Pergamon

Computers Math. Applic. Vol. 30, No. 10, pp. 37–54, 1995
Copyright©1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/95 $9.50 + 0.00

0898-1221(95)00155-7

Properties and Implementation of r-Adams Methods Based on Mixed-Type Interpolation

M. Van Daele, G. Vanden Berghe and H. De Meyer*
Vakgroep Toegepaste Wiskunde en Informatica, Universiteit-Gent
Krijgslaan 281–S9, B9000 Gent, Belgium
Marnix.VanDaele@rug.ac.be

(Received July 1994; accepted March 1995)

Abstract—We investigate the properties of the coefficients of modified r-Adams methods for the integration of ODEs. The derivation of these methods is, in contrast with the classical Adams methods, not based on a polynomial interpolation theory, but rather starts from a mixed interpolation theory in which a parameter κ is involved. It will be shown that the coefficients of the modified methods possess properties which make these methods very attractive. Further, we will discuss the role of so-called over-implicit modified r-Adams schemes in the construction of more general linear multistep methods. Our second goal is to show that the modified Adams-Bashforth/Adams-Moulton methods are very well suited to be implemented as a predictor-corrector pair. In particular, we will discuss the choice of the interpolation parameter when such a method is applied to general systems of equations. Numerical tests are performed to support the theory.

Keywords—Numerical integration, Interpolation, Multistep methods.

1. INTRODUCTION

Many methods have been developed in the last several decades for the step-by-step integration of ordinary differential equations. Some of them take advantage of special properties of the ODE solution that may be known in advance. Several exponential fitting methods have thus been developed for problems where the solution has an exponential or an oscillatory character. Early work in this area has been carried out by Gautschi [1], Stiefel and Bettis [2], Bettis [3], and Lyche [4]. In the seventies and the eighties, new methods have been presented by Raptis and Allison [5], Ixaru and Rizea [6], van der Houwen en Sommeijer [7], Neta and Ford [8], Neta [9], Raptis and Cash [10]. More recent work is due to Panovsky and Richardson [11], Coleman [12], Coleman and Booth [13], Simos [14], and some of the present authors [15,16] to name a few.

In these earlier papers [15,16], the construction of new integration methods of Adams-type for first-order differential equations of the form

$$y' = f(x,y), \qquad y(a) = \eta, \tag{1.1}$$

based on a mixed–type interpolation technique [17,18] has been discussed. The mixed interpolation functions are a combination of polynomials up to degree $q-2$, and trigonometric polynomials of first order with respect to a frequency κ (i.e., a linear combination of the functions $\sin \kappa x$ and $\cos \kappa x$; note that in the case that κ is a purely imaginary value, the last two functions are replaced by $e^{\kappa x}$ and $e^{-\kappa x}$). The error being known at interpolation level, the order of the local

*Research Director at the National Fund for Scientific Research (N.F.W.O. Belgium).

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

truncation error of the corresponding difference method can be raised by a suitable choice of the interpolation parameter κ.

In [17,18], it was also shown that in the limit for κ going to 0, this mixed-type interpolation theory results in polynomial interpolation through the same set of points. Consequently, the newly developed integration schemes reduce to the known ones if κ tends to zero.

Since the Adams-Bashforth and Adams-Moulton schemes have attractive properties which assure that they can be implemented very efficiently as a predictor-corrector (PC) pair, these methods, which we will call ABM schemes, are used worldwide. The purpose of this paper is to show that our modified ABM schemes, based on mixed interpolation, satisfy analogous properties and can be implemented nearly as cheaply as the classical schemes.

Secondly, the so-called r-Adams methods are known to be the building blocks to construct families of general linear differential methods. Before examining the implementation of PC-pairs of ABM-type, we will first examine the construction of modified general linear differential methods based on modified r-Adams methods.

In the next section, we start with the introduction of the necessary definitions and notations and the derivation of some general properties.

2. GENERAL PROPERTIES

In [15,16], it was shown that the modified schemes of Adams type can be obtained by replacing $f(x, y(x))$ in the identity

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x))\, dx \tag{2.1}$$

by the mixed interpolation function $I_k^r(x) = I_k^r(x_{n+r} + sh) = \tilde{I}_k^r(s)$ through the $k+1$ equidistant points $x_{n+r} - ih$, $i = 0(1)k$. For $r = 0$ and $r = 1$, one thus obtains the modified Adams-Bashforth and Adams-Moulton formulae. For the time being, r can be any integer.

In [17,18], it was shown that

$$\begin{aligned}\tilde{I}_k^r(s) = &\sum_{i=0}^{k} (-1)^i \binom{-s}{i} \nabla^i f_{n+r} \\ &- \kappa^2 \phi_k((s+k)h) \nabla^{k-1} f_{n+r} - \kappa^2 \phi_{k+1}((s+k)h) \nabla^k f_{n+r},\end{aligned} \tag{2.2}$$

under the condition that $\forall\, l \in \mathbb{Z} : \theta \neq l\pi$ with $\theta := \kappa h$ and where the functions $\phi_n(x)$ are defined in [17]. Moreover, it has been shown [18] that the error related to (2.2) can be written, if $k\kappa h < \pi$, as

$$E_k^r(f, x) = h^{k-1} \phi_k(x - x_{n+r-k}) \left[\kappa^2 f^{(k-1)}(\xi, y(\xi)) + f^{(k+1)}(\xi, y(\xi))\right], \tag{2.3}$$

where $x_{n+r-k} < \xi < x_{n+r}$. Since the first part in the right-hand side of (2.2) corresponds to the polynomial interpolant through the same set of points, one can already deduce that by setting κ equal to 0 in a modified formula, the known classical analogue will be retrieved.

With the general properties concerning the interpolation theory being introduced, we can proceed with the construction of modified Adams methods. Defining

$$\alpha_i^r = (-1)^i \int_0^1 \binom{r-s}{i} ds,$$

$$\varphi_i^l = \int_0^1 \phi_i((s+l)h)\, ds,$$

one obtains, after the introduction of (2.2) in (2.1), that $y(x_{n+1})$ is approximated by y_{n+1} through

$$y_{n+1} - y_n = h \sum_{i=0}^{k} \beta_i^{r,k} \nabla^i f_{n+r}, \tag{2.4}$$

where

$$\begin{aligned} \beta_i^{r,k} &= \alpha_i^r, \qquad i = 0(1)k-2, \\ \beta_{k-1}^{r,k} &= \alpha_{k-1}^r - \kappa^2 \varphi_k^{-r+k}, \\ \beta_k^{r,k} &= \alpha_k^r - \kappa^2 \varphi_{k+1}^{-r+k}. \end{aligned} \tag{2.5}$$

The constants α_i^r are the classical ones which can, e.g., be calculated by means of generating functions (see [19–21]).

As can be seen from (2.5), only the last two coefficients $\beta_{k-1}^{r,k}$ and $\beta_k^{r,k}$ are modified with respect to the polynomial case. This property was already observed by Bettis [3], by means of a completely different theory.

There is also a second interesting remark that can be made about (2.5): if one wishes to find out the classical relation that corresponds to any modified equation, it suffices to set κ (or equivalently θ) equal to 0 and to replace any $\beta_j^{r,k}$-value by the corresponding α_j^r-value.

We now define

$$\rho_{k+1}^r(E) = E^{k+1-r} - E^{k-r}, \qquad k \geq r \geq 0, \tag{2.6}$$

and

$$\sigma_{k+1}^r(E) = \sum_{i=0}^{k} \beta_i^{r,k} \nabla^i E^k, \qquad k \geq 0, \tag{2.7}$$

where E is the shift operator defined by $Ef(x) = f(x+h)$ such that $\nabla = 1 - E^{-1}$. After shifting, (2.4) results in the modified scheme

$$\rho_{k+1}^r(E) y_n = h \sigma_{k+1}^r(E) f_n, \qquad k \geq r \geq 0. \tag{2.8}$$

To obtain an expression for the truncation error associated with the scheme (2.8), one can proceed as follows. Since $\phi_k(x)$ belongs to the space $\{\sin \kappa x, \cos \kappa x, 1, x, \dots, x^{k-1}\}$, which is a $k+2$ dimensional Chebyshev space provided that $0 < (k+1)\kappa h < \pi$, $\phi_k(x)$ has at most $k+1$ distinct zeros in $[0, (k+1)h]$ (see, e.g., [22, Theorem 1.14]). Since $\phi_k(jh) = 0$, $j = 0(1)k$, by construction, $\phi_k(x)$ has a constant sign in each interval $]jh, (j+1)h[$, $j = 0(1)k$. On account of the mean value theorem, the error of the scheme (2.4) can, for $k \geq r \geq 0$, be written as

$$\begin{aligned} y(x_{n+1}) - y_{n+1} &= \int_{x_n}^{x_{n+1}} E_k^r(f, x)\, dx \\ &= h^k \left[\kappa^2 f^{(k-1)}(\eta, y(\eta)) + f^{(k+1)}(\eta, y(\eta))\right] \int_{-r}^{1-r} \phi_k((s+k)h)\, ds \\ &= h^k \varphi_k^{-r+k} \left[\kappa^2 y^{(k)}(\eta) + y^{(k+2)}(\eta)\right], \end{aligned} \tag{2.9}$$

with $x_{n+r-k} < \eta < \max(x_{n+1}, x_{n+r})$.

This expression can be written in a more appropriate way, using the relationships given in [17], that exist between the $\phi_n(x)$ functions. Indeed, a first relationship is

$$\phi_n(x+h) = \phi_n(x) + \phi_{n-1}(x), \qquad n \geq 1.$$

After integration, one thus finds

$$\varphi_n^l = \varphi_n^{l-1} + \varphi_{n-1}^{l-1}, \qquad n \geq 1, \quad \forall l \in \mathbb{Z}. \tag{2.10}$$

The second relation can be written as

$$\kappa^2 \phi_{n-2}(x-h) = 2(1 - \cos\theta) \left[\binom{\frac{x}{h-1}}{n-1} - \kappa^2 \phi_n(x) \right].$$

One then obtains, with $x = (s - r + k)h$ and $n = k + 1$, that

$$\kappa^2 \varphi_{k-1}^{-r+k-1} = 2(1 - \cos\theta)\beta_k^{r,k}, \qquad k \geq 1. \tag{2.11}$$

Using (2.11), one can now rewrite (2.9) in terms of the β-coefficients. One thus finds that the error related to (2.8) is given by

$$y(x_{n+k+1-r}) - y_{n+k+1-r} = \frac{2(1-\cos\theta)}{\theta^2}\beta_{k+1}^{r,k+1} h^{k+2}\left[\kappa^2 y^{(k)}(\eta) + y^{(k+2)}(\eta)\right], \tag{2.12}$$

with $x_n < \eta < \max(x_{n+1+k-r}, x_{n+k})$.

Let C_{p+1} be the error constant of the linear difference scheme based on mixed interpolation where p is the order of the method; i.e., if $\mathcal{L}$ is the linear difference operator associated with the modified method, then

$$\mathcal{L}[z(x); h] = \sum_{q=p+1}^{\infty} C_q h^q \left[\kappa^2 z^{(q-2)}(x) + z^{(q)}(x)\right].$$

Using this notation, we can now formally define the modified k-step r-Adams method as follows:

$$\begin{aligned}
&r = 0, \quad \rho_k^0(E)y_n = h\sigma_k^0(E)f_n, \quad k \geq 1, \quad p = k, \quad C_{p+1} = \frac{2(1-\cos\theta)}{\theta^2}\beta_k^{0,k},\\
&r \geq 1, \quad \rho_{k+1}^r(E)y_n = h\sigma_{k+1}^r(E)f_n, \quad k \geq r, \quad p = k+1, \quad C_{p+1} = \frac{2(1-\cos\theta)}{\theta^2}\beta_{k+1}^{r,k+1}.
\end{aligned}$$

Taking into account the explicit form of the trunction error associated with the k-step r-Adams method, it is obvious how to choose the interpolation parameter κ in order to raise the order of the method. Indeed, attributing to κ in each integration step the value for which

$$\kappa^2 y^{(k+\epsilon-1)}(\eta) + y^{(k+\epsilon+1)}(\eta) = 0, \qquad \epsilon = \begin{cases} 0, & r = 0,\\ 1, & r \neq 0, \end{cases}$$

the principal term of the truncation error vanishes.

However, in practice, one can only try to approximate this value: first of all, the higher order derivatives have to be re-expressed using the differential equation in terms of x and $y(x)$. This, of course, requires that f is given in an analytical form. Secondly, one needs to calculate $y(\eta)$ where η is an unknown interior value. Since previous integration steps resulted in approximations of $y(x)$ in knot points x_i in the neighbourhood of η, one will attribute to κ the value for which

$$\kappa^2 y^{(k+\epsilon-1)}(x_j) + y^{(k+\epsilon+1)}(x_j) = 0, \qquad \epsilon = \begin{cases} 0, & r = 0,\\ 1, & r \neq 0, \end{cases}$$

for some knot point x_j, (e.g., if an approximation for y_{n+1} is wanted, then for small k, the choice $j = n$ may be used). In any case, the order of the method is raised by at least one unit, as has already been shown elsewhere (see [15,16]).

Now that we have introduced our modified methods, we will more closely look at the β-coefficients that appear in the right-hand side of these methods.

Although our aim is to discuss the implementation of PC-pairs of modified Adams-Bashforth, Adams-Moulton type, i.e., 0-Adams and 1-Adams schemes, we first derive some general properties.

3. PROPERTIES OF THE COEFFICIENTS OF r-ADAMS METHODS

In the following, we use the standard notation δ_{ij}, which is 1 for $i = j$ and 0 otherwise.

LEMMA 1. *For* $k \geq 1$,

1. $\beta_j^{r-1,k} = \sum\limits_{i=0}^{j} \beta_i^{r,k} - 2(1-\cos\theta)\delta_{j,k-1}\beta_k^{r-1,k}, \qquad j = 0(1)k,$
2. $\beta_j^{r,k} - \beta_j^{r,k-1} = 2(1-\cos\theta)(\delta_{j,k-2} - \delta_{j,k-1})\beta_k^{r,k}, \qquad j = 0(1)k-1,$
3. $\beta_j^{r-1,k} - \beta_j^{r,k} = \beta_{j-1}^{r-1,k-1}, \qquad j = 1(1)k,$

   $\beta_0^{r,k} - \beta_0^{r-1,k} = 2(1-\cos\theta)\left(\beta_1^{r,1}\delta_{0,k} + \beta_1^{r-1,1}\delta_{1,k}\right),$
4. $\beta_j^{r-1,k} - \beta_{j-1}^{r-1,k} = \beta_j^{r,k} + 2(1-\cos\theta)(\delta_{j,k} - \delta_{j,k-1})\beta_k^{r-1,k}, \qquad j = 1(1)k.$

PROOF. The proofs of Lemma 1.1–Lemma 1.4 follow from (2.5), (2.10), and (2.11) and the following properties of the classical coefficients α_i^r:

$$\alpha_j^{r-1} = \sum_{i=0}^{j} \alpha_i^r,$$
$$\alpha_j^{r-1} - \alpha_j^r = \alpha_{j-1}^{r-1}, \qquad j \geq 1.$$

For more details see, for instance, [21]. ∎

LEMMA 2. *For* $k \geq 2$,

1. $\sigma_{k+1}^r(\xi) = \xi\sigma_k^r(\xi) + \beta_k^{r,k}(\xi-1)^{k-2}(\xi^2 - 2\xi\cos\theta + 1),$
2. $\sigma_{k+1}^{r-1}(\xi) = \sigma_{k+1}^r(\xi) + (\xi-1)\sigma_k^{r-1}(\xi)$,
3. $\beta_{k-1}^{r-1,k-1}\sigma_{k+1}^r(\xi) = \beta_k^{r-1,k}\xi\sigma_k^r(\xi) - \beta_k^{r,k}\sigma_k^{r-1}(\xi),$
4. $\xi\sigma_k^r(\xi) - \sigma_k^{r-1}(\xi) = \beta_{k-1}^{r-1,k-1}(\xi-1)^{k-2}(\xi^2 - 2\xi\cos\theta + 1).$

PROOF. For Lemma 2.1, we use (2.7) to obtain for $k \geq 1$

$$\sigma_{k+1}^r(E) - E\sigma_k^r(E) = \sum_{i=0}^{k} \beta_i^{r,k}(E-1)^i E^{k-i} - \sum_{i=0}^{k-1} \beta_i^{r,k-1}(E-1)^i E^{k-i},$$

which, due to Lemma 1.2, yields for $k \geq 2$

$$\begin{aligned}\sigma_{k+1}^r(E) - E\sigma_k^r(E) &= \beta_k^{r,k}\left[(E-1)^k + 2(1-\cos\theta)((E-1)^{k-2}E^2 - (E-1)^{k-1}E)\right]\\ &= \beta_k^{r,k}(E-1)^{k-2}(E^2 - 2E\cos\theta + 1).\end{aligned} \tag{3.1}$$

The proof of Lemma 2.2 makes use of Lemma 1.3. One easily obtains for $k \geq 2$ that

$$\begin{aligned}\sigma_{k+1}^r(E) - \sigma_{k+1}^{r-1}(E) &= \sum_{j=0}^{k}\left(\beta_j^{r,k} - \beta_j^{r-1,k}\right)\nabla^j E^k\\ &= \sum_{j=0}^{k-1}\beta_j^{r-1,k-1}\nabla^j E^{k-1}(E-1)\\ &= (E-1)\sigma_k^{r-1}(E).\end{aligned}$$

To prove Lemma 2.3, one can start from (3.1) with $r = r_0$ and $r = r_0 - 1$. One then obtains for $k \geq 2$

$$\beta_k^{r_0-1,k}\left(\sigma_{k+1}^{r_0}(\xi) - \xi\sigma_k^{r_0}(\xi)\right) = \beta_k^{r_0,k}\left(\sigma_{k+1}^{r_0-1}(\xi) - \xi\sigma_k^{r_0-1}(\xi)\right),$$

while Lemma 2.2 states that

$$\sigma_{k+1}^{r_0-1}(\xi) - \xi\sigma_k^{r_0-1}(\xi) = \sigma_{k+1}^{r_0}(\xi) - \sigma_k^{r_0-1}(\xi).$$

Consequently,

$$\beta_k^{r_0-1,k}\left(\sigma_{k+1}^{r_0}(\xi) - \xi\sigma_k^{r_0}(\xi)\right) = \beta_k^{r_0,k}\left(\sigma_{k+1}^{r_0}(\xi) - \sigma_k^{r_0-1}(\xi)\right),$$

which proves Lemma 2.3, since $\beta_k^{r-1,k} - \beta_k^{r,k} = \beta_{k-1}^{r-1,k-1}$.

To prove Lemma 2.4, we substitute the right-hand side of Lemma 2.1 in the left-hand side of the expression for Lemma 2.3. One obtains for $k \geq 2$

$$\beta_{k-1}^{r-1,k-1}\left[\xi\sigma_k^r(\xi) + \beta_k^{r,k}(\xi-1)^{k-2}(\xi^2 - 2\xi\cos\theta + 1)\right] = \beta_k^{r-1,k}\xi\sigma_k^r(\xi) - \beta_k^{r,k}\sigma_k^{r-1}(\xi).$$

Again, the lemma follows from $\beta_k^{r-1,k} - \beta_k^{r,k} = \beta_{k-1}^{r-1,k-1}$. ∎

The properties listed in Lemma 2 give effective means to generate the $\sigma_{k+1}(\xi)$ functions recursively. As in Lemma 1, these properties are modifications of classical ones which can be retrieved by setting θ equal to 0, and replacing $\beta_j^{r,k}$ by α_j^r.

For $r > 1$, over-implicit schemes are constructed. In [21], it is shown that these r-Adams schemes for $k \geq r$ are very useful if one wishes to build more general difference schemes. Indeed, as is the case with polynomial r-Adams methods, one is able to construct two families of linear k-step methods with $k-1$ parameters $A_r, r = 1(1)k-1$, of the form

$$\sum_{r=0}^{k-1} A_r \nabla y_{n+1-r} = h\sum_{r=0}^{k-1} A_r \sum_{i=0}^{k-1+\epsilon} \beta_i^{r+\epsilon,k-1+\epsilon}\nabla^i f_{n+\epsilon}, \qquad A_0 := 1. \tag{3.2}$$

For $\epsilon = 0$, this family consists of explicit schemes, for $\epsilon = 1$ implicit schemes are found. In general, the constants A_r should be chosen such that the resulting scheme satisfies at least minimal stability requirements. To ascertain zero-stability, e.g., one finds that since

$$\begin{aligned}
&\rho(\xi) = (\xi - 1)\left(\xi^{k-1} + a_1\xi^{k-2} + \cdots + a_{k-1}\right) \text{ with} \\
&\quad a_i = A_i - A_{i-1}, \qquad i = 1(1)k-1, \text{ the polynomial} \\
&\hat{\rho}(\xi) = \left(\xi^{k-1} + a_1\xi^{k-2} + \cdots + a_{k-1}\right)
\end{aligned}$$

should have all its roots inside or on the unit circle, no multiple roots on the unit circle and no root at $+1$.

Due to (2.12), one also finds

$$y(x_{n+1}) - y_{n+1} = \frac{2(1-\cos\theta)}{\theta^2}\sum_{r=0}^{k-1} A_r\beta_{k+\epsilon}^{r+\epsilon,k+\epsilon}h^{k+\epsilon+1}\left[\kappa^2 y^{(k+\epsilon-1)}(\eta_r) + y^{(k+\epsilon+1)}(\eta_r)\right], \tag{3.3}$$

where $x_{n+\epsilon-k} < \eta_r < \max(x_{n+1-r}, x_{n+\epsilon})$. Developing the derivatives in the right-hand side in a Taylor series around x_n, one finds that the order of (3.2) is $k+\epsilon$ and the error constant is

$$C_{k+\epsilon+1} = \frac{2(1-\cos\theta)}{\theta^2}\sum_{r=0}^{k-1} A_r\beta_{k+\epsilon}^{r+\epsilon,k+\epsilon}. \tag{3.4}$$

Again, the order can be raised by a suitable choice of the interpolation parameter κ, namely by setting

$$\kappa^2 y^{(k-1+\epsilon)}(x_j) + y^{(k+1+\epsilon)}(x_j) = 0,$$

for some knot point x_j.

Table 1. The elements of $\mathcal{A}_2$.

$\beta_j^{r,2}$	$j=0$	$j=1$	$j=2$
$r=0$	1	$-1+\frac{\sin(3\theta/2)}{\theta\cos(\theta/2)}$	$\frac{1}{2(1-\cos\theta)}-\frac{\cos\theta}{\theta\sin\theta}$
$r=1$	1	$-1+\frac{\sin(\theta/2)}{\theta\cos(\theta/2)}$	$\frac{1}{2(1-\cos\theta)}-\frac{1}{\theta\sin\theta}$
$r=2$	1	$-1-\frac{\sin(\theta/2)}{\theta\cos(\theta/2)}$	$\frac{1}{2(1-\cos\theta)}-\frac{\cos\theta}{\theta\sin\theta}$

Table 2. The elements of $\mathcal{A}_3$.

$\beta_j^{r,3}$	$j=0$	$j=1$	$j=2$	$j=3$
$r=0$	1	$\frac{1}{2}$	$-\frac{3}{2}+\frac{1}{2(1-\cos\theta)}-\frac{\cos 2\theta}{\theta\sin\theta}$	$\frac{3}{4(1-\cos\theta)}-\frac{\sin(3\theta/2)}{2\theta\sin(\theta/2)\sin\theta}$
$r=1$	1	$-\frac{1}{2}$	$-\frac{1}{2}+\frac{1}{2(1-\cos\theta)}-\frac{\cos\theta}{\theta\sin\theta}$	$\frac{1}{4(1-\cos\theta)}-\frac{1}{2\theta\sin\theta}$
$r=2$	1	$-\frac{3}{2}$	$\frac{1}{2}+\frac{1}{2(1-\cos\theta)}-\frac{1}{\theta\sin\theta}$	$-\frac{1}{4(1-\cos\theta)}+\frac{1}{2\theta\sin\theta}$
$r=3$	1	$-\frac{5}{2}$	$\frac{3}{2}+\frac{1}{2(1-\cos\theta)}-\frac{\cos\theta}{\theta\sin\theta}$	$-\frac{3}{4(1-\cos\theta)}+\frac{\sin(3\theta/2)}{2\theta\sin(\theta/2)\sin\theta}$

For each k-value, both families in (3.2) depend on a set of values $\{\beta_j^{r,k}\}$ $(0 \le j \le k,\ 0 \le r \le k)$ which can be written down in a two-dimensional array $\mathcal{A}_k$. For $k=2$ and $k=3$, the arrays are given in Table 1 and Table 2, respectively.

Apart from the properties listed in Lemma 1, one also finds that due to Lemma 1.3, the columns of $\mathcal{A}_k$ possess a fair amount of structure.

LEMMA 3.

1. $\beta_l^{r,k} = \sum\limits_{i=0}^{j}(-1)^i\binom{j}{i}\beta_{j+l}^{r+i,k+j}, \qquad 0 \le l \le k,$

2. $\beta_j^{j-r,k} = (-1)^j[\beta_j^{r,k} + 2(1-\cos\theta)\delta_{j,k-1}(\beta_k^{r,k}+\beta_k^{r+1,k})], \qquad 0 \le r \le j \le k.$

PROOF. The first relation can easily be proved by means of induction on j. Clearly, equality holds for $j=0$ and $j=1$. Suppose that equality holds for $j=j_0$, one then finds using Lemma 1.3 and Pascal's identity,

$$\begin{aligned}
\beta_l^{r,k} &= \sum_{i=0}^{j_0}(-1)^i\binom{j_0}{i}\left(\beta_{j_0+1+l}^{r+i,k+j_0+1} - \beta_{j_0+1+l}^{r+i+1,k+j_0+1}\right)\\
&= \beta_{j_0+1+l}^{r,k+j_0+1} + \sum_{i=1}^{j_0}(-1)^i\left[\binom{j_0}{i}+\binom{j_0}{i-1}\right]\beta_{j_0+1+l}^{r+i,k+j_0+1} + (-1)^{j_0+1}\beta_{j_0+1+l}^{r+j_0+1,k+j_0+1}\\
&= \sum_{i=0}^{j_0+1}(-1)^i\binom{j_0+1}{i}\beta_{j_0+1+l}^{r+i,k+j_0+1},
\end{aligned}$$

which proves the validity of Lemma 3.1 for $j=j_0+1$.

To prove Lemma 3.2, we will first show that for all $r \in \{0,1,\dots,j\}$

$$\alpha_j^{j-r} = (-1)^j\alpha_j^r. \tag{3.5}$$

Indeed,

$$\alpha_j^{j-r} = (-1)^j\int_0^1\binom{j-r-s}{j}\,ds = \int_0^1\binom{r+s-1}{j}\,ds = \int_0^1\binom{r-t}{j}\,dt = (-1)^j\alpha_j^r.$$

Since $\beta_j^{r,k} = \alpha_j^r$, $j=0(1)k-2$, we now only have to prove Lemma 3.2 for $j=k-1$ and $j=k$. To that aim, we will use the property (see, e.g., [18])

$$\phi_n\left(\frac{nh}{2}-x\right) = (-1)^{n+1}\phi_n\left(\frac{nh}{2}+x\right), \qquad n>0. \tag{3.6}$$

Using (3.6), one finds that

$$\phi_{j+1}((s-j+r+k)h) = (-1)^j \phi_{j+1}((2j-k-r+1-s)h),$$

so that for $j = k-1$ or $j = k$

$$\begin{aligned}
\beta_j^{j-r,k} &= \alpha_j^{j-r} - \kappa^2 \varphi_{j+1}^{-j+r+k} \\
&= (-1)^j \alpha_j^r - \kappa^2 \int_0^1 \phi_{j+1}((s-j+r+k)h)\, ds \\
&= (-1)^j \left[\alpha_j^r - \kappa^2 \int_0^1 \phi_{j+1}((2j-k-r+s)\, ds\right] \\
&= (-1)^j \left[\alpha_j^r - \kappa^2 \varphi_{j+1}^{2j-k-r}\right].
\end{aligned}$$

For $j = k$, one finds

$$\beta_k^{k-r,k} = (-1)^k \left[\alpha_k^r - \kappa^2 \varphi_{k+1}^{-r+k}\right] = (-1)^k \beta_{k+1}^{r,k},$$

while $j = k-1$ results by using (2.5), (2.11), and Lemma 1.3 in

$$\begin{aligned}
\beta_{k-1}^{k-1-r,k} &= (-1)^{k-1} \left[\alpha_{k-1}^r - \kappa^2 \varphi_k^{-r+k-2}\right] \\
&= (-1)^{k-1} \left[\beta_{k-1}^{r,k} + \kappa^2 \left(\varphi_k^{-r+k} - \varphi_k^{-(r+2)+k}\right)\right] \\
&= (-1)^{k-1} \left[\beta_{k-1}^{r,k} + 2(1-\cos\theta) \left(\beta_{k+1}^{r,k+1} - \beta_{k+1}^{r+2,k+1}\right)\right] \\
&= (-1)^{k-1} \left[\beta_{k-1}^{r,k} + 2(1-\cos\theta) \left(\beta_k^{r,k} + \beta_k^{r+1,k}\right)\right].
\end{aligned}$$

This completes the proof of Lemma 3. ∎

Lemma 3.2 explains why, in some cases, the order of a method turns out to be higher than expected according to (3.3). Consider, e.g., the case $k = 2$, $\epsilon = 1$, $A_0 = A_1 = 1$. The corresponding implicit method is given by

$$y_{n+1} - y_{n-1} = h \left[\frac{\theta - \sin\theta}{\theta(1-\cos\theta)}(f_{n-1} + f_{n+1}) + \frac{2(\sin\theta - \theta\cos\theta)}{\theta(1-\cos\theta)} f_n\right]. \tag{3.7}$$

The error constant, computed according to (3.4), vanishes due to Lemma 3.2. However, setting $k = 3$, $A_0 = A_1 = 1$, and $A_2 = 0$, the same (fourth order) method is found and from (3.4), one finds that the error constant is given by

$$C_5 = \frac{2(1-\cos\theta)}{\theta^2}\left(\beta_4^{1,4} + \beta_4^{2,4}\right) = \frac{3\sin\theta - \theta\cos\theta - 2\theta}{3\theta^3(1-\cos\theta)}. \tag{3.8}$$

The results (3.7) and (3.8) have already been found earlier. Indeed, in [15,16], they have been found to constitute the modified Simpson rule.

4. IMPLEMENTATION OF ABM METHODS AS PC-PAIRS

Most modern predictor-corrector methods for nonstiff problems use Adams-Bashforth methods as predictors and Adams-Moulton methods as correctors (the $r = 1$ and $r = 0$ cases of the previous paragraph). One of the main reasons in favour of this class of methods is that they can be implemented very efficiently as is explained by Lambert in [21]. In this section, we will have a closer look at the implementation of the corresponding modified methods.

However, before discussing the implementation of ABM-methods, let us first of all look at the use of general modified linear multistep methods in $\mathrm{P(EC)^{\mu}E^{1-t}}$ mode ($t \in \{0,1\}$), where P, E, and C stand for the application of the predictor, the evaluation of the right-hand side of the differential equation and the application of the corrector, respectively.

Let

$$\begin{aligned} \text{P:}\quad & \sum_{j=0}^{k} a_j^* y_{n+j} = h \sum_{j=0}^{k-1} b_j^* f_{n+j}, \\ \text{C:}\quad & \sum_{j=0}^{k} a_j y_{n+j} = h \sum_{j=0}^{k} b_j f_{n+j}, \end{aligned} \tag{4.1}$$

then $\text{P(EC)}^{\mu}\text{E}^{1-t}$ mode gives

$$\begin{aligned} \text{P:}\quad & y_{n+k}^{[0]} + \sum_{j=0}^{k-1} a_j^* y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} b_j^* f_{n+j}^{[\mu-t]}, \\ \text{(EC)}^{\mu}\text{:}\quad & f_{n+k}^{[\nu]} = f\left(x_{n+k}, y_{n+k}^{[\nu]}\right), \\ & y_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} a_j y_{n+j}^{[\mu]} = h b_k f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} b_j f_{n+j}^{[\mu-t]}, \qquad \nu = 0(1)\mu - 1, \\ \text{E}^{1-t}\text{:}\quad & f_{n+k}^{[\mu]} = f\left(x_{n+k}, y_{n+k}^{[\mu]}\right), \qquad \text{if } t = 0. \end{aligned} \tag{4.2}$$

In the above scheme (4.2), no distinction has been made between scalar problems or systems of equations. However, it should be understood that when applying (4.2) to an m-dimensional system

$${}^{i}y' = {}^{i}f(x,\ {}^{1}y, \dots,\ {}^{m}y), \qquad i = 1(1)m,$$

not only the y_{n+j} and $f_{n+j}^{[\nu]}$ values have become vectors, but also the a_j, a_j^*, b_j, and b_j^* values. Indeed, instead of starting with one parameter κ, we start with a set of m interpolation parameters ${}^{i}\kappa$, one for each of the m equations. To each of these parameters, one can then associate the linear difference operators ${}^{i}\mathcal{L}^*$ and ${}^{i}\mathcal{L}$ given by

$$\begin{aligned} {}^{i}\mathcal{L}^*[z(x); h] &= {}^{i}C_{p^*+1}^* h^{p^*+1} \left[{}^{i}\kappa^2 z^{(p^*-1)}(x) + z^{(p^*+1)}(x)\right] + \mathcal{O}\left(h^{p^*+2}\right), \\ {}^{i}\mathcal{L}[z(x); h] &= {}^{i}C_{p+1} h^{p+1} \left[{}^{i}\kappa^2 z^{(p-1)}(x) + z^{(p+1)}(x)\right] + \mathcal{O}\left(h^{p+2}\right). \end{aligned}$$

Each of the ${}^{i}\mathcal{L}^*$ has order p^* and each of the ${}^{i}\mathcal{L}$ has order p.

Making the localizing assumption that $y_{n+j}^{[\nu]} = y_{n+j}, j = 0(1)k - 1$ and indicating by $\tilde{y}_{n+k}^{[\nu]}$ the approximation to y at x_{n+k} generated under this assumption, one finds for every component ${}^{i}y$, $i = 1(1)m$

$${}^{i}y(x_{n+k}) - {}^{i}\tilde{y}_{n+k}^{[0]} = {}^{i}C_{p^*+1}^* h^{p^*+1} \left[{}^{i}\kappa^2\ {}^{i}y^{(p^*-1)}(x_n) + {}^{i}y^{(p^*+1)}(x_n)\right] + \mathcal{O}\left(h^{p*+2}\right),$$

and

$$\begin{aligned} {}^{i}y(x_{n+k}) - {}^{i}\tilde{y}_{n+k}^{[\nu+1]} &= h \sum_{j=1}^{m} {}^{i}b_k \frac{\partial\ {}^{i}f}{\partial {}^{j}y} \left(x_{n+k},\ {}^{1}y_{\nu_1}, \dots,\ {}^{m}y_{\nu_m}\right) \left[{}^{j}y(x_{n+k}) - {}^{j}\tilde{y}_{n+k}^{[\nu]}\right] + {}^{i}C_{p+1} h^{p+1} \\ &\quad \times \left[{}^{i}\kappa^2\ {}^{i}y^{(p-1)}(x_n) + {}^{i}y^{(p+1)}(x_n)\right] + \mathcal{O}(h^{p+2}), \qquad \nu = 0(1)\mu - 1, \end{aligned} \tag{4.3}$$

where ${}^{j}y_{\nu_j}$ is some value between ${}^{j}y(x_{n+k})$ and ${}^{j}\tilde{y}_{n+k}^{[\nu]}$.

It is then straightforward to show that if $p^* \geq p$ (or $p^* < p$ and $\mu > p - p^*$), the PC method and the corrector have the same order and the same principal local truncation error (PLTE), i.e., for $i = 1(1)m$

$${}^{i}y(x_{n+k}) - {}^{i}y_{n+k}^{[\mu]} = {}^{i}C_{p+1} h^{p+1} \left[{}^{i}\kappa^2\ {}^{i}y^{(p-1)}(x_n) + {}^{i}y^{(p+1)}(x_n)\right] + \mathcal{O}\left(h^{p+2}\right).$$

It thus becomes clear how to choose the interpolation parameters ${}^i\kappa^2$ in order to raise the order of the method: if the PC-pair corresponding to ${}^i\mathcal{L}^*$ and ${}^i\mathcal{L}$ for $i = 1(1)m$ is applied in order to obtain an approximation for ${}^iy_{n+k}$, $i = 1(1)m$ where $p^* \geq p$ (or $p^* < p$ and $\mu > p - p^*$), the choice

$$ {}^i\kappa^2 = -\frac{{}^iy^{(p+1)}(x_n)}{{}^iy^{(p-1)}(x_n)} \tag{4.4} $$

will raise the order of the method to $p+1$. Again, we want to stress the fact that the derivatives can be calculated in any knot point x_n. If occasionally ${}^iy(x_n)^{(p-1)} \approx 0$, ${}^i\kappa^2$ will be attributed a very large (positive or negative) value. To prevent a significant loss of accuracy in the solution ${}^iy_{n+k}$, a good strategy may be to set ${}^i\kappa$ to zero in that step, i.e., to use the known polynomial method.

The condition $p^* \geq p$ (or $p^* < p$ and $\mu > p - p^*$) is essential for the choice (4.4). If this condition is not fulfilled, one finds that, in general, a system of transcendental equations in ${}^j\kappa$, $j = 1(1)m$ has to be solved. This situation should, therefore, be avoided. In the rest of this paper, we will always assume that this is the case. Also, the choice (4.4) means that each ${}^i\kappa^2$ can be calculated from the knowledge of the component ${}^iy$ and its derivatives in x_n solely, i.e., only one component of the solution is of importance. Therefore, whenever possible we will drop the superscript in the notation for the rest of the article.

If $p^* = p$, the classical schemes allow an estimation for the PLTE by means of Milne's device. The modified schemes allow the same technique from

$$ C^*_{p+1}h^{p+1}\left[\kappa^2 y^{(p-1)}(x_n) + y^{(p+1)}(x_n)\right] = y(x_{n+k}) - \tilde{y}^{[0]}_{n+k} + \mathcal{O}\left(h^{p+2}\right), $$

and

$$ C_{p+1}h^{p+1}\left[\kappa^2 y^{(p-1)}(x_n) + y^{(p+1)}(x_n)\right] = y(x_{n+k}) - \tilde{y}^{[\mu]}_{n+k} + \mathcal{O}\left(h^{p+2}\right), $$

one finds after subtraction and neglecting higher order terms that

$$ \text{PLTE} = C_{p+1}h^{p+1}\left[\kappa^2 y^{(p-1)}(x_n) + y^{(p+1)}(x_n)\right] = W\left(y^{[\mu]}_{n+k} - \tilde{y}^{[0]}_{n+k}\right), $$

with

$$ W = \frac{C_{p+1}}{C^*_{p+1} - C_{p+1}}, \tag{4.5} $$

i.e., the Milne estimate is obtained from the same formula as in the classical case. Since the choice (4.4) would mean that this estimate is zero, a different choice for κ^2 will be needed. When Milne's device is used to perform local extrapolation (L) in a P(ECL)$^\mu$E^{1-t}, ($t \in \{0,1\}$), the choice

$$ \kappa^2 = -\frac{y^{(p+2)}(x_n)}{y^{(p)}(x_n)} \tag{4.6} $$

is proposed, since C$_{(p)}$L≡C$_{(p+1)}$ where p is the order of the corrector.

P(ECL)$^\mu$E^{1-t}:

$$ \begin{array}{lll} \text{P:} & y^{[0]}_{n+k} + \displaystyle\sum_{j=0}^{k-1} a^*_j y^{[\mu]}_{n+j} = h\sum_{j=0}^{k-1} b^*_j f^{[\mu-t]}_{n+j}, & \\ (\text{ECL})^\mu: & f^{[\nu]}_{n+k} = f(x_{n+k}, y^{[\nu]}_{n+k}), & \\ & \hat{y}^{[\nu+1]}_{n+k} + \displaystyle\sum_{j=0}^{k-1} a_j y^{[\mu]}_{n+j} = hb_k f^{[\nu]}_{n+k} + h\sum_{j=0}^{k-1} b_j f^{[\mu-t]}_{n+j}, & \nu = 0(1)\mu - 1, \\ & y^{[\nu+1]}_{n+k} = (1+W)\hat{y}^{[\nu+1]}_{n+k} - Wy^{[0]}_{n+k}, & \\ \text{E}^{1-t}: & f^{[\mu]}_{n+k} = f(x_{n+k}, y^{[\mu]}_{n+k}), & \text{if } t = 0. \end{array} \tag{4.7} $$

As was mentioned in the beginning of this section, the preceding algorithms can be written in a computationally convenient and economical form if the PC-pair is a modified ABM-method. Indeed, these methods, when expressed in backward difference form, possess simple and attractive structures which can be fully exploited in the framework of PC-methods. Consider, e.g., the PC-pair

$$\begin{aligned}
&\text{P:} \qquad y_{n+1} - y_n = h\sum_{i=0}^{k-1} \beta_i^{0,k-1}\nabla^i f_n, \quad p^* = k, \quad C^*_{p^*+1} = \frac{2(1-\cos\theta)}{\theta^2}\beta_k^{0,k},\\
&\text{C:} \qquad y_{n+1} - y_n = h\sum_{i=0}^{k-1} \beta_i^{1,k-1}\nabla^i f_{n+1}, \quad p = k, \quad C_{p+1} = \frac{2(1-\cos\theta)}{\theta^2}\beta_k^{1,k}.
\end{aligned}$$

Let ${\nabla^i}_l$ be defined as

$$\nabla_l^i f_p = \begin{cases} \nabla^i f_p^{[\mu]}, & p \neq n+1,\\ \nabla^i f_{n+1}^{[\mu]} + f_{n+1}^{[l]} - f_{n+1}^{[\mu]}, & p = n+1, \end{cases}$$

one can then, for instance, write the $\mathrm{P(EC)}^\mu\mathrm{E}^{1-t}$ algorithm as

$$\begin{aligned}
&\text{P:} && y_{n+1}^{[0]} = y_n^{[\mu]} + h\sum_{i=0}^{k-1} \beta_i^{0,k-1}\nabla^i f_n^{[\mu-t]}, &&\\
&(\text{EC})^\mu\text{:} && f_{n+1}^{[\nu]} = f(x_{n+1}, y_{n+1}^{[\nu]}), &&\\
& && && \nu = 0(1)\mu-1, \qquad (4.8)\\
& && y_{n+1}^{[\nu+1]} = y_n^{[\nu]} + h\sum_{i=0}^{k-1} \beta_i^{1,k-1}\nabla_\nu^i f_{n+1}^{[\mu-t]}, &&\\
&\mathrm{E}^{1-t}\text{:} && f_{n+1}^{[\mu]} = f(x_{n+1}, y_{n+1}^{[\mu]}), && \text{if } t = 0.
\end{aligned}$$

The computational effort of this algorithm (computation and storage of $\nabla_\nu^i f_{n+1}^{[\mu-t]}$, $i = 0(1)k-1$ for every $\nu \in \{0, 1, \dots, k-1\}$) can be reduced to the computation of just one such difference if we make use of the following lemma.

LEMMA 4. *For* $k \geq 2$,

1. $\displaystyle\sum_{j=0}^{k-1} \left(\beta_j^{r,k-1}\nabla^j f_{n+1} - \beta_j^{r-1,k-1}\nabla^j f_n\right) = \beta_{k-1}^{r-1,k-1}\left(\nabla^k f_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f_n\right),$

2. $\displaystyle\sum_{j=0}^{k} \beta_j^{r,k}\nabla^j f_{n+1} - \sum_{j=0}^{k-1} \beta_j^{r-1,k-1}\nabla^j f_n = \beta_k^{r-1,k}\left(\nabla^{k+1} f_{n+1} + 2(1-\cos\theta)\nabla^{k-1} f_n\right).$

PROOF. We rewrite the left-hand side of the first equation as $S(\nabla)f_{n+1}$ where

$$\begin{aligned}
S(\nabla) &:= \sum_{j=0}^{k-1}\left(\beta_j^{r,k-1}\nabla^j - \beta_j^{r-1,k-1}\nabla^j(1-\nabla)\right)\\
&= \beta_0^{r,k-1} - \beta_0^{r-1,k-1} + \sum_{j=1}^{k-1}\left(\beta_j^{r,k-1} - \beta_j^{r-1,k-1}\right)\nabla^j + \sum_{j=0}^{k-1}\beta_j^{r-1,k-1}\nabla^{j+1},
\end{aligned}$$

so that for $k \geq 2$, due to Lemma 1.3, Lemma 1.2, and Lemma 1.3, respectively,

$$\begin{aligned}
S(\nabla) &= \beta_0^{r,k-1} - \beta_0^{r-1,k-1} - \sum_{j=1}^{k-1}\beta_{j-1}^{r-1,k-2}\nabla^j + \sum_{j=0}^{k-1}\beta_j^{r-1,k-1}\nabla^{j+1}\\
&= \beta_0^{r,k-1} - \beta_0^{r-1,k-1} + 2(1-\cos\theta)\sum_{j=1}^{k-1}\left(\delta_{j,k-2} - \delta_{j,k-1}\right)\beta_{k-1}^{r-1,k-1}\nabla^j + \beta_{k-1}^{r-1,k-1}\nabla^k
\end{aligned}$$

$$= \beta_{k-1}^{r-1,k-1} \left[2(1-\cos\theta)\nabla^{k-2}(1-\nabla) + \nabla^k\right],$$

from which Lemma 4.1 can be deduced.

To prove the second part, one finds with Lemma 1.2 that

$$\begin{aligned}\sum_{j=0}^{k} \beta_j^{r,k}\nabla^j f_{n+1} - \sum_{j=0}^{k-1} \beta_j^{r-1,k-1}\nabla^j f_n &= \beta_k^{r,k}\nabla^k f_{n+1} + \sum_{j=0}^{k-1}\left(\beta_j^{r,k}\nabla^j f_{n+1} - \beta_j^{r-1,k-1}\nabla^j f_n\right)\\ &= \beta_k^{r,k}\nabla^k f_{n+1} + 2(1-\cos\theta)\beta_k^{r,k}\left(\nabla^{k-2} f_{n+1} - \nabla^{k-1} f_{n+1}\right)\\ &\quad + \sum_{j=0}^{k-1}\left(\beta_j^{r,k-1}\nabla^j f_{n+1} - \beta_j^{r-1,k-1}\nabla^j f_n\right),\end{aligned}$$

from which Lemma 4.2 can be derived by means of Lemma 4.1 and Lemma 1.3. ▮

Due to Lemma 4.1, one finds for $k \geq 2$, on subtracting the expression for $y_{n+1}^{[0]}$ from that for $y_{n+1}^{[1]}$ that

$$y_{n+1}^{[1]} = y_{n+1}^{[0]} + h\beta_{k-1}^{0,k-1}\left[\nabla_0^k f_{n+1}^{[\mu-t]} + 2(1-\cos\theta)\nabla^{k-2} f_n^{[\mu-t]}\right]. \tag{4.9}$$

Also, since $\nabla_\nu^i f_{n+1}^{[\mu-t]} - \nabla_{\nu-1}^i f_{n+1}^{[\mu-t]} = f_{n+1}^{[\nu]} - f_{n+1}^{[\nu-1]}$, one finds due to Lemma 1.1

$$y_{n+1}^{[\nu+1]} = y_{n+1}^{[\nu]} + h\beta_{k-1}^{0,k-1}\left(f_{n+1}^{[\nu]} - f_{n+1}^{[\nu-1]}\right), \qquad \nu = 0(1)\mu-1. \tag{4.10}$$

Clearly, the implementation of equations (4.8) is equivalent to that of

$$\begin{array}{ll} \mathrm{P}: & y_{n+1}^{[0]} = y_n^{[\mu]} + h\displaystyle\sum_{i=0}^{k-1}\beta_i^{0,k-1}\nabla^i f_n^{[\mu-t]}, \\ (\mathrm{EC})^\mu: & f_{n+1}^{[0]} = f\left(x_{n+1}, y_{n+1}^{[0]}\right), \\ & y_{n+1}^{[1]} = y_{n+1}^{[0]} + h\beta_{k-1}^{0,k-1}\left[\nabla_0^k f_{n+1}^{[\mu-t]} + 2(1-\cos\theta)\nabla^{k-2} f_n^{[\mu-t]}\right], \\ & f_{n+1}^{[\nu]} = f\left(x_{n+1}, y_{n+1}^{[\nu]}\right), \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \nu = 1(1)\mu-1, \\ & y_{n+1}^{[\nu+1]} = y_{n+1}^{[\nu]} + h\beta_{k-1}^{0,k-1}\left(f_{n+1}^{[\nu]} - f_{n+1}^{[\nu-1]}\right), \\ \mathrm{E}^{1-t}: & f_{n+1}^{[\mu]} = f(x_{n+1}, y_{n+1}^{[\mu]}), \qquad \text{if } t = 0. \end{array} \tag{4.11}$$

It is assumed that the back data $\nabla^i f_n^{[\mu-t]}$, $i = 0(1)k-1$, have been stored. To carry out the sequence (4.11), one needs to compute the differences $\nabla_0^k f_{n+1}^{[\mu-t]}$ and $\nabla_{\mu-1}^k f_{n+1}^{[\mu-t]}$ and a value for θ is necessary in order to be able to compute the β-values. The differences needed can be obtained from

$$\nabla_0^{i+1} f_{n+1}^{[\mu-t]} = \nabla_0^i f_{n+1}^{[\mu-t]} - \nabla^i f_n^{[\mu-t]}, \qquad i = 0(1)k-1,$$

and

$$\nabla_{\mu-1}^k f_{n+1}^{[\mu-t]} = \nabla_0^k f_{n+1}^{[\mu-t]} + f_{n+1}^{[\mu-t]} - f_{n+1}^{[0]},$$

while a value for θ, i.e., κ is due to (4.4) given by

$$\kappa^2 = -\frac{y^{(k+1)}(x_n)}{y^{(k-1)}(x_n)}.$$

Finally, to be ready for the next step, the back data can be updated by computing

$$\nabla^{i+1} f_{n+1}^{[\mu-t]} = \nabla^i f_{n+1}^{[\mu-t]} - \nabla^i f_n^{[\mu-t]}, \qquad i = 0(1)k-2.$$

In an analogous way, the algorithms corresponding to P(ECL)$^\mu$E^{1-t} mode ($t \in \{0,1\}$) can be described. As is the case with the polynomial ABM-schemes when applied in P(ECL)$^\mu$E^{1-t}-mode, it turns out that $\mathrm{P}_{(k)}(\mathrm{EC}_{(k)}\mathrm{L})^\mu\mathrm{E}^{1-t} \equiv \mathrm{P}_{(k)}(\mathrm{EC}_{(k+1)})^\mu\mathrm{E}^{1-t}$, where the subscript denotes the order of the method. Indeed, from (4.7), it turns out that (with $\hat{y}^{[0]}_{n+1} \equiv y^{[0]}_{n+1}$)

$$y^{[\nu+1]}_{n+1} - y^{[\nu]}_{n+1} = (1+W)\left(\hat{y}^{[\nu+1]}_{n+1} - \hat{y}^{[\nu]}_{n+1}\right), \qquad \nu = 0(1)\mu-1.$$

Following the same strategy as the one that led to (4.9) and (4.10), one finds with $W = \beta^{1,k}_k/(\beta^{0,k}_k - \beta^{1,k}_k)$ such that $1+W = \beta^{0,k}_k/\beta^{0,k-1}_{k-1}$, that

$$y^{[1]}_{n+1} = y^{[0]}_{n+1} + h\beta^{0,k}_k\left[\nabla^k_0 f^{[\mu-t]}_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f^{[\mu-t]}_n\right], \tag{4.12}$$

$$y^{[\nu+1]}_{n+1} = y^{[\nu]}_{n+1} + h\beta^{0,k}_k\left(f^{[\nu]}_{n+1} - f^{[\nu-1]}_{n+1}\right), \qquad \nu = 0(1)\mu-1. \tag{4.13}$$

On the other hand, if a corrector of order $k+1$ is used, then on account of Lemma 4.2, one finds from (4.11) for the $\nu = 0$ case

$$\begin{aligned} y^{[1]}_{n+1} &= y^{[0]}_{n+1} + h\sum_{i=0}^{k}\beta^{1,k}_i\nabla^i f^{[\mu-t]}_{n+1} - h\sum_{i=0}^{k-1}\beta^{0,k-1}_i\nabla^i f^{[\mu-t]}_n \\ &= y^{[0]}_{n+1} + h\beta^{0,k}_0\left[\nabla^k_0 f^{[\mu-t]}_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f^{[\mu-t]}_n\right], \end{aligned}$$

while again (4.13) is found for $\nu = 0(1)\mu-1$, i.e., both algorithms use the same formulas.

If Milne's device is also used to perform step-size control, then one also needs an expression for this estimation (denoted by T) of the PLTE. Due to Lemma 4.1 and Lemma 4.3, one finds

$$T_{n+1} = h\beta^{1,k}_k\left[\nabla^k_{\mu-1} f^{[\mu-t]}_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f^{[\mu-t]}_n\right].$$

The P(ECL)$^\mu$E^{1-t} mode, $t \in \{0,1\}$, can thus be implemented as follows:

$$\begin{array}{lll}
\text{P:} & y^{[0]}_{n+1} = y^{[\mu]}_n + h\displaystyle\sum_{i=0}^{k-1}\beta^{0,k-1}_i\nabla^i f^{[\mu-t]}_n, & \\
(\text{ECL})^\mu\text{:} & f^{[0]}_{n+1} = f\left(x_{n+1}, y^{[0]}_{n+1}\right), & \\
 & y^{[1]}_{n+1} = y^{[0]}_{n+1} + h\beta^{0,k}_k\left[\nabla^k_0 f^{[\mu-t]}_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f^{[\mu-t]}_n\right], & \\
 & f^{[\nu]}_{n+1} = f\left(x_{n+1}, y^{[\nu]}_{n+1}\right), & (4.14) \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \nu = 1(1)\mu-1, & \\
 & y^{[\nu+1]}_{n+1} = y^{[\nu]}_{n+1} + h\beta^{0,k}_k\left(f^{[\nu]}_{n+1} - f^{[\nu-1]}_{n+1}\right), & \\
\text{E}^{1-t}\text{:} & f^{[\mu]}_{n+1} = f\left(x_{n+1}, y^{[\mu]}_{n+1}\right), \quad \text{if } t = 0, & \\
\text{T:} & T_{n+1} = h\beta^{1,k}_k\left[\nabla^k_{\mu-1} f^{[\mu-t]}_{n+1} + 2(1-\cos\theta)\nabla^{k-2} f^{[\mu-t]}_n\right]. &
\end{array}$$

Now that we have shown that the computational effort and storage of the differences can be reduced significantly, one may wonder what computational overhead is associated with the computation of κ^2. First of all, an expression for ${}^i\kappa^2$, $i = 1(1)m$ has to be derived by differentiation of the differential system of interest. These expressions are then used to compute one set of β-values in each knot point (in fact, only $\beta^{0,k}_{k-1}$ and $\beta^{0,k}_k$ have to be recomputed in each point). This means that, in general, $k-2+2m$ values are needed, i.e., there is no storage overhead when applied to scalar equations with fixed k during the integration process. If k varies, special care has to be taken since for each k-considered different $\beta_{k-1}^{r,k}$ and $\beta_k^{r,k}$ values have to be used.

To support the theory, we consider the application of our methods on two popular problems. For bigger problems, analogous results can be obtained. As a first example, we will investigate the application of a $\mathrm{P}_{(k)}(\mathrm{EC}_{(k)}\mathrm{L})^{\mu}$-mode to the problem

$$y' = \sqrt{1-y^2}\sqrt{1-my^2}, \qquad y(0) = 0, \tag{4.15}$$

for which the exact solution is given by the elliptic sine function. In this example, we use $m = 0.25$. This problem has already been discussed in [15] in the framework of explicit methods.

Since local extrapolation is getting used, we take $\mu \geq 2$ in order to be able to attribute a value to κ^2. If the current value that is being computed is y_{n+1}, we attribute to κ^2 the value κ_n^2 given by

$$\kappa_n^2 = -\frac{y^{(k+2)}(x_n)}{y^{(k)}(x_n)}, \tag{4.16}$$

where the higher order derivatives are re-expressed in terms of x_n and $y(x_n)$ by means of the differential equation in (4.15). One thus obtains

$$\kappa_n^2 = \begin{cases} \dfrac{1+14m+m^2-20m(1+m)y_n^2+24m^2y_n^4}{1+m-2my_n^2}, & k=2,\\[2mm] \dfrac{1+14m+m^2-60m(1+m)y_n^2+120m^2y_n^4}{1+m-6my_n^2}, & k=3,\\[2mm] \dfrac{1}{1+14m+m^2-60m(1+m)y_n^2+120m^2y_n^4} & \\ \times(1+135m+135m^2+m^3 & \\ -m(182+868m+182m^2)y_n^2 & \\ +840m^2(1+m)y_n^4-720m^3y_n^6), & k=4. \end{cases} \tag{4.17}$$

With these values for κ_n^2, the β-coefficients are computed once and used throughout the iteration.

One might argue that there are more efficient ways to calculate κ_n^2. Indeed, one can approximate the derivatives in (4.16) by means of backward differences of f since the algorithm already needs these differences. However, there is a technical problem at the start of the integration process since differences up to order $k+1$ are needed and only differences up to order $k-1$ can be calculated from the starting values. Since we did not try to obtain a self-starting code, i.e., we did not consider the problem of how to obtain the necessary starting values anyhow, we didn't follow this idea to calculate κ_n^2 neither.

The results are shown in Table 3 and Table 4. In Table 3, the differences $y(x_{n+1}) - y_{n+1}$ (denoted as Δ_m if mixed interpolation is used and Δ_c for the classical methods) are shown for $k = 2$, 3 and 4 and $\mu = 2$ (upper value) and 3 (lower value) in different knot points for $h = 0.1$. In Table 4, the corresponding results are shown for $h = 0.01$. For both values of h, exact starting values are used. The starting points are chosen such that the first calculated y-value is located in $x = 0.6$ in both cases.

For the modified methods, the values of κ_n^2 are tabulated in the different knot points. Since they are based on the approximation of y, different values of κ_n^2 are expected for different μ-values. However, it turns out that in most cases, the κ_n^2 values are the same (to the accuracy shown). Only if this is not the case, the two different values are tabulated.

Most of the theoretical considerations can be observed from Table 3 and Table 4. Indeed, since the first calculated value for both tables is the same and since exact starting values are used, order comparisons can be made at this point. One can, e.g., easily deduce from the results for both h values that the modified methods possess order $k+2$. The polynomial methods, on the

Table 3. Absolute errors produced by the modified (Δ_m) and classical (Δ_c) k-step ABM-methods with stepsize $h = 0.1$ applied in P(ECL)$^\mu$ mode for $k = 2$, 3 and 4 and $\mu = 2$ and 3 to problem (4.15). Exact starting values are assumed; $y(0.6)$ is the first calculated value. The notation (n) means 10^{-n}.

x	$k = 2$			$k = 3$			$k = 4$		
	Δ_m	κ_n^2	Δ_c	Δ_m	κ_n^2	Δ_c	Δ_m	κ_n^2	Δ_c
0.6	8.041 (7)	2.839	−6.257 (6)	−4.198 (8)	0.782	−1.948 (7)	2.426 (9)	7.205	2.086 (7)
	8.173 (8)		−6.186 (6)	−7.655 (8)		−2.417 (7)	1.504 (8)		2.042 (7)
0.8	4.757 (6)	2.184	−1.837 (5)	2.855 (7)	−2.733	4.844 (7)	−4.206 (8)	5.158	5.180 (7)
	1.670 (7)		−1.629 (5)	−2.498 (7)		8.374 (8)	5.458 (8)		5.032 (7)
1.0	8.276 (6)	1.469	−2.436 (5)	2.457 (6)	−10.544	1.847 (6)	−9.369 (8)	2.325	5.384 (7)
	1.655 (7)		−2.101 (5)	−5.162 (7)		9.602 (7)	9.466 (8)		5.753 (7)
1.2	1.025 (5)	0.788	−2.303 (5)	1.748 (5)	−47.330	3.072 (6)	−1.657 (7)	−2.563	2.538 (7)
	3.915 (8)		−1.949 (5)	−1.811 (6)	−47.339	1.688 (6)	1.303 (7)		4.405 (7)
1.4	9.683 (6)	0.229	−1.489 (5)	7.437 (4)	71.817	3.390 (6)	−3.110 (7)	−22.149	−2.807 (7)
	−2.244 (7)	0.228	−1.290 (5)	−4.552 (5)	71.250	1.718 (6)	1.992 (7)	−22.148	2.194 (7)

Table 4. Absolute errors produced by the modified (Δ_m) and classical (Δ_c) k-step ABM-methods with stepsize $h = 0.01$ applied in P(ECL)$^\mu$ mode for $k = 2$, 3 and 4 and $\mu = 2$ and 3 to problem (4.15). Exact starting values are assumed; $y(0.6)$ is the first calculated value. The notation (n) means 10^{-n}.

x	$k = 2$			$k = 3$			$k = 4$		
	Δ_m	κ_n^2	Δ_c	Δ_m	κ_n^2	Δ_c	Δ_m	κ_n^2	Δ_c
0.6	7.109 (12)	2.558	−6.443 (10)	−2.411 (14)	−0.526	8.576 (13)	1.039 (15)	6.362	1.861 (13)
	1.090 (12)		−6.415 (10)	−8.188 (14)		8.096 (13)	2.136 (15)		1.862 (13)
0.8	3.297 (10)	1.864	−1.159 (8)	5.843 (12)	−5.354	6.241 (11)	−2.615 (14)	4.010	2.743 (12)
	2.329 (11)		−1.146 (8)	−1.708 (12)		5.923 (11)	4.517 (14)		2.766 (12)
1.0	5.968 (10)	1.153	−1.700 (8)	4.006 (11)	−18.637	1.571 (10)	−6.937 (14)	0.564	2.969 (12)
	4.193 (11)		−1.680 (8)	−3.443 (12)		1.504 (10)	8.149 (14)		3.063 (12)
1.2	7.101 (10)	0.516	−1.623 (8)	5.213 (10)	−392.547	2.190 (10)	−1.375 (13)	−7.073	1.530 (12)
	4.985 (11)		−1.604 (8)	−1.143 (11)		2.098 (10)	1.115 (13)		1.745 (12)
1.4	6.057 (10)	0.037	−1.054 (8)	−5.693 (10)	41.906	2.022 (10)	−3.673 (13)	−157.885	−4.204 (13)
	4.197 (11)		−1.048 (8)	4.866 (12)		1.942 (10)	2.250 (13)		−4.087 (14)

other hand, always have order $k + 1$. Also, one can verify that the $\mu = 2$ and $\mu = 3$ cases possess the same order, which again confirms the theory.

Secondly, some interesting properties can be deduced from this example. First of all, initially all modified methods produce better results than the corresponding classical ones as was expected by the theory. However, since the solution of problem (4.15) is far from a linear combination of a sine, a cosine, and a polynomial function, κ_n^2 is far from a constant value. It even turns out that for $k = 3$ and $k = 4$, κ_n^2 is attributed very large (positive and negative) values. It can be seen from Tables 3 and 4 that in these points, the accuracy attained by the modified methods decreases significantly due to these (too) large values.

A reasonable question is of course: what is the computational cost of our methods in comparison with their classical counterparts? To answer this question, we have incorporated Table 5. Here t_c and t_m denote the times in (sec/100000) that are needed to proceed with $k = 2$ in P(ECL)$^\mu$-mode from x_0 to $x_N = 1.4$ where x_0 is chosen in such a way that the first calculated y-value is located in $x = 0.6$ (actually, times were measured in sec/100 on a 486-66 MHz PC and the program was executed 1000 times). For $k = 2$, we obtained from Table 3 and Table 4 that κ^2 is small, which means that the modified coefficients can easily be written in Taylor-series expansions where terms up to κ^4 are kept. As expected, the execution of the modified method is slower, but the gain in accuracy clearly compensates this disadvantage for high accuracies.

Table 5. Absolute errors (Δ) and execution times (t) (in sec/100.000) in $x = 1.4$ produced by the modified (Δ_m) and classical (Δ_c) k-step ABM-methods for various stepsizes applied in P(ECL)$^\mu$ mode for $k = 2$ and 3 and $\mu = 2$ to problem (4.15). The notation (n) means 10^{-n}.

$\mu = 2$				
h	Δ_c	t_c	Δ_m	t_m
0.100	−1.489 (5)	208	1.177 (5)	242
0.050	−1.519 (6)	269	5.679 (7)	335
0.025	−1.733 (7)	390	3.098 (8)	516
0.020	−8.722 (8)	450	1.234 (8)	612
0.010	−1.054 (8)	747	7.292 (10)	1054
0.005	−1.295 (9)	1351	4.430 (11)	1955
$\mu = 3$				
h	Δ_c	t_c	Δ_m	t_m
0.100	−1.290 (5)	231	1.878 (6)	269
0.050	−1.450 (6)	308	1.121 (7)	373
0.025	−1.703 (7)	472	6.688 (9)	593
0.020	−8.608 (8)	549	2.710 (9)	703
0.010	−1.048 (8)	945	1.655 (10)	1252
0.005	−1.292 (9)	1736	1.021 (11)	2345

As a second example, we consider the popular test problem

$$z'' + z = 0.001e^{\mathrm{i}x}, \qquad z(0) = 1, \quad z'(0) = 0.9995\,\mathrm{i}, \tag{4.18}$$

for which the exact solution is given by $z(x) = (1 - 0.0005\mathrm{i}x)e^{\mathrm{i}x}$.

This system, known as the Stiefel-Bettis problem, has been used earlier several times to study methods designed for problems with nearly periodic solutions (see, e.g., [3,13,16]).

To solve this problem, we write (4.18) as

$$\begin{aligned} {}^1y' &= {}^2y, & {}^1y(0) &= 1, \\ {}^2y' &= -{}^1y + 0.001\cos x, & {}^2y(0) &= 0, \\ {}^3y' &={}^4 y, & {}^3y(0) &= 0, \\ {}^4y' &= -\ {}^3y + 0.001\sin x, & {}^4y(0) &= 0.9995, \end{aligned} \tag{4.19}$$

for which the exact solution is given by

$$\begin{aligned} {}^1y(x) &= \cos x + 0.0005x\sin x, \\ {}^2y(x) &= -0.9995\sin x + 0.0005x\cos x, \\ {}^3y(x) &= \sin x - 0.0005x\cos x, \\ {}^4y(x) &= 0.9995\cos x + 0.0005x\sin x. \end{aligned} \tag{4.20}$$

The values attributed to ${}^i\kappa^2$ according to (4.6) for $k = 2$, 3 and 4 are

$$k = 2: \begin{cases} {}^1\kappa^2 = \dfrac{{}^1y_n - 0.002\cos x_n}{{}^1y_n - 0.001\cos x_n}, \\ {}^2\kappa^2 = \dfrac{{}^2y_n + 0.002\sin x_n}{{}^2y_n + 0.001\sin x_n}, \\ {}^3\kappa^2 = \dfrac{{}^3y_n - 0.002\sin x_n}{{}^3y_n - 0.001\sin x_n}, \\ {}^4\kappa^2 = \dfrac{{}^4y_n - 0.002\cos x_n}{{}^1y_n - 0.001\cos x_n}, \end{cases} \tag{4.21}$$

$$k = 3 : \begin{cases} {}^1\kappa^2 = \dfrac{{}^2y_n + 0.002 \sin x_n}{{}^2y_n + 0.001 \sin x_n}, \\ {}^2\kappa^2 = \dfrac{{}^1y_n - 0.003 \cos x_n}{{}^1y_n - 0.002 \cos x_n}, \\ {}^3\kappa^2 = \dfrac{{}^4y_n - 0.002 \cos x_n}{{}^4y_n - 0.001 \cos x_n}, \\ {}^4\kappa^2 = \dfrac{{}^3y_n - 0.003 \sin x_n}{{}^3y_n - 0.002 \sin x_n}, \end{cases} \tag{4.22}$$

$$k = 4 : \begin{cases} {}^1\kappa^2 = \dfrac{{}^1y_n - 0.003 \cos x_n}{{}^1y_n - 0.002 \cos x_n}, \\ {}^2\kappa^2 = \dfrac{{}^2y_n + 0.003 \sin x_n}{{}^2y_n + 0.002 \sin x_n}, \\ {}^3\kappa^2 = \dfrac{{}^3y_n - 0.003 \sin x_n}{{}^3y_n - 0.002 \sin x_n}, \\ {}^4\kappa^2 = \dfrac{{}^4y_n - 0.003 \cos x_n}{{}^4y_n - 0.002 \cos x_n}. \end{cases} \tag{4.23}$$

We applied the $\mathrm{P}_{(k)}(\mathrm{EC}_{(k)}\mathrm{L})^{\mu}$ algorithm to problem (4.19) for $k = 2$ and $k = 3$ with $\mu = 2$. The results are shown in Table 6, where the errors in $|z(40\pi)|$ are listed. The results are produced with exact starting values. Each time x_0 was set to π.

Table 6. Absolute errors in $|z(40\pi)|$ produced by the modified (Δ_m) and classical (Δ_c) k-step ABM-methods for various stepsizes applied in $\mathrm{P(ECL)}^{\mu}$ mode for $k = 2$ and 3 and $\mu = 2$ to problem (4.18). The column Δ_f corresponds to our modified method with fixed κ values. The notation (n) means 10^{-n}.

h	$k = 2$			$k = 3$		
	Δ_m	Δ_f	Δ_c	Δ_m	Δ_f	Δ_c
$\pi/4$	1.220 (3)	1.362 (4)	9.716 (1)	−5.329 (4)	1.077 (4)	−1.953 (0)
$\pi/8$	7.894 (5)	1.500 (5)	−6.036 (2)	−3.804 (6)	9.130 (7)	−1.500 (1)
$\pi/16$	4.513 (6)	1.047 (6)	−3.131 (2)	−2.610 (7)	9.939 (8)	−5.315 (3)

For both k-values, 3 results are given. As in previous tables, Δ_c stands for the classical case, Δ_m for the modified one. It is obvious from the above results that the modified methods give a considerable gain due to the choice (4.6) with $p = k$. As it turns out that this results in ${}^i\kappa^2 \approx 0.999$ for all values of x, we have also performed our modified scheme where each ${}^i\kappa^2$ is given this fixed value 0.999. The results obtained (see columns labelled Δ_f) even produce better results than with variable ${}^i\kappa^2$.

From the above, we may conclude that our modified methods can be implemented in a predictor-corrector pair without any problem. So far however, these methods have only been studied in fixed-step implementations. The study of variable step and/or variable order implementations has not been performed yet. Developing a code in which all of this is included is beyond the scope of this paper. These matters remain challenges for future work.

REFERENCES

1. W. Gautschi, Numerical integration of ordinary differential equations based on trigonometric polynomial, *Numer. Math* **3**, 381–397 (1961).
2. E. Stiefel and D.G. Bettis, Stabilization of Cowell's method, *Numer. Math.* **13**, 154–175 (1969).
3. D.G. Bettis, Numerical integration of products of Fourier and ordinary polynomials, *Numer. Math.* **14**, 421–434 (1970).
4. T. Lyche, Chebyshevian multistep methods for ordinary differential equations, *Numer. Math.* **19**, 65–75 (1972).

5. A. Raptis and A.C. Allison, Exponential-fitting methods for the numerical solution of the Schrödinger equation, *Comput. Phys. Comm.* **4**, 1–4 (1978).
6. L.G. Ixaru and M. Rizea, A Numerov-like scheme for the numerical solution of the Schrödinger equation in the deep continuum spectrum of energies, *Comput. Phys. Comm.* **19**, 23–27 (1980).
7. P.J. van der Houwen and B.P. Sommeijer, Linear multistep methods with reduced truncation errors for periodic initial value problems, *IMA J. Numer. Anal.* **4**, 479–489 (1984).
8. B. Neta and C.H. Ford, Families of methods for ordinary differential equations based on trigonometric polynomials, *J. Comput. Appl. Math.* **10**, 33–38 (1984).
9. B. Neta, Families of backward differentiation methods based on trigonometric polynomials, *J. Comput. Math.* **20**, 67–75 (1986).
10. A.D. Raptis and J.R. Cash, Exponential and Bessel fitting methods for the numerical solution of the Schrödinger equation, *Comp. Phys. Comm.* **44**, 95–103 (1987).
11. J. Panovsky and D.L. Richardson, A family of implicit Chebyshev methods for the numerical integration of second-order differential equations, *J. Comput. Appl. Math.* **23**, 35–51 (1988).
12. J.P. Coleman, Numerical methods for $y'' = f(x,y)$ via rational approximation of the cosine, *IMA J. Numer. Anal.* **9**, 145–165 (1992).
13. J.P. Coleman and A.S. Booth, Analysis of a family of Chebyshev methods for $y'' = f(x,y)$, *J. Comput. Appl. Math.* **44**, 95–114 (1992).
14. T.E. Simos, Some new four-step exponential-fitting methods for the numerical solution of the radial Schrödinger equation, *IMA J. Numer. Anal.* **11**, 347–356 (1991).
15. J. Vanthournout, G. Vanden Berghe and H. De Meyer, Families of backward differentiation methods based on a new type of mixed interpolation, *Computers Math. Applic.* **20** (11), 19–30 (1990).
16. J. Vanthournout, H. De Meyer and G. Vanden Berghe, Multistep methods for ordinary differential equations based on algebraic and first order trigonometric polynomials, In *Computational Ordinary Differential Equations*, (Edited by J.R. Cash and I. Gladwell), pp. 61–72, Clarendon Press, Oxford, (1992).
17. H. De Meyer, J. Vanthournout and G. Vanden Berghe, On a new type of mixed interpolation, *J. Comput. Appl. Math.* **30**, 55–69 (1990).
18. H. De Meyer, J. Vanthournout, G. Vanden Berghe and A. Vanderbauwhede, On the error estimation for a mixed type of interpolation, *J. Comput. Appl. Math.* **32**, 407–415 (1990).
19. E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin, (1987).
20. P. Henrici, *Discrete Variable Methods for Ordinary Differential Equations*, J. Wiley and Sons, New York, (1962).
21. J.D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley, Chichester, (1991).
22. G. Nürnberger, *Approximation by Spline Functions*, Springer-Verlag, Berlin, (1989).