

Economical Error Estimates for Block Implicit Methods for ODEs via Deferred Correction *

Luigi Brugnano[†] Cecilia Magherini[‡]

Dipartimento di Matematica “U. Dini”
Viale Morgagni 67/A
50134 Firenze, Italy

Abstract

Deferred correction is a widely used tool for improving the numerical approximation to the solution of ODE problems [10, 11, 12, 13, 16, 17, 18, 19, 20, 21, 23]. Indeed, it allows to estimate the error due to the use of discrete methods. Such an estimate may be a global one, in the case of continuous BVPs, or a local one, when IVPs are to be approximated [2, 7]. Recently, it has been implemented in the computational code BiM [5] for the numerical solution of stiff ODE-IVPs. In this paper we analyze deferred correction in connection with the methods used in that code, resulting in an overall simplification of the procedure, due to the properties of the underlying methods. The analysis is then extended to more general methods.

Keywords: Numerical Methods for ODEs, Deferred Correction, Error Estimate, Vandermonde Matrix.

MSC: 65L05, 65L06, 65L50.

1 Introduction

Deferred correction is a useful framework for error estimation when solving ODEs [10, 11, 12, 13, 16, 17, 18, 19, 20, 21, 23]. 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, [11, 12]), 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., [2, 7]). This is exactly the use of deferred correction which has been recently considered in the computational

*Research supported by I.N.d.A.M. and Italian M.I.U.R.

[†]brugnano@math.unifi.it

[‡]magherini@math.unifi.it

code `BiM` for the numerical solution of ODE-IVPs [5, 6]. This code is based on the so called *block implicit methods* [22], i.e. methods that, when applied to the solution of the IVP

$$y' = f(t, y), \quad t \in [t_0, T], \quad y(t_0) = y_0 \in \mathbb{R}^m, \quad (1)$$

provide, at the n th step of integration, a discrete problem in the form

$$F(\mathbf{y}_n) \equiv A \otimes I_m \mathbf{y}_n - h_n B \otimes I_m \mathbf{f}_n - \boldsymbol{\eta}_n = \mathbf{0}. \quad (2)$$

In the previous equation, A and B are $r \times r$ nonsingular matrices defining the method, I_m denotes, as usual, the identity matrix of dimension m , h_n is the current stepsize, the block vectors

$$\mathbf{y}_n = (y_{n1}, \dots, y_{nr})^T, \quad \mathbf{f}_n = (f_{n1}, \dots, f_{nr})^T, \quad (3)$$

where

$$y_{nj} \approx y(t_{nj}), \quad f_{nj} = f(t_{nj}, y_{nj}), \quad t_{nj} = t_n + c_j h_n, \quad j = 1, \dots, r, \quad (4)$$

contain the discrete solution, and the vector $\boldsymbol{\eta}_n$ only depends on already known quantities. Instances of methods falling in this class are the majority of implicit Runge-Kutta methods, a number of General Linear methods and, more recently, block BVMs [7]. Under suitable assumptions, such methods can be implemented as *blended methods* [3, 4, 8], thus allowing the definition of efficient nonlinear splittings for solving the corresponding discrete problems.

Blended implicit methods have been recently implemented in the computational code `BiM` [5]. In the two references [5, 6], most of the computational details of this code are described. In particular, in [5] it is mentioned that deferred correction has been used for estimating local truncation errors. Nevertheless, when revising paper [5], we realized that, because of the properties of the methods used in the code, deferred correction allows a noticeable *short cut* in its actual implementation. This, in turn, has allowed us to greatly simplify the data structure of the code itself. The analysis of this short cut is the main concern of this paper. In particular, in Section 2 some preliminary results, concerning the factorization of a Vandermonde matrix, are given in order to obtain the main results on deferred correction stated in Sections 3 and 4. Finally, some concluding remarks are contained in Section 5.

2 Preliminary results

In this section we report some results concerning the factorization of a Vandermonde matrix (actually, its transpose), to be used later: though most of them are known (see, for example, [1, 9, 14, 15]), nevertheless, they are here cast in the most general and appropriate form for subsequent reference.

The matrix which we shall consider is the one defined by the abscissae $\{c_i\}$ defining the block method (2)–(4):

$$V = \begin{pmatrix} 1 & c_1^1 & \dots & c_1^{r-1} \\ \vdots & \vdots & & \vdots \\ 1 & c_r^1 & \dots & c_r^{r-1} \end{pmatrix}. \quad (5)$$

Hereafter, we shall assume

$$0 < c_1 < c_2 < \dots < c_r,$$

so that the matrix V is nonsingular. As an example, for the methods implemented in the code **BiM**, one has: $c_i = i$, $i = 1, \dots, r$.

In order to state the required results, we also need to introduce the following notations:

- \mathbf{e}_i , $i = 1, \dots, r$, is the i th unit vector in \mathbb{R}^r ;
- $\omega_j(x) = \prod_{k=1}^{j-1} (x - c_k)$, $j = 1, \dots, r$, is the j th Newton polynomial defined by the considered abscissae;
- $x^j[c_1, \dots, c_i]$ is the divided difference of the function x^j over the abscissae c_1, \dots, c_i .

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

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

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

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

Lemma 1 *The matrices*

$$L = (w_j(c_i))_{i,j=1,\dots,r}, \quad U = (x^{j-1}[c_1, \dots, c_i])_{i,j=1,\dots,r}, \quad (6)$$

are lower and unit upper triangular, respectively.

Then, the following result follows.

Lemma 2 *Let V, L, U be defined according to (5) and (6). Then,*

$$V = LU. \quad (7)$$

Proof In fact, for all $i, j = 1, \dots, r$, from (6) one has:

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

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, \dots, c_r , evaluated at c_i . \square

Concerning the two factors L and U , the following result holds true (see also [9]).

Lemma 3

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

and $U^{-1} = (u_{ij})$, such that

$$\sum_{i=1}^j u_{ij} x^{i-1} \equiv \omega_j(x), \quad j = 1, \dots, r. \quad (8)$$

From Lemma 3, the following result follows.

Lemma 4 Let $g(x)$ be a given function and let $g_i = g(t_0 + c_i h)$, $i = 1, \dots, r$. Then,

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

Proof From Lemma 3, for all $i = 1, \dots, r$, one obtains that

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

3 Deferred correction for block implicit methods

In this section, we shall use the previous results to obtain a remarkably simple implementation of deferred correction for the block methods implemented in the code BiM [5]. Concerning the latter methods, they are such that (see (2))

$$\begin{aligned}
[\mathbf{a} | A] &\equiv \left(\begin{array}{c|ccc} \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(r)} & \alpha_1^{(r)} & \dots & \alpha_r^{(r)} \end{array} \right), \\
[\mathbf{b} | B] &\equiv \left(\begin{array}{c|ccc} \beta_0^{(1)} & \beta_1^{(1)} & \dots & \beta_r^{(1)} \\ \vdots & \vdots & & \vdots \\ \beta_0^{(r)} & \beta_1^{(r)} & \dots & \beta_r^{(r)} \end{array} \right),
\end{aligned} \tag{9}$$

where the coefficients on the i th row of the two matrices define an r -step LMF of order r , and $\boldsymbol{\eta}_n = -\mathbf{a} \otimes y_n + h_n \mathbf{b} \otimes f_n$. Since A is nonsingular, and taking into account consistency, we can assume, without loss of generality, that

$$A = I_r, \quad \mathbf{a} = -\mathbf{e} \equiv - (1 \dots 1)^T. \tag{10}$$

The order r conditions for the LMF defining each row of the matrices in (9) can then be cast in matrix form, by introducing the matrix

$$D = \text{diag}(c_1 \dots c_r),$$

as follows:

$$\begin{aligned}
D\mathbf{e} - \mathbf{b} - B\mathbf{e} &= \mathbf{0}, & (11) \\
D^i\mathbf{e} - i B D^{i-1}\mathbf{e} &= \mathbf{0}, \quad i = 2, \dots, r. & (12)
\end{aligned}$$

Remark 1 *In general, when the scaling (10) is not considered, the previous equations are, respectively,*

$$A D \mathbf{e} - \mathbf{b} - B \mathbf{e} = \mathbf{0}, \quad A D^i \mathbf{e} - i B D^{i-1} \mathbf{e} = \mathbf{0}, \quad i = 2, \dots, r.$$

We observe that, from (11), the vector \mathbf{b} turns out to be uniquely determined, provided all LMFs are consistent, from the choice of the matrix B . The latter turns out to be uniquely determined by the order conditions (12) and by fixing its spectrum [4, 8]. However, since we are dealing with order r LMFs, one obtains that, for $i = r + 1$, (12) becomes

$$D^{r+1}\mathbf{e} - (r+1) B D^r \mathbf{e} = \mathbf{v}_{r+1} \equiv \begin{pmatrix} v_{r+1,1} \\ \vdots \\ v_{r+1,r} \end{pmatrix}. \tag{13}$$

In more detail, by setting hereafter $c_0 = 0$, the right-hand side of the previous equation reads, component-wise, as follows (see (9), (10) and Remark 1):

$$v_{r+1,k} = \sum_{i=0}^r \left(\alpha_i^{(k)} c_i^{r+1} - (r+1) \beta_i^{(k)} c_i^r \right), \quad k = 1, \dots, r, \tag{14}$$

i.e., $\frac{1}{(r+1)!}v_{r+1,k}$ is the coefficient of the leading term of the truncation error of the LMF defined by the k th rows of the matrices in (9). Then, from (12)-(13) one obtains that (see (5))

$$D^2V - BDVG = \mathbf{v}_{r+1}\mathbf{e}_r^T, \quad \text{where} \quad G = \text{diag}(2 \dots r+1), \quad (15)$$

from which

$$B = (D^2V - \mathbf{v}_{r+1}\mathbf{e}_r^T)G^{-1}V^{-1}D^{-1} \quad (16)$$

follows. Now, in order to apply deferred correction, we need an additional couple of matrices in the form (9), whose rows define r -step LMFs of order (at least) $r+1$ (see, for example, [2, 7]), which are defined over the same set of abscissae $\{c_i\}$. By using the same normalization (10), and denoting by $[\mathbf{b}_1 | B_1]$ the remaining matrix, then the corresponding order conditions are given by:

$$\begin{aligned} D\mathbf{e} - \mathbf{b}_1 - B_1\mathbf{e} &= \mathbf{0}, & (17) \\ D^i\mathbf{e} - iB_1D^{i-1}\mathbf{e} &= \mathbf{0}, \quad i = 2, \dots, r+1. & (18) \end{aligned}$$

Similarly to what seen in (11), now (17) uniquely defines the vector \mathbf{b}_1 , once B_1 is fixed. For the latter matrix, from (18) one readily obtains that

$$B_1 = D^2VG^{-1}V^{-1}D^{-1}, \quad (19)$$

that is, the matrix is uniquely determined by the order conditions. Such matrix can be used for estimating the truncation error of the method defined by (9). Indeed, if we consider the very first application of the method, thus neglecting, for sake of brevity, the index n in (2), we have that the discrete solution satisfies the equation

$$I_r \otimes I_m \mathbf{y} - hB \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h\mathbf{b} \otimes f_0 = \mathbf{0}. \quad (20)$$

Deferred correction is then implemented by *plugging in* the above discrete solution in the discrete problem defined by the block method (17)–(19), thus obtaining (see, for example, [2, 7])

$$I_r \otimes I_m \mathbf{y} - hB_1 \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h\mathbf{b}_1 \otimes f_0 \approx -\boldsymbol{\tau}. \quad (21)$$

In the above equation, $\boldsymbol{\tau}$ is the vector of the (local) truncation errors of the method (9). In more detail (see (13)-(14)), one has:

$$\boldsymbol{\tau} = \frac{h^{r+1}}{(r+1)!} \mathbf{v}_{r+1} \otimes y_0^{(r+1)} + O(h^{r+2}), \quad (22)$$

where $y_0^{(r+1)}$ denotes the $(r+1)$ st derivative of the (local) solution $y(t)$ at t_0 . The following result precisely quantifies the approximation to the truncation error provided by the left-hand side of equation (21).

Theorem 1 (Main Result) Let $g(t)$ be any function such that

$$g(t_0 + c_i h) = f(t_0 + c_i h, y_i), \quad i = 0, \dots, r.$$

Then,

$$\begin{aligned} I_r \otimes I_m \mathbf{y} - h B_1 \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h \mathbf{b}_1 \otimes f_0 &= \\ &= -\frac{h^{r+1}}{r+1} \mathbf{v}_{r+1} \otimes g[t_0 + c_0 h, \dots, t_0 + c_r h]. \end{aligned} \quad (23)$$

Remark 2 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 (1)) $y' = f(t, y)$, it follows that that, under suitable smoothness assumptions for f ,

$$g[t_0 + c_0 h, \dots, t_0 + c_r h] = \frac{1}{r!} y_0^{(r+1)} + O(h).$$

Consequently, (23) provides a $O(h^{r+2})$ approximation to the (opposite of the) leading term at the right-hand side of equation (22).

Proof From equation (20), by setting

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

where

$$f_i = f(t_0 + c_i h, y_i) \equiv g(t_0 + c_i h), \quad i = 0, \dots, r,$$

and taking into account (11)–(19), we obtain:

$$\begin{aligned} I_r \otimes I_m \mathbf{y} - h B_1 \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h \mathbf{b}_1 \otimes f_0 &= \\ &= h([\mathbf{b} | B] - [\mathbf{b}_1 | B_1]) \otimes I_m \hat{\mathbf{f}} = h(B - B_1)[- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -h \mathbf{v}_{r+1} \mathbf{e}_r^T G^{-1} V^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -\frac{h}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T V^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} = (*). \end{aligned}$$

From (6)–(7), property **P2**, Lemmas 3 and 4, and considering that $c_0 = 0$, one then obtains:

$$\begin{aligned} (*) &= -\frac{h}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T U^{-1} L^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -\frac{h}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T L^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -\frac{h}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T L^{-1} \begin{bmatrix} \frac{-1}{c_1} & \frac{1}{c_1} & & \\ \vdots & & \ddots & \\ \frac{-1}{c_r} & & & \frac{1}{c_r} \end{bmatrix} \otimes I_m \hat{\mathbf{f}} \end{aligned}$$

$$\begin{aligned}
&= -\frac{h^2}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T L^{-1} \begin{bmatrix} \frac{-1}{(c_1-c_0)h} & \frac{1}{(c_1-c_0)h} & & \\ \vdots & & \ddots & \\ \frac{-1}{(c_r-c_0)h} & & & \frac{1}{(c_r-c_0)h} \end{bmatrix} \otimes I_m \hat{\mathbf{f}} \\
&= -\frac{h^2}{r+1} \mathbf{v}_{r+1} \mathbf{e}_r^T L^{-1} \otimes I_m \begin{pmatrix} g[t_0 + c_0h, t_0 + c_1h] \\ \vdots \\ g[t_0 + c_0h, t_0 + c_rh] \end{pmatrix} \\
&= -\frac{h^{r+1}}{r+1} \mathbf{v}_{r+1} \left(\left(\prod_{k=2}^r (c_1 - c_k)h \right)^{-1} \dots \left(\prod_{k=1}^{r-1} (c_r - c_k)h \right)^{-1} \right) \otimes I_m \\
&\quad \begin{pmatrix} g[t_0 + c_0h, t_0 + c_1h] \\ \vdots \\ g[t_0 + c_0h, t_0 + c_rh] \end{pmatrix} \\
&= -\frac{h^{r+1}}{r+1} \mathbf{v}_{r+1} \otimes g[t_0 + c_0h, \dots, t_0 + c_rh]. \quad \square
\end{aligned}$$

4 Generalizations

The result of Theorem 1 has been directly used in the actual implementation of the code `BiM` [5], starting from its release 1.1. Nevertheless, it is worth mentioning that this result can be generalized to the case of a general block implicit method, that is when the LMFs in (9) have order $p \leq r$ and those of the block method used for the deferred correction have order $q > p$. In such a case, equation (11) still holds true, whereas (12) holds true for $i = 2, \dots, p \leq r$. Moreover, equation (13) now becomes

$$D^{\nu+1} \mathbf{e} - (\nu+1) B D^\nu \mathbf{e} = \mathbf{v}_{\nu+1} \equiv \begin{pmatrix} v_{\nu+1,1} \\ \vdots \\ v_{\nu+1,r} \end{pmatrix}, \quad \nu = p, \dots, r, \quad (24)$$

where (compare with (14))

$$v_{\nu+1,k} = \sum_{i=0}^r \left(\alpha_i^{(k)} c_i^{\nu+1} - (\nu+1) \beta_i^{(k)} c_i^\nu \right), \quad k = 1, \dots, r, \quad (25)$$

i.e., the vector of the truncation errors of the block method is given by (compare with (22))

$$\boldsymbol{\tau} = \frac{h^{p+1}}{(p+1)!} \mathbf{v}_{p+1} \otimes y_0^{(p+1)} + \frac{h^{p+2}}{(p+2)!} \mathbf{v}_{p+2} \otimes y_0^{(p+2)} + \dots \quad (26)$$

As a consequence, equation (16) now becomes

$$B = \left(D^2 V - \sum_{\nu=p}^r \mathbf{v}_{\nu+1} \mathbf{e}_\nu^T \right) G^{-1} V^{-1} D^{-1}. \quad (27)$$

Similarly, the matrix B_1 of the block method used for the deferred correction, made up of order $q > p$ LMFs, will be given by (compare with (19))

$$B_1 = \left(D^2 V - \sum_{\nu=q}^r \hat{\mathbf{v}}_{\nu+1} \mathbf{e}_\nu^T \right) G^{-1} V^{-1} D^{-1}, \quad (28)$$

with the vectors $\hat{\mathbf{v}}_{\nu+1}$ defined similarly to the vectors $\mathbf{v}_{\nu+1}$. This allows us to generalize the result of Theorem 1 as follows (the notation is the same used in that theorem).

Theorem 2 (Generalization of Theorem 1) *With reference to (27)-(28), let define the vectors*

$$\mathbf{w}_{i+1} = \begin{cases} \mathbf{v}_{i+1}, & \text{if } i < q, \\ \mathbf{v}_{i+1} - \hat{\mathbf{v}}_{i+1}, & \text{if } i \geq q. \end{cases} \quad (29)$$

Then,

$$\begin{aligned} I_r \otimes I_m \mathbf{y} - h B_1 \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h \mathbf{b}_1 \otimes f_0 &= \\ &= -\frac{h^{p+1}}{p+1} \mathbf{v}_{p+1} \otimes g[t_0 + c_0 h, \dots, t_0 + c_p h] \\ &\quad - \sum_{j=p+1}^r h^{j+1} \left(\sum_{i=p}^j \frac{u_{ij}}{i+1} \mathbf{w}_{i+1} \right) \otimes g[t_0 + c_0 h, \dots, t_0 + c_j h], \end{aligned} \quad (30)$$

where the coefficient u_{ij} are defined according to (8).

Proof The proof strictly follows that of Theorem 1, by taking into account (24)–(28):

$$\begin{aligned} I_r \otimes I_m \mathbf{y} - h B_1 \otimes I_m \mathbf{f} - \mathbf{e} \otimes y_0 - h \mathbf{b}_1 \otimes f_0 &= \\ &= h([\mathbf{b} | B] - [\mathbf{b}_1 | B_1]) \otimes I_m \hat{\mathbf{f}} = h(B - B_1)[- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -h \sum_{i=p}^r \mathbf{w}_{i+1} \mathbf{e}_i^T G^{-1} V^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -h \sum_{i=p}^r \frac{1}{i+1} \mathbf{w}_{i+1} \mathbf{e}_i^T U^{-1} L^{-1} D^{-1} [- \mathbf{e} | I_r] \otimes I_m \hat{\mathbf{f}} \\ &= -h^2 \sum_{i=p}^r \frac{1}{i+1} \mathbf{w}_{i+1} \mathbf{e}_i^T U^{-1} L^{-1} \otimes I_m \begin{pmatrix} g[t_0 + c_0 h, t_0 + c_1 h] \\ \vdots \\ g[t_0 + c_0 h, t_0 + c_r h] \end{pmatrix} \\ &= -h^2 \sum_{i=p}^r \frac{1}{i+1} \mathbf{w}_{i+1} \mathbf{e}_i^T U^{-1} \otimes I_m \begin{pmatrix} g[t_0 + c_0 h, t_0 + c_1 h] \\ hg[t_0 + c_0 h, t_0 + c_1 h, t_0 + c_2 h] \\ \vdots \\ h^{r-1} g[t_0 + c_0 h, \dots, t_0 + c_r h] \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= -h^2 \sum_{i=p}^r \frac{1}{i+1} \mathbf{w}_{i+1} \otimes \sum_{j=i}^r u_{ij} h^{j-1} g[t_0 + c_0 h, \dots, t_0 + c_j h] \\
&= - \sum_{j=p}^r h^{j+1} \left(\sum_{i=p}^j \frac{u_{ij}}{i+1} \mathbf{w}_{i+1} \right) \otimes g[t_0 + c_0 h, \dots, t_0 + c_j h].
\end{aligned}$$

The thesis then follows from (29), by considering that $q > p$ and (due to Lemmas 1 and 3) $u_{jj} = 1$ for all j . \square

Remark 3 Evidently, (30) provides a $O(h^{p+2})$ approximation to the (opposite of the) leading term of the truncation error (26), assuming f suitably smooth.

5 Conclusions

In this paper we have proved that deferred correction, when used in connection with block implicit methods defined by order r LMFs, greatly simplifies in its practical implementation. The result of Theorem 1 has been actually used in the computational code BiM [5], allowing to halve, in the practice, the data structure of that code. Indeed, for each method implemented in the code, which is defined by a suitable matrix B , the corresponding matrix B_1 is no longer required to obtain the error estimate via deferred correction, since the vector \mathbf{v}_{r+1} is known and, moreover, we can directly compute the divided difference at the right-hand side of equation (23). In addition to this, the results of Theorems 1 and 2 provide a much better insight into deferred correction, when used with block implicit methods.

Acknowledgements

The authors are very indebted to professor Donato Trigiante, for the useful discussions and his valuable comments.

References

- [1] L. Aceto, D. Trigiante. The Matrices of Pascal and other Greats, *Amer. Math. Monthly* **108** (2001) 232-245.
- [2] L. Brugnano. Boundary Value Methods for the Numerical Solution of Ordinary Differential Equations, *Lecture Notes in Mathematics* **1196** (1997) 78-89.
- [3] L. Brugnano. Blended Block BVMs (B_3 VMs): A Family of Economical Implicit Methods for ODEs, *Jour. Comput. Appl. Math.* **116** (2000) 41-62.
- [4] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs, *Appl. Numer. Math.* **42** (2002) 29-45.

- [5] L. Brugnano, C. Magherini. The BiM Code for the Numerical Solution of ODEs, *Jour. Comput. Appl. Math.* **164-165** (2004) 145–158.
Code web page: <http://math.unifi.it/~brugnano/BiM/index.html>
- [6] L. Brugnano, C. Magherini. Some Linear Algebra Issues Concerning the Implementation of Blended Implicit Methods. *Numer. Lin. Alg. Appl.* (to appear).
- [7] L. Brugnano, D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Gordon and Breach, Amsterdam, 1998.
- [8] L. Brugnano, D. Trigiante. Block Implicit Methods for ODEs, in *Recent Trends in Numerical Analysis*, D. Trigiante ed., Nova Science Publ. Inc., New York, 2001, pp. 81–105.
- [9] J. C. Butcher. A Transformation for the Analysis of DIMSIMs, *BIT* **34** (1994) 25–32,
- [10] J. R. Cash. Iterated Deferred Correction Algorithms for Two-Point BVPs, *WSSIA* **2** (1993) 113–125.
- [11] 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.
- [12] M. Lentini, V. Pereyra. A Variable Order Finite Difference Method for Nonlinear Multipoint Boundary Value Problems, *Math. Comp.* **28** (1974) 981–1024.
- [13] B. Lindberg. Error Estimation and Iterative Improvement for Discretization Algorithms, *BIT* **20** (1980) 486–500.
- [14] H. Oruç, H. K. Akmaz. Symmetric Functions and the Vandermonde Matrix, *Jour. Comput. Appl. Math.* **172** (2004) 49–64.
- [15] H. Oruç, G. M. Phillips. Explicit Factorization of the Vandermonde Matrix, *Lin. Alg. Appl.* **315** (2000) 113–123.
- [16] V. Pereira. On Improving an Approximate Solution of a Functional Equation by Deferred Corrections, *Numer. Math.* **8** (1966) 376–391.
- [17] V. Pereira. Iterated Deferred Corrections for Nonlinear Operator Equations, *Numer. Math.* **10** (1967) 316–323.
- [18] H. J. Stetter. The Defect Correction Principle and Discretization Methods, *Numer. Math.* **29** (1978) 425–443.
- [19] H. J. Stetter. Global Error Estimation in ODE-Solvers, *Lecture Notes in Mathematics* **630** (1978) 245–258.

- [20] R. D. Skeel. A Theoretical Framework for Proving Accuracy Results for Deferred Corrections, *SIAM J. Numer. Anal.* **19** (1981) 171–196.
- [21] R. D. Skeel. Thirteen Ways to Estimate Global Error, *Numer. Math.* **48** (1986) 1–20.
- [22] H. A. Watts, L. F. Shampine. A-stable Block One-step Methods, *BIT* **12** (1972) 252–266.
- [23] P. Zadunaisky. On the Estimation of Errors Propagated in the Numerical Solution of Ordinary Differential Equations, *Numer. Math.* **27** (1976) 21–39.