$$F(Z) = 1 + \sum_{n=1}^{\infty}(-1)^{n+1}\frac{\lambda^n}{n+1}L'_{n+1}\left(\frac{n+1}{\lambda}\right)Z^n,$$

where $L_n(x)$ denotes the Laguerre polynomial $\sum_{i=0}^{n}\binom{n}{i}(-x)^i/i!$. Now, if we substitute $z = \infty$ in $\pi(z,t)$, we see that the method satisfies $\rho(M(\infty)) = 0$ iff all the b_i's vanish. In this case, the rational approximation we look for can be obtained by truncating F after the Z^{r-1}-term and imposing that

$$L'_{r+1}\left(\frac{r+1}{\lambda}\right) = 0. \tag{3.13}$$

It can be shown that $\pi(z,t)$ satisfies the Hurwitz criterion for all values of r between 1 and 8 [4] if we take for λ the smallest middle root of (3.13). The method $\mathcal{M}_-$ from previous section is an example of such a L-stable method for $r = 2$.

Methods with vanishing stability matrix at infinity Imposing that $M(\infty) = 0$ is a much stringer requirement and leaves very few free parameters. If we denote by $l_j(x)$ the j^{th} Lagrange polynomial based on the abscissae c_j, $j = 1, \ldots, r$, we have the following theorem :

THEOREM 3.3. *Let us consider a r-step r-stage DIMSIM of the form*

$$\left[\begin{array}{c|c} A & I_r \\ \hline B & V \end{array}\right],$$

with $A = diag(\lambda_1, \ldots, \lambda_s)$ *and* $M(\infty) = V - BA^{-1} = 0$. *Then it is of order* r *and stage-order* r *iff*

$$\begin{aligned} \lambda_i &= \frac{l_i(1+c_i)}{l_i(1+c_i) + l_i'(1+c_i)},\ i = 1, \ldots, r, \\ v_{i,j} &= \frac{l_j^2(1+c_i)}{(l_i(1+c_i) + l_i'(1+c_i))l_j(c_j)},\ i = 1, \ldots, r,\ j = 1, \ldots, r. \end{aligned}$$

Note that if $c_i = -r + 1 + i$, $i = 1, \ldots, r$, than the method obtained is the r-step Backward Differentiation Formula (BDF). However, BDF suffer from a limited stability and this seems to be a common feature of all these methods [5]. In order to get A-stable methods, one consequently needs to relax some of the conditions :

THEOREM 3.4. *Let us consider a* r*-step* r*-stage DIMSIM of the form*

$$\left[\begin{array}{c|c} A & I_r \\ \hline B & V \end{array}\right],$$

with $A = diag(\lambda_1, \ldots, \lambda_s)$ *and* $M(\infty) = V - BA^{-1} = 0$. *Then it is of order* $r-1$ *and stage-order* $r-1$ *iff*

$$V = L - AL',$$

where the (i, j) *elements of* L *and* L' *are respectively*

$$l_j(1+c_i) \text{ and } l_j'(1+c_i).$$

These methods have one order less but many potential advantages : they can be easily written in the Nordsieck form, they allow easy local error estimate and they are A-stable for some choices of the diagonal elements of A and of the c_i's up to very high orders (at least 12 [17]).

4. PRACTICAL IMPLEMENTATION OF A SIX-PROCESSORS FIFTH-ORDER METHOD

We shall now discuss various aspects of the implementation of a method, whose characteristics are :

- $r = s = 6$, $p = q = 5$,
- $M(\infty) = 0$,
- A-stable.

It has been derived using Theorem 3.4 with the special requirement that V is upper-triangular for the Nordsieck formulation

$$z(x,h) = \begin{bmatrix} y(x) \\ hy'(x) \\ \vdots \\ \frac{h^p}{p!} y^{(p)} \end{bmatrix}.$$

Note that it can be shown that V is upper-triangular iff [5]

$$\lambda_i = \mu + \nu c_i.$$

Its tableau has the form :

$$\left[\begin{array}{cccccc|cccccc}
\frac{2}{5} & 0 & 0 & 0 & 0 & 0 & 1 & -\frac{2}{5} & 0 & 0 & 0 & 0 \\
0 & \frac{7}{15} & 0 & 0 & 0 & 0 & 1 & -\frac{4}{15} & -\frac{11}{75} & -\frac{6}{125} & -\frac{1}{75} & -\frac{32}{9375} \\
0 & 0 & \frac{8}{15} & 0 & 0 & 0 & 1 & -\frac{2}{15} & -\frac{4}{15} & -\frac{24}{125} & -\frac{208}{1875} & -\frac{544}{9375} \\
0 & 0 & 0 & \frac{3}{5} & 0 & 0 & 1 & 0 & -\frac{9}{25} & -\frac{54}{125} & -\frac{243}{625} & -\frac{972}{3125} \\
0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 1 & \frac{2}{15} & -\frac{32}{75} & -\frac{96}{125} & -\frac{1792}{1875} & -\frac{9728}{9375} \\
0 & 0 & 0 & 0 & 0 & \frac{11}{15} & 1 & \frac{4}{15} & -\frac{7}{15} & -\frac{6}{5} & -\frac{29}{15} & -\frac{8}{3} \\
\hline
0 & 0 & 0 & 0 & 0 & \frac{11}{15} & 1 & \frac{4}{15} & -\frac{7}{15} & -\frac{6}{5} & -\frac{29}{15} & -\frac{8}{3} \\
-\frac{2}{5} & \frac{35}{12} & -\frac{80}{9} & 15 & -\frac{50}{3} & \frac{1507}{180} & 0 & \frac{2}{3} & -\frac{2}{15} & -\frac{12}{5} & -\frac{92}{15} & -\frac{34}{3} \\
-\frac{25}{6} & \frac{2135}{72} & -\frac{260}{3} & \frac{535}{4} & -\frac{1925}{18} & \frac{275}{8} & 0 & 0 & \frac{1}{3} & -\frac{6}{5} & -\frac{34}{5} & -\frac{56}{3} \\
-\frac{175}{12} & \frac{7175}{72} & -\frac{2450}{9} & \frac{1475}{4} & -\frac{8875}{36} & \frac{4675}{72} & 0 & 0 & 0 & 0 & -\frac{44}{15} & -\frac{44}{3} \\
-\frac{125}{6} & \frac{9625}{72} & -\frac{1000}{3} & \frac{1625}{4} & -\frac{4375}{18} & \frac{1375}{24} & 0 & 0 & 0 & 0 & -\frac{1}{3} & -\frac{16}{3} \\
-\frac{125}{12} & \frac{4375}{72} & -\frac{1250}{9} & \frac{625}{4} & -\frac{3125}{36} & \frac{1375}{72} & 0 & 0 & 0 & 0 & 0 & -\frac{2}{3}
\end{array}\right],$$

and $c = [0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1]^T$.

4.1. Local error estimation

In its variable step-size version, the previous method uses the formulae

$$\begin{aligned} Y^{[n+1]} &= h_n(A \otimes I_m)F(Y^{[n+1]}) + (UD(\delta_n) \otimes I_m)y^{[n]}, \\ y^{[n+1]} &= h_n(B \otimes I_m)F(Y^{[n+1]}) + (VD(\delta_n) \otimes I_m)y^{[n]}, \end{aligned}$$

where $\delta_n = \frac{h_n}{h_{n-1}}$ and[1]

$$D(\delta_n) = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & \delta_n & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \delta_n^p \end{bmatrix}.$$

It has been shown in [7], that the global error of a DIMSIM may be controlled through an accurate estimate of the quantity

$$err = h_n^{p+1}(v^T\varphi_p)y^{(p+1)}(x_{n-1}),$$

where v^T is the left-eigenvector of V such that $v_1 = 1$ associated with eigenvalue 1 and where

$$\varphi_p = [\frac{1}{(p+1)!\,0!}, \frac{1}{p!\,1!}, \cdots, \frac{1}{1!\,p!}]^T - \frac{1}{p!}Bc^p.$$

[1] Contrary to the methods with a rank-one V-matrix, zero-stability of previous method is not strictly guaranteed for variable step-size, unless very restrictive assumptions are made.

It can be checked that we have here

$$C := v^T \varphi_5 = \frac{5539}{4500000} \approx 0.0012.$$

Now, the stage-order being equal to the order, an asymptotically correct estimate of $h_n^6 (v^T \varphi_5) y^{(6)}(x_{n-1})$ can be obtained by a linear combination of the stage derivatives satisfying

$$\sum_{i=1}^{6} \gamma_i h_n f(Y_i^{[n+1]}) = h_n^6 C y^{(6)}(x_{n-1}) + \mathcal{O}(h_n^7),$$

and a simple Taylor expansion gives

$$\begin{aligned} [\gamma_1, \ldots, \gamma_6] &= C\ [0, \ldots, 0, 1]\ [e, c, \ldots, \frac{1}{5!} c^5]^{-1}, \\ &= 3125\ C\ [-1, 5, -10, 10, -5, 1]. \end{aligned}$$

4.2. Simplified Newton iterations

As explained in Section 2.2, we have to solve $s = 6$ sub-systems of the form

$$Y_i^{[n+1]} = h_n \lambda_i f(Y_i^{[n+1]}) + \sum_{j=1}^{6} u_{ij} y_j^{[n]},$$

by the *simplified Newton method*. For the i^{th} sub-system, it takes the form

$$\begin{aligned} (I_m - h_n \lambda_i J) \Delta Z_i^{(k)} &= -Z_i^{(k)} + h_n \lambda_i f(Z_i^{(k)} + \sum_{j=1}^{6} u_{ij} y_j^{[n]}), \\ Z_i^{(k+1)} &= Z_i^{(k)} + \Delta Z_i^{(k)}, \end{aligned}$$

where $Z_i^{(k)}$ is a hopefully convergent approximation to $Y_i^{[n+1]} - \sum_{j=1}^{6} u_{i,j} y_j^{[n]}$. For the control of the iteration, we shall use the strategy developed in [27].

4.3. Step-size control strategy

We shall use the standard formula

$$h_{new} = h_{old} \left(\frac{TOL}{\|err\|} \right)^{\frac{1}{6}},$$

in association with a formula based on PID-control [23]. However, in order to save some computations, we will recompute the Jacobian only if h_{new} satisfies

$$h_{new} \geq 1.3 h_{old} \text{ or } h_{new} \leq 0.8 h_{old}$$

in order to avoid too frequent LU-decompositions of the matrices $I_m - h\lambda_i J$ [27].

5. Numerical tests

In order to demonstrate the correct behavior of the step-size control mechanism, let us consider the Van der Pol equation as in [27] pp. 135 :

$$\left\{ \begin{array}{rclcrcl} y_1' &=& y_2, & & y_1(0) &=& 2, \\ y_2' &=& \frac{1}{\varepsilon}\left((1-y_1^2)y_2 - y_1\right), & & y_2(0) &=& -0.6, \end{array} \right. \tag{5.14}$$

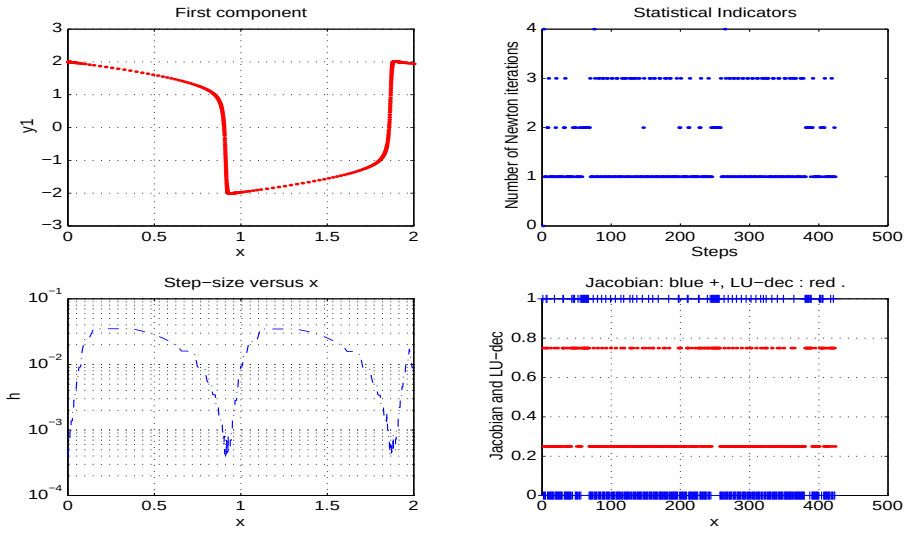


Figure 1. Results for $\varepsilon = 10^{-2}$

with $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-6}$. We have run our Matlab program with $TOL = ATOL = RTOL = 10^{-6}$ for both cases. It can be observed from Figure 1 and Figure 2 that the step-size is chosen appropriately even in the case of extremely rapid change of the solution. On Figure 3, we present work-precision diagrams for our method and the code VODE, which is based on a Nordsieck formulation of linear multi-step methods. The number of function evaluations for the DIMSIM method is divided by the *theoretical* speed-up, i.e. 6. So is done for the number of LU-decompositions.

As a second example, we consider the Brusselator problem :

$$\begin{array}{rcl} \dfrac{\partial u}{\partial t} &=& A + u^2 v - (B+1)u + \beta \dfrac{\partial^2 u}{\partial x^2} \\ \dfrac{\partial v}{\partial t} &=& Bu - u^2 v + \beta \dfrac{\partial^2 v}{\partial x^2} \end{array} \tag{5.15}$$

with $0 \leq x \leq 1$, $A = 1$, $B = 3$, $\beta = 1/50$ and boundary conditions

$$\begin{array}{ll} u(0,t) = u(1,t) = 1, & v(0,t) = v(1,t) = 3, \\ u(x,0) = 1 + \sin(2\pi x), & v(x,0) = 3. \end{array} \tag{5.16}$$

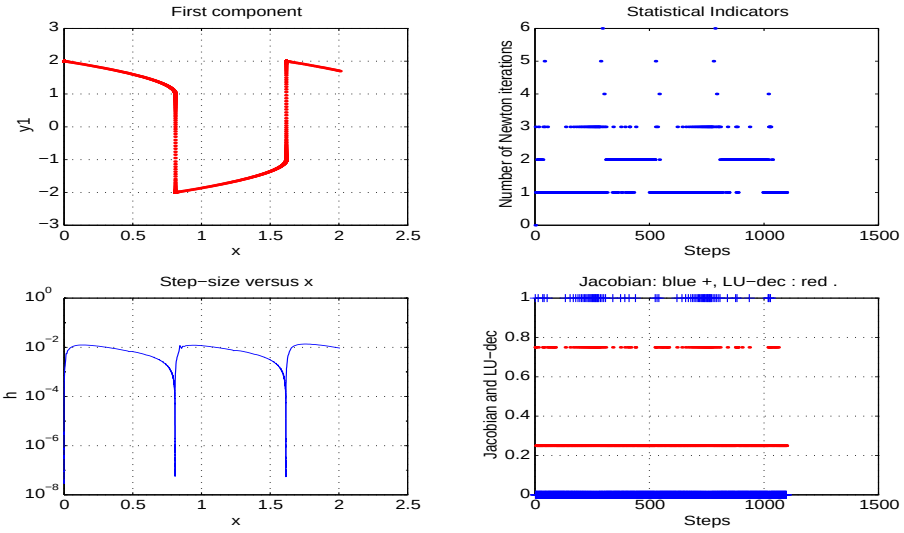


FIGURE 2. Results for $\varepsilon = 10^{-6}$

Using finite differences on a grid of N points we obtain a $2N$-dimensional ODE-system [27]. We have run our program for $N = 20$ for different tolerances. The behavior of the code for $TOL = RTOL = ATOL = 10^{-6}$ is described on Figure 4. A work-precision diagrams are given in Figure 5 for both our method and VODE.

6. CONCLUSION

Multi-value methods have been introduced primarily as a unifying tool in the theory of numerical methods. We have tried to show however that they are much more than this and that they contain a very rich range of new methods with clear advantages over traditional multi-step methods or Runge-Kutta methods. With respect to their construction, DIMSIMs seem very attractive. Several sub-classes of methods have been identified for stiff problems, where stability requirements usually lead to prohibitive costs. DIMSIMs combine the advantages of Runge-Kutta methods (A-stability for high orders) and of multi-step methods (low computational cost). Up to now, the difficulties encountered in the implementation of multi-value methods have prevented them from being used efficiently. To a large extent, these difficulties have been overcome and the tests presented here illustrate that a DIMSIM code can now be equipped with all the features of a modern ODE-solver.

Finally, let us mention that DIMSIMs exhibit no order reduction for differential-algebraic systems of index 1 or 2. As a consequence, they should be considered as good candidates for the resolution of such systems. Another potential field of application concerns Hamiltonian systems : it seems however

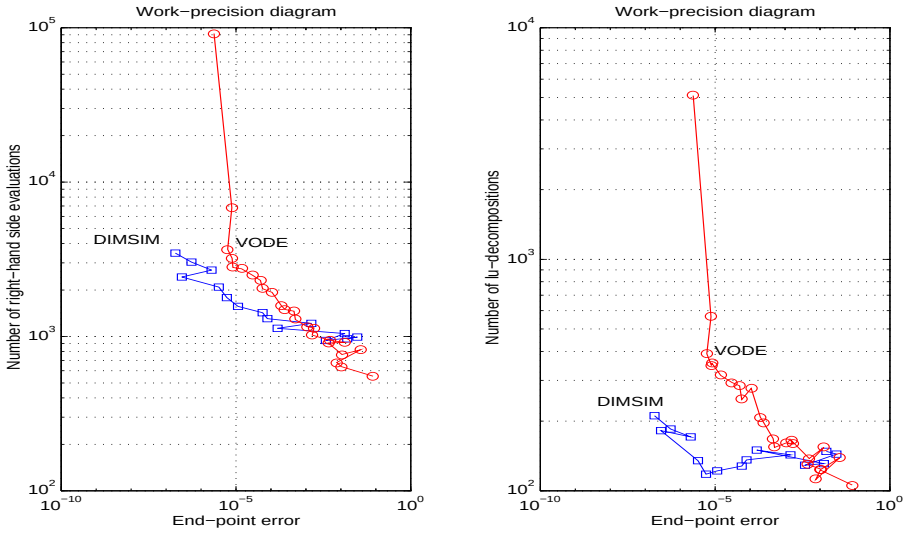


FIGURE 3. Work-precision diagrams for the DIMSIM method and VODE.

very likely, according to recent results [24], that there exists no symplectic multi-value method apart from standard symplectic Runge-Kutta.

REFERENCES

1. K. BURRAGE (1978). A special family of Runge-Kutta methods for solving stiff differential equations. *BIT* **18**, 22–41.
2. K. BURRAGE (1995). *Parallel and Sequential Methods for Ordinary Differential Equations.* Oxford Science Publications, 1995. Numerical Mathematics and Scientific Computation.
3. K. BURRAGE, P. MOSS (1980). Simplifying assumptions for the order of partitioned multivalue methods. *BIT* **20**, 452–465.
4. J. C. BUTCHER (1996). Order and Stability of Parallel Methods for stiff problems, 1996.
5. J. C. BUTCHER, P. CHARTIER (1994). *The construction of DIMSIMs for stiff ODEs and DAEs.* Technical Report 308, University of Auckland, Auckland, New Zealand, July 1994.
6. J. C. BUTCHER, P. CHARTIER (1997). A generalization of singly-implicit Runge-Kutta methods. *Applied Numerical Mathematics* **24**, 343–350.
7. J. C. BUTCHER, P. CHARTIER, Z. JACKIEWICZ (1997). Nordsieck representation of DIMSIMs. *Numerical Algorithms.* **16**, 209–230.
8. J.C. BUTCHER (1987). *The Numerical Analysis of Ordinary Differential Equations.* John Wiley and Sons.
9. J.C. BUTCHER (1993). Diagonally-Implicit Multi-Stage Integration Methods. *Applied Numerical Mathematics* **11**(380), 347–363. North-Holland.

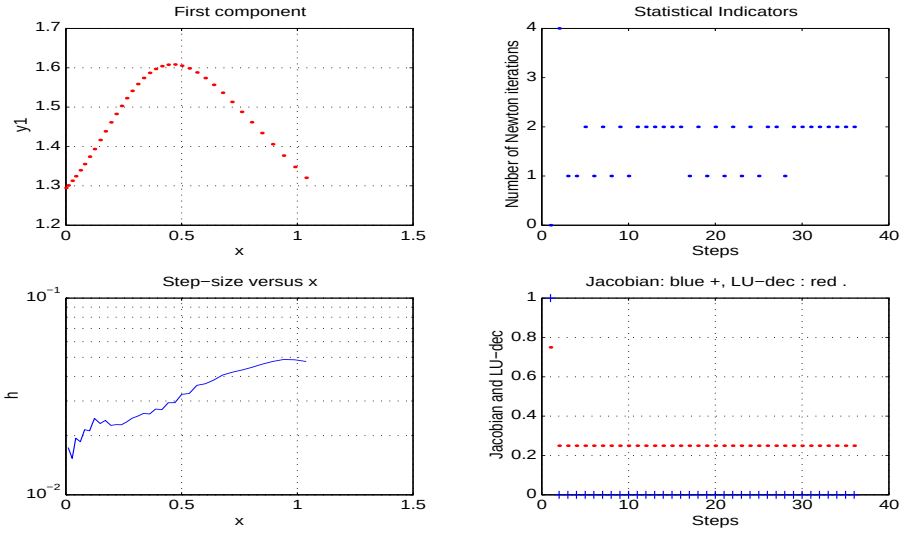


FIGURE 4. Results for $TOL = 10^{-6}$

10. J.C. BUTCHER (1993). General linear methods for the parallel solution of ordinary differential equations. *WSSIAA* **2**, 99–111. In memory of L. Collatz.
11. J.C. BUTCHER (1994). A transformation for the analysis of DIMSIMs. *BIT* **34**, 25–32.
12. J.C. BUTCHER, P. CHARTIER (1995). Parallel general linear methods for stiff ordinary differential and differential algebraic equations. *Applied Numerical Mathematics* **17**, 213–222.
13. J.C. BUTCHER, Z. JACKIEWICZ (1993). Diagonally implicit general linear methods for ordinary differential equations. *BIT* **33**, 452–472.
14. J.C. BUTCHER, Z. JACKIEWICZ, H.D. MITTELMANN (1997). Nonlinear optimization approach to the construction of general linear methods of high-order. *J. Comput. Appl. Math.* **81**, 181–196.
15. J.C. BUTCHER, Z. ZACKIEWICZ (1996). *Construction of high-order DIMSIMs for ordinary differential equations.* Submitted.
16. P. CHARTIER (1993). *Parallélisme dans la résolution numérique des problèmes de valeur initiale pour les équations différentielles ordinaires et algébriques.* PhD thesis, Université de Rennes 1, Rennes, France.
17. P. CHARTIER (1994). L-stable parallel one-block methods for ordinary differential equations. *SIAM Journal of Numerical Analysis* **31**(2).
18. M.T. CHU, H HAMILTON (1987). Parallel solution of ODEs by multi-block methods. *SIAM Journal of Scientific and Statistical Computing* **8**, 342–353.
19. M.A. FRANKLIN (1978). Parallel Solution of Ordinary Differential Equations. *IEEE Transactions on Computers*, C-**27**(5), 413–420, may.
20. C.W. GEAR (1986). *The potential for parallelism in ordinary differen-*

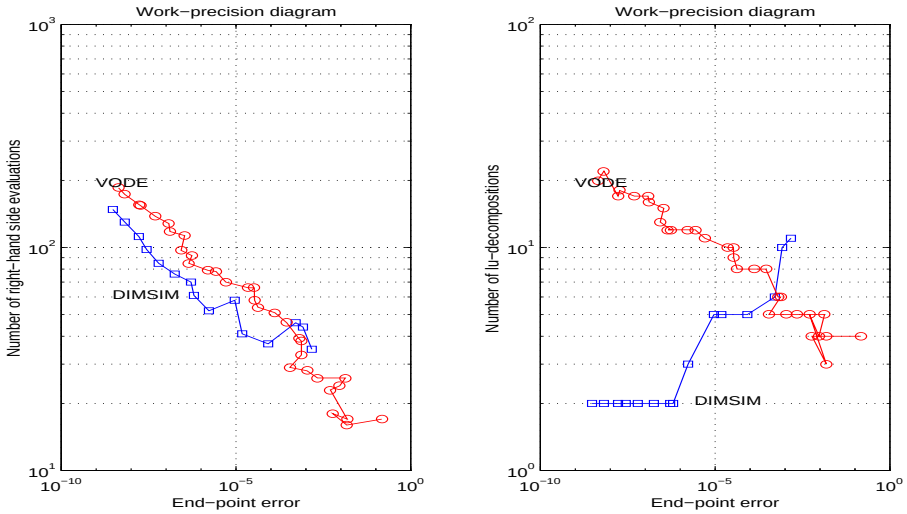


FIGURE 5. Work-precision diagrams for the DIMSIM method and VODE.

tial equations. Technical Report 1246, University of Illinois at Urbana-Champaign, U.S.A.

21. S. GONZÁLEZ-PINTO, C. GONZÁLEZ-CONCEPTIÓN, J.I. MONTIJANO (1993). Iterative Schemes for Gauss Methods. *Computers and Mathematics with Applications* **27**(7), 67–81.
22. S. GONZÁLEZ-PINTO, J.I. MONTIJANO, L. RÁNDEZ (1995). Iterative schemes for three-stage implicit Runge-Kutta methods. *Applied Numerical Mathematics* **17**, 363–382.
23. K. GUSTAFSSON, M. LUNDH, G. SÖDERLIND (1988). A PI stepsize control for the numerical solution of ordinary differential equations. *BIT* **28**, 270–287.
24. E. HAIRER, P. LEONE (1997). Order Barriers for Symplectic Multi-Value Methods. preprint.
25. E. HAIRER, S.P. NØRSETT, G. WANNER (1991). *Solving Ordinary Differential Equations, Nonstiff Problems.* Springer-Verlag, second revised edition. Volume 1.
26. E. HAIRER, G. WANNER (1974). On the Butcher group and general multi-value methods. *Computing* **13**, 1–15.
27. E. HAIRER, G. WANNER (1996). *Solving Ordinary Differential Equations, Stiff Problems and Differential Algebraic Problems.* Springer-Verlag, second edition. Volume 2.
28. K.R. JACKSON, S.P. NØRSETT (1990). The potential for parallelism in Runge-Kutta methods. part 1: RK formulas in Standard Form.
29. K.R. JACKSON, S.P. NØRSETT (1991). The potential for parallelism in

Runge-Kutta methods. part 2: RK predictor-corrector formulas.
30. O.A. Karakashian, W. Rust (1989). On the parallel implementation of implicit Runge-Kutta methods. *SIAM Journal of Scientific and Statistical Computing* **9**, 1085–1090.
31. W.M. Lioen (1996). On the diagonal approximation of full matrices. *Journal of Computational and Applied Mathematics* **75**, 35–42..
32. B.P. Sommeijer, W. Couzy, P.J. van der Houwen (1992). A-stable parallel block methods for ordinary and integro-differential equations. *Applied Numerical Mathematics* **9**, February.
33. D. Stoffer (1993). General linear methods: connection to one step methods and invariant curves. *Numerische Mathematik* **64**, 395–407.
34. P.J. van der Houwen, B.P. Sommeijer (1990). Parallel iteration of high-order Runge-Kutta methods with stepsize control. *Journal of Computational and Applied Mathematics* **29**, 111–127.
35. P.J. van der Houwen, B.P. Sommeijer (1990). Parallel Ode Solvers. In *1990 International Conference on SUPERCOMPUTING*. ACM PRESS, June.
36. P.J. van der Houwen, B.P. Sommeijer (1991). Iterated Runge-Kutta Methods on Parallel Computers. *SIAM Journal of Scientific and Statistical Computing* **12**(5), 1000–1028, September.
37. P.J. van der Houwen, B.P. Sommeijer, W. Couzy (1992). Embedded diagonally implicit Runge-Kutta algorithms on parallel computers. *Mathematics of Computation*.
38. J. White, F. Odeh, A.S. Vincentelli, A. Ruehli (1985). Waveform relaxation : theory and practice. *Transactions of Society for Computer Simulation* **2**, 95–133.