J Math Chem (2009) 46:866–895
DOI 10.1007/s10910-009-9561-1

ORIGINAL PAPER

A family of numerical multistep methods with three distinct schemes: explicit advanced step-point (EAS) methods and the EAS1 approach

G. Psihoyios

Received: 29 March 2007 / Accepted: 30 September 2008 / Published online: 23 July 2009

Abstract In this work, for the first time in an article, we present in a comprehensive way the explicit advanced step-point (EAS) methods. The EAS methods is a family of methods designed for the numerical solution of non-stiff and mildly stiff initial value problems (IVPs) and comprises three distinct schemes: EAS1, EAS2 and EAS3. A thorough theoretical analysis of the EAS family of predictor–corrector methods is presented in terms of their accuracy and stability characteristics and requirements, as well as the rationale for creating the three distinct schemes mentioned above. In this paper we also examine in detail one of the three schemes, the EAS1 methods. EAS1 are assessed for the very first time, are meticulously studied and their superior regions of absolute stability are presented. Furthermore the computational efficiency of EAS1 is examined and comparative numerical results are presented with the use of a variable step, variable order EAS1 code. The numerical results provide good evidence that EAS1 could be seen as superior to the well established Adams methods for the numerical solution of mildly stiff initial value problems.

Keywords Explicit advanced step-point (EAS) methods · Mildly stiff and non-stiff initial value problems · EAS1 methods · Predictor–corrector methods · Multistep methods · Absolute stability · Local error estimates · Adams methods

G. Psihoyios
Clore Laboratory, University of Buckingham, Buckingham MK18 1EG, UK

G. Psihoyios (✉)
P.O. Box 61030, Amaroussion, Athens 15101, Greece
e-mail: g.psihoyios@ntlworld.com

1 Introduction

The term “Explicit Advanced Step-point” (EAS) methods appeared in [1], where most of the first stage of the work on EAS methods was carried out. Before the completion of [1], two introductory works had appeared [2,3], which were not only preliminary, but most importantly short and in their early stages. There is no noteworthy overlapping between the present work and these two earlier papers. In 2004 paper [4] was published, which deals with some general formulae for the stability functions of EAS methods. Prior to the appearance of [5], surprisingly enough, despite the fact that much work has been done on the EAS methods, effectively only their general form was known. In other words, before [5] and the present work, the methods themselves, essentially, had never had their capabilities assessed in journal articles. This is the second in a series of papers that aims to remedy this fundamental gap in the existing literature on the EAS schemes. It is worth mentioning that although there were reasons beyond our control that had postponed the timely publication of the present research, which should had appeared before [5], naturally, we assume full responsibility for any hold-up.[1]

The second stage of the work on EAS methods began with their trigonometric and exponential fitting, which took the EAS methods to a different level and empowered them to better tackle certain types of problems. The trigonometric and exponential fitting of EAS methods for $k = 1$ can be found in [6] and further research publications should be expected in the future. Following the same methodology as in [6], we applied, for the first time in the literature, trigonometric fitting to the Adams predictor–corrector methods (also known as Adams–Bashforth–Moulton methods) [7–13], which forms part of the second stage of the work mentioned above. As we will see in the next sections, the original Adams methods (not the trigonometrically fitted versions) are crucial to the present research. This work deals solely with the explicit advanced step-point (EAS) methods; readers who may be interested in the implicit advanced step-point (IAS) methods are referred to [14–19].

The purpose of the present work is to comprehensively study and assess, for the first time, the EAS methods and in particular the EAS1 scheme. We will examine in detail the general approach to the EAS methods, their general accuracy characteristics, the theoretical basis on which we build their accuracy and stability requirements, their stability graphs (EAS1), their local error estimates and their implementation. It should be stated that the EAS1 scheme could be successfully used for the solution of certain chemical reaction problems.

This rest of this work is organised as follows: In sect. 2 we give the EAS general form; in sect. 3 we discuss in detail the accuracy of the EAS schemes; in sect. 4 we establish our theoretical accuracy and stability requirements; in sect. 5 we discuss the rationale behind the development of three distinct EAS schemes; in sect. 6 we examine the EAS1 scheme and we present twelve absolute stability region graphs; in sect. 7 we consider the implementation of the EAS1 into a variable order/variable step code;

[1] Please note that in [5], due to a typo, this present work is wrongly cited as having been accepted in the journal “Mathematical and Computer Modeling” instead of the correct “Journal of Mathematical Chemistry”.

in sect. 8 we present our numerical results and comparisons and in sect. 9 we draw some conclusions for this work.

2 Adams predictor–corrector methods and the EAS general form

In 1975 Shampine and Gordon [20] presented one of the most successful implementations of the Adams class of methods, which was based on the very well known Adams–Bashforth and Adams–Moulton formulae, used as a Predictor–Corrector (P–C) pair. Shampine and Gordon used the following formula in the implementation of their P–C scheme:

$$\left.\begin{aligned} \bar{y}_{n+1} &= y_n + h \sum_{i=1}^{k} \gamma_{i-1} \nabla^{i-1} f_n \\ y_{n+1} &= \bar{y}_{n+1} + h\gamma_k \nabla^k \bar{f}_{n+1} \end{aligned}\right\} \tag{1}$$

where $\bar{f}_{n+1} = f(x_{n+1}, \bar{y}_{n+1})$.

The main idea for developing the numerical EAS schemes for non-stiff IVPs is as follows:

1. Use an explicit k-step predictor to compute $y^{(p)}_{n+k}$, where the superscript (p) stands for "predictor" .
2. Evaluate $f^{(p)}_{n+k} = f(x_{n+k}, y^{(p)}_{n+k})$ and use $y^{(p)}_{n+k}$ together with $f^{(p)}_{n+k}$, in the next stage, for the formulation of the explicit second (k+1)-step predictor.
3. Use an explicit $(k+1)$-step predictor to compute $y^{(p)}_{n+k+1}$.
4. Evaluate $f^{(p)}_{n+k+1} \equiv f(x_{n+k+1}, y^{(p)}_{n+k+1}$ and using

$$\begin{aligned} &\left(y_{n+j}, y'_{n+j}\right), \qquad 0 \le j \le k-1 \\ &\left(y^{(p)}_{n+i}, f^{(p)}_{n+i}\right), \qquad i = k, k+1 \end{aligned}$$

compute the corresponding explicit corrected solution $y^{(c)}_{n+k}$, where the superscript (c) stands for "corrector" .

According to the above steps our approach takes the following very general form:

$$\left.\begin{aligned} &y^{(p)}_{n+k} + \sum_{j=0}^{k-1} a_j y_{n+j} = h \sum_{j=0}^{k-1} b_j f_{n+j} \\ &y^{(p)}_{n+k+1} + \bar{a}_k y^{(p)}_{n+k} + \sum_{j=0}^{k-1} \bar{a}_j y_{n+j} = h \sum_{j=0}^{k-1} \bar{b}_j f_{n+j} + h\bar{b}_k f^{(p)}_{n+k} \\ &y^{(c)}_{n+k} + \sum_{j=0}^{k-1} \hat{a}_j y_{n+j} = h \sum_{j=0}^{k-1} \hat{b}_j f_{n+j} + h\left[\hat{b}_k f^{(p)}_{n+k} + \hat{b}_{k+1} f^{(p)}_{n+k+1}\right] \end{aligned}\right\} \tag{2}$$

where the superscripts (p) and (c) stand for "predictor" and "corrector", respectively. In order to compute the corrected solution $y^{(c)}_{n+k}$, besides using information at the current point x_{n+k}, we also use information at the advanced step-point x_{n+k+1}, as well as at previous points.

The form (2) is the most general form that our methods may take. The obvious problem now is the proper choice of coefficients. This will be discussed in detail in sects. 4, 5 and 6. For the time being it suffices to say that one of the most effective approaches is to use Adams coefficients and to choose for (2):

$$a_{k-1} = \bar{a}_k = \hat{a}_{k-1} \equiv -1, a_j = \bar{a}_0 = \bar{a}_{j+1} = \hat{a}_j = 0, \quad 0 \le j \le k-2. \tag{3}$$

An additional advantage we have by adopting this approach (in terms of changing order and storing information) is that we are able to express our formulae in backward difference form. Very briefly, we need to identify an interpolation polynomial $p_{k+2}(x)$ of degree $(k+1)$ which passes through the points $(x_{n+2}, f_{n+2}), (x_{n+1}, f_{n+1}), \ldots,$ (x_{n-k+1}, f_{n-k+1}) and this is done in an equivalent way as with the Adams–Moulton formula (for more details the reader is referred to [4]). Finally using (3) and our newly identified interpolation polynomial [4], we can express the EAS methods in backward difference form as follows:

$$\left.\begin{aligned}
y_{n+k}^{(p)} &= y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i \nabla^i f_{n+k-1}\\
y_{n+k+1}^{(p)} &= y_{n+k}^{(p)} + h\sum_{i=0}^{k}\gamma_i \nabla^i f_{n+k}^{(p)} - \gamma_k h a \nabla^k f_{n+k}^{(p)}\\
y_{n+k}^{(c)} &= y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i \nabla^i f_{n+k-1} + \gamma_k h \nabla^k f_{n+k}^{(p)} + h\bar{a}\nabla^{k+1} f_{n+k+1}^{(p)}
\end{aligned}\right\} \tag{4}$$

where a and $\bar{a}$ are free coefficients.

We may describe the steps followed by (4) as follows:

Step 1: Compute a predicted solution $y_{n+k}^{(p)}$ of order k using a standard k-step Adams predictor

$$y_{n+k}^{(p)} = y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i \nabla^i f_{n+k-1}.$$

Step 2: Evaluate $f_{n+k}^{(p)} = f(x_{n+k}, y_{n+k}^{(p)})$.

Step 3: Compute a second predicted solution $y_{n+k+1}^{(p)}$ of order k at x_{n+k+1} using an explicit linear multistep formula of the form

$$y_{n+k+1}^{(p)} = y_{n+k}^{(p)} + h\sum_{i=0}^{k}\gamma_i \nabla^i f_{n+k}^{(p)} - \gamma_k h a \nabla^k f_{n+k}^{(p)}.$$

Step 4: Evaluate $f_{n+k+1}^{(p)} = f\left(x_{n+k+1}, y_{n+k+1}^{(p)}\right)$.

Step 5: Compute a corrected solution of order $(k+1)$ at x_{n+k} as

$$y_{n+k}^{(c)} = y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i\nabla^i f_{n+k-1} + \gamma_k h\nabla^k f_{n+k}^{(p)} + h\bar{a}\nabla^{k+1} f_{n+k+1}^{(p)}.$$

Step 6: Evaluate $f_{n+k}^{(c)} = f\left(x_{n+k}, y_{n+k}^{(c)}\right)$ and assuming that $y_{n+k}^{(c)}$ satisfies the local error requirement (discussed in sects. 3.1 and 7) we set $y_{n+k} = y_{n+k}^{(c)}$, $f_{n+k} = f\left(x_{n+k}, y_{n+k}^{(c)}\right)$.

3 Accuracy of EAS methods

The general form (4) contains three difference equations and each one of them has a different local truncation error (LTE). Our main interest is in the corrector but the LTEs of the two predictors are indispensable for the calculation of the LTE of the corrector. We can show using Taylor series that:

- The first predictor $y_{n+k}^{(p)}$ from (4) has the following LTE

$$\text{LTE}_{P1} \equiv y(x_{n+k}) - y_{n+k}^{(p)} = \gamma_k h^{k+1} y^{(k+1)}(x_n) + 0(h^{k+2}) \tag{5}$$

where $P1$ stands for predictor-1 and $y_{n+k}^{(p)}$ is of order k.

- The second predictor $y_{n+k+1}^{(p)}$ from (4) has a LTE of the form

$$\text{LTE}_{P2} \equiv y(x_{n+k+1}) - y_{n+k+1}^{(p)} = \gamma_k(1+a)h^{k+1} y^{(k+1)}(x_n) + 0(h^{k+2}) \tag{6}$$

where $P2$ stands for predictor-2 and $y_{n+k+1}^{(p)}$ is also of order k.

When a predictor–corrector scheme is applied in a PECE (prediction-evaluation-correction-evaluation) mode, as it is basically the case with EAS also, the LTE of the corrector may be “polluted” by that of the predictors. The level of this “pollution” can be shown by the LTE of the corrector. Now instead of attempting to work with Taylor expansions, we will work directly with the general case, using the definition of the LTE, and thus establishing the LTE of the corrector $\forall\ \ k = 1, 2, \ldots$. This results in a new proof for the LTE of the EAS methods.

Theorem 1 *The EAS schemes corrector $y_{n+k}^{(c)}$ from (4) has a $(k+1)$ order LTE given by*

$$\begin{aligned}\text{LTE} \equiv{}& [\bar{a}(1+a) + \gamma_k - (k+1)\bar{a}]\gamma_k h^{k+2}\frac{\vartheta f}{\vartheta y} y^{(k+1)}(x_n)\\ &- \left(\bar{a} - \gamma_{k+1}^*\right) h^{k+2} y^{(k+2)}(x_n) + 0(h^{k+3})\end{aligned} \tag{7}$$

where $\gamma_{k+1}^* = \gamma_{k+1} - \gamma_k$, is the respective Adams–Moulton coefficient.

Proof We know that the LTE is given by

$$\mathrm{LTE} \equiv y(x_{n+k}) - y_{n+k}^{(c)}.$$

Our corrector is of order $(k+1)$ and as we see from (4), we need $f_{n+k}^{(p)}$ and $f_{n+k+1}^{(p)}$. These quantities are given by:

$$f_{n+k}^{(p)} - f(x_{n+k}, y(x_{n+k})) \approx -\frac{\vartheta f}{\vartheta y}\mathrm{LTE}_{P1}(h^{k+1}), \tag{8}$$

where $\mathrm{LTE}_{P1}(h^{k+1})$ represents the LTE of $y_{n+k}^{(p)}$ from (5), and

$$f_{n+k+1}^{(p)} - f(x_{n+k+1}, y(x_{n+k+1})) \approx -\frac{\vartheta f}{\vartheta y}\mathrm{LTE}_{P2}(h^{k+1}), \tag{9}$$

where $\mathrm{LTE}_{P2}(h^{k+1})$ represents the LTE of $y_{n+k+1}^{(p)}$ from (6).

Using (4) for $y_{n+k}^{(c)}$, we finally get:

$$\begin{aligned} y_{n+k}^{(c)} &= y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i\nabla^i f_{n+k-1} + \gamma_k h\nabla^k f_{n+k}^{(p)} + h\bar{a}\nabla^{k+1} f_{n+k+1}^{(p)} = \cdots \\ &= y_{n+k-1} + h\sum_{i=0}^{k-1}\gamma_i\nabla^i f_{n+k-1} + \gamma_k h\nabla^k f_{n+k} + h\bar{a}\nabla^{k+1} f_{n+k+1} \\ &\quad + \gamma_k h\left(\nabla^k f_{n+k}^{(p)} - \nabla^k f_{n+k}\right) + h\bar{a}\left(\nabla^{k+1} f_{n+k+1}^{(p)} - \nabla^{k+1} f_{n+k+1}\right). \end{aligned} \tag{10}$$

Accordingly $y(x_{n+k})$ becomes:

$$\begin{aligned} y(x_{n+k}) = y(x_{n+k-1}) &+ h\sum_{i=0}^{k-1}\gamma_i\nabla^i f(x_{n+k-1}) \\ &+ \gamma_k h\nabla^k f(x_{n+k}) + h\bar{a}\nabla^{k+1} f(x_{n+k+1}) + 0(h^{k+3}). \end{aligned} \tag{11}$$

Making the usual localizing assumption and subtracting (10) from (11) we get:

$$\begin{aligned} \mathrm{LTE} &\equiv y(x_{n+k}) - y_{n+k}^{(c)} \\ &= \gamma_k h\left(\nabla^k f(x_{n+k}) - \nabla^k f_{n+k}^{(p)}\right) + h\bar{a}\left(\nabla^{k+1} f(x_{n+k+1}) - \nabla^{k+1} f_{n+k+1}^{(p)}\right) \\ &\quad - \left(\bar{a} - \gamma_{k+1}^{*}\right) h^{k+2} y^{(k+2)}(x_n) \end{aligned}$$

$$
\begin{aligned}
&= \gamma_k h \left(f(x_{n+k}) - f_{n+k}^{(p)} \right) \\
&\quad + h\bar{a} \left(\left(f(x_{n+k+1}) - f_{n+k+1}^{(p)} \right) - (k+1) \left(f(x_{n+k}) - f_{n+k}^{(p)} \right) \right) \\
&\quad - \left(\bar{a} - \gamma_{k+1}^{*} \right) h^{k+2} y^{(k+2)}(x_n).
\end{aligned}
$$

If we put (9) and (10) into the previous equation we have:

$$
\begin{aligned}
\text{LTE} = \gamma_k h \frac{\partial f}{\partial y} \gamma_k h^{k+1} y^{(k+1)}(x_n) + h\bar{a} \frac{\partial f}{\partial y} \gamma_k (1+a) h^{k+1} y^{(k+1)}(x_n) \\
- h\bar{a}(k+1) \frac{\partial f}{\partial y} \gamma_k h^{k+1} y^{(k+1)}(x_n) - (\bar{a} - \gamma_{k+1}^{*}) h^{k+2} y^{(k+2)}(x_n).
\end{aligned}
$$

Thus, we finally obtain:

$$
\begin{aligned}
\text{LTE} &\equiv y(x_{n+k}) - y_{n+k}^{(c)} \\
&= [\bar{a}(1+a) + \gamma_k - (k+1)\bar{a}] \gamma_k h^{k+2} \frac{\vartheta f}{\vartheta y} y^{(k+1)}(x_n) \\
&\quad - \left(\bar{a} - \gamma_{k+1}^{*} \right) h^{k+2} y^{(k+2)}(x_n) + 0(k^{k+3})
\end{aligned}
$$

for every $k = 1, 2, \ldots$. (Note that in practice k may take values between 1 and 12) □

We notice that with the above predictor–corrector approach we have two free parameters at our disposal, namely a and $\bar{a}$. We can also see that the Shampine and Gordon code [20] corresponds to $a = \bar{a} = 0$.

Let us now consider the LTE for $k = 1 (\gamma_0 = 1, \gamma_1 = \frac{1}{2})$ from (7):

$$
\begin{aligned}
y(x_{n+1}) - y_{n+1}^{(c)} &= \left[\bar{a}(1+a) + \frac{1}{2} - 2\bar{a} \right] \cdot \frac{1}{2} h^3 f_y y_n'' - \left[\frac{1}{2} + \bar{a} \right] h^3 y_n''' \\
&= \left[\frac{1}{4} + \frac{\bar{a}}{2}(1+a) - \bar{a} \right] h^3 f_y y_n'' - \left[\frac{1}{12} + \bar{a} \right] h^3 y_n'''.
\end{aligned}
$$

As we just stated, the Adams case results from $\bar{a} = a = 0$ and has $A_1 = \frac{1}{4}$and $A_2 = -\frac{1}{12}$ where with A_1 and A_2 we symbolize the Adams coefficients of $f_y y_n''$ *and* y_n''', respectively. Assuming that our system is autonomous, i.e. $y' = f(y)$, we may substitute:

$$
\begin{aligned}
y'' &= f' = f_y f \\
y''' &= f'' = f_{yy} f^2 + f_y^2 f
\end{aligned}
$$

into the previous calculation and we get:

$$\begin{aligned} y(x_{n+1}) - y^{(c)}_{n+1} &= \left[\frac{1}{4} + \frac{\bar{a}}{2}(1+a) - \bar{a}\right] h^3 f_y^2 \cdot f - \left[\frac{1}{12} + \bar{a}\right] h^3 \left(f_{yy} f^2 + f_y^2 f\right) \\ &= \left[\frac{1}{6} + \frac{\bar{a}}{2}(a-3)\right] h^3 f_y^2 f - \left[\frac{1}{12} + \bar{a}\right] h^3 f_{yy} f^2. \end{aligned}$$

Now the Adams coefficients are:

$$A_1 = \frac{1}{6} \text{ and } A_2 = -\frac{1}{12}$$

where A_1 and A_2 are the coefficients of $f_y^2 f$ *and* $f_{yy} f^2$, respectively.

If we define the coefficients B_1 and B_2 from (7), as:

$$\begin{aligned} B_1 &= \left[\bar{a}(1+a) + \gamma_k - (k+1)\bar{a}\right] \gamma_k \\ B_2 &= -\left(\bar{a} - \gamma^*_{k+1}\right) \end{aligned} \tag{12}$$

and if we extend the previous analysis $\forall \;\; k$, we are able to **rewrite the LTE (7) in terms of elementary differentials as**:

$$\left.\begin{aligned} \text{LTE} &= (B_1 + B_2) h^{k+2} (\textit{elementary differ.})_1 \\ &\quad + B_2 h^{k+2} (\textit{elementary differ.})_2 \\ &= \mathrm{E}_1 h^{k+2} (\textit{elementary differ.})_1 \\ &\quad + \mathrm{E}_2 h^{k+2} (\textit{elementary differ.})_2 \end{aligned}\right\} \tag{13}$$

where $(\textit{elementary differ.})_i$, $i = 1$ or 2, are subsets of all elementary differentials of order $(k+2)$. The form we will use for the LTE of our scheme (4) will be, from now on, the one given in (13). The advantage of (13) compared to (7) is that it groups together like quantities and we would expect this to produce more reliable results for linear problems when estimating the LTE.

3.1 Local error estimation

Ideally we would like to obtain a cheap estimate of the local error expansion (13) and to base a step-size control procedure on this estimate. Unfortunately (13) is too complicated to estimate in a reliable and economical way. In view of this we adopted the well known Fehlberg embedding [21] approach where we estimate and control the error in a solution which is asymptotically less accurate than the one we actually accept. To do this we compute an approximation $\breve{y}_{n+k}$ at x_{n+k}, without additional function evaluations, using an **embedded formula** of the form:

$$\begin{aligned} \breve{y}_{n+k} = y_{n+k-1} &+ h \sum_{i=0}^{k-1} \gamma_i \nabla^i f_{n+k-1} + h(\tilde{a}_2 + (k+1)\tilde{a}_1) \nabla^k f^{(p)}_{n+k} \\ &+ h\tilde{a}_1 \nabla^{k+1} f^{(p)}_{n+k+1} \end{aligned} \tag{14}$$

where $\tilde{a}_1$ and $\tilde{a}_2$ are free parameters. For (14) we require that

$$y(x_{n+k}) - \breve{y}_{n+k} = 0(h^{k+1}) \tag{15}$$

and assuming h is small we can neglect $0(h^{k+2})$ compared with $0(h^{k+1})$. Thus our embedded formula (14) is of order k compared to (13) which is of order $(k+1)$.

The LTE of our embedded formula is given by:

$$\mathrm{LTE}_e \equiv y(x_{n+k}) - \breve{y}_{n+k} = h^{k+1}(\gamma_k - \tilde{a}_2 - (k+1)\tilde{a}_1)y^{(k+1)}(x_n) + 0(h^{k+2})$$

where the subscript e stands for "embedded".

Now we estimate the local error of EAS methods by subtracting (7) from (15):

$$\left.\begin{aligned}\text{Local Error} \approx y^{(c)}_{n+k} - \breve{y}_{n+k} = h(\gamma_k - \tilde{a}_2 - (k+1)\tilde{a}_1)\nabla^k f^{(p)}_{n+k}\\ + h(\bar{a} - \tilde{a}_1)\nabla^{k+1} f^{(p)}_{n+k+1}.\end{aligned}\right\} \tag{16}$$

If the norm of (16) is less than a prescribed tolerance we perform local extrapolation and accept $y^{(c)}_{n+k}$ as the final solution at x_{n+k}. Because of the particularly simple form of (16) we are able, besides controlling the step-size, to change order (discussed in sect. 7) in a straightforward way.

4 Stability and accuracy requirements

For our corrector formula, the coefficients to be chosen are a and $\bar{a}$, and for our embedded formula, which is used for local error estimation, the coefficients to be chosen are $\tilde{a}_1,\ \tilde{a}_2$. The choice of these coefficients will directly influence the stability and accuracy of our methods. Our investigations have shown that the simultaneous requirements of accuracy and high stability are often contradictory requirements. Stability depends on the choice of the free parameters $a, \bar{a}, \tilde{a}_1, \tilde{a}_2$ but, as we see from (12) to (13) and (16), these same parameters form the coefficients of the LTE, which means that accuracy is also influenced at the same time. If we do not pay attention to the magnitude of the LTE and we choose the free parameters so as to maximise stability, the result is that we may lose in accuracy. If, on the other hand, we try to restrict the size of the LTE and local error coefficients in order to gain in accuracy, then we may lose in stability. This observation led us to develop three different classes of EAS methods that will be discussed in sect. 5. It also means that any numerical search designed to obtain good accuracy and stability properties needs to be carried out very carefully.

In order to define our stability and accuracy requirements we need to find a theoretical basis on which to make our choices and thus restrict accordingly our error coefficients by assigning appropriate values to our free parameters.

Accuracy

Since EAS methods require 3 functions per step whereas the Adams predictor–corrector formulae in PECE mode require 2, we decided to *"scale" the computational effort* involved with our formulae. To explain how this is done let us consider the case $k = 1$. The LTE in terms of elementary differentials, as given in sect. 3, is for the EAS corrector formula:

$$y(x_{n+1}) - y_{n+1}^{(c)} = \left[\frac{1}{6} + \frac{\bar{a}}{2}(a-3)\right] h^3 f_y^2 f - \left[\frac{1}{12} + \bar{a}\right] h^3 f_{yy} f^2 + 0(h^4)$$

and for the well known 1-step Adams formula in PECE mode

$$\frac{1}{6} h^3 f_y^2 f - \frac{1}{12} h^3 f_{yy} f^2 + 0(h^4).$$

If we now integrate from, let us say, x_n to $x_n + 2h$ using the EAS method with a stepsize h the LTE will be for sufficiently small h:

$$\text{LTE} = 2\left[\frac{1}{6} + \frac{\bar{a}}{2}(a-3)\right] h^3 f_y^2 f - 2\left[\frac{1}{12} + \bar{a}\right] h^3 f_{yy} f^2 + 0(h^4).$$

Similarly if we integrate from x_n to $x_n + 2h$ using the Adams method with stepsize $2h/3$, so that both methods do the same amount of work (i.e. use six function evaluations), the LTE is:

$$3 \cdot \frac{1}{6}\left(\frac{2h}{3}\right)^3 f_y^2 f - 3\frac{1}{12}\left(\frac{2h}{3}\right)^3 f_{yy} f^2.$$

For the EAS methods to be asymptotically more accurate than the Adams methods we need (where the norms used can be any convenient norms)

$$\left\| \left[\frac{1}{6} + \frac{\bar{a}}{2}(a-3)\right] f_y^2 f - \left[\frac{1}{12} + \bar{a}\right] f_{yy} f^2 \right\| < \left(\frac{2}{3}\right)^2 \left\| \frac{1}{6} f_y^2 f - \frac{1}{12} f_{yy} f^2 \right\| ,$$

assuming that the $0(h^4)$ terms are negligible. This can be achieved by requiring

$$\left.\begin{array}{c} \frac{1/6+\frac{\bar{a}}{2}(a-3)}{\frac{1}{12}+\bar{a}} = \frac{1/6}{1/12} = 2 \\ \text{and } \left|\frac{1}{12} + \bar{a}\right| < \left(\frac{2}{3}\right)^2 \frac{1}{12} \end{array}\right\} \tag{17}$$

Generalising (17) for each step-number k, and keeping in mind that all EAS formulae have only 2 terms in their LTE, we have the following **two EAS accuracy requirements**:

$$(1) \quad \text{(EAS LTE coeff.)} < \left(\frac{2}{3}\right)^{k+1} \text{(Adams LTE coeff.)} \tag{18}$$

(2) Using the symbolism of (13) for the LTE, we want:

$$\frac{E_1}{E_2} \approx \frac{\tilde{E}_1}{\tilde{E}_2} \tag{19}$$

where $\tilde{E}_1$ and $\tilde{E}_2$ are the corresponding error coefficients of the conventional Adams P–C formulae.

We now have to consider what happens though to the free parameters $\tilde{a}_1$ and $\tilde{a}_2$ of the embedded formula. We note that by making appropriate choices for $\tilde{a}_1$ and $\tilde{a}_2$ we can make the leading term in the LTE of the embedded formula vanish. If $\tilde{a}_1$ and $\tilde{a}_2$ are chosen so that this error term has a large modulus, we can expect to have a good local error estimate, but a relatively inefficient integration. On the other hand, as this error term tends to zero, a more accurate solution is expected and a faster integration, but in general the error estimate gets worse and worse, and thus the integration becomes increasingly unreliable. This is a design difficulty frequently encountered with embedded error estimates. Shampine [22] investigated this problem for explicit R–K formulae and he suggested choosing formula coefficients so that the local error estimate mimics what happens when Richardson extrapolation is used to estimate the error. It would be useful to have a corresponding theory for multistep methods so we know how to choose the local error estimate. What we have chosen to do for our embedded formula is to apply condition (18) in this case also. This has a logical basis since we compare all our error coefficients with the corresponding ones of the Adams P–C approach. Hence, we obtain the following **accuracy requirement for the EAS embedded formula**:

$$(3)\quad \text{(EAS Local Error coeff.)} < \left(\frac{2}{3}\right)^{k+1} \times \text{(Adams Local Error coeff.)} \tag{20}$$

Stability

An obvious stability requirement for our corrector formula would be to have a "large" region of absolute stability. Of course, according to how we define the meaning of "large" we will obtain different formulas. For example, "large" could mean a "large" intercept on the real axis or a "large" intercept on the imaginary axis. On the other hand, the accuracy requirements above affect stability. Thus, we decided that since the Adams formulae require 2 functions per step while EAS formulae require 3 functions per step, we require that the stability interval of our method should be "at least $\frac{3}{2}$ times that of the conventional method". We may write our **principal stability requirement for the EAS corrector** as:

(4) For each specific step-number $k = 1, 2, \ldots, 12$ we should have :

$$\text{(Adams real stability interval)} \leq \frac{3}{2}\text{(EAS real stability interval)} \tag{21}$$

In practise the above requirement gives also a reasonably large intercept on the imaginary axis. Requirement (21) ensures that both approaches have roughly *the same stability per function evaluation*. Since (21) can not always be satisfied, we also focus on another important aspect of absolute stability, i.e. the area. Using the whole area of absolute stability as our criterion, we can formulate **two complementary stability requirements for the EAS corrector**, which can be summarised as follows:

$$(5)\quad \left.\begin{array}{l}\text{For each specific step-number } k = 1, 2, \ldots, 12 \text{ the EAS entire absolute}\\ \text{stability region area should be at least 1.5 times larger, compared to}\\ \text{the corresponding Adams total absolute stability region area.}\end{array}\right\} \quad (22a)$$

$$(6)\quad \left.\begin{array}{l}\text{For each specific step-number } k = 1, 2, \ldots, 12 \text{ we should maximise}\\ \text{the EAS entire absolute stability region area, subject to the}\\ \text{corresponding accuracy, LTE or local error coefficient, requirements.}\end{array}\right\} \quad (22b)$$

We note that requirement (22b) does not apply to EAS1, since the EAS1 scheme does not strictly follow (by design) the accuracy requirements (18), (19) and (20). Although requirement (21) is crucial and it is our principal stability requirement, we believe that the above two complementary area requirements (22a) and (22b) are also important and depending on the problem at hand, they could become vital.

As far as the embedded formula is concerned, it is not necessary for this formula to have good stability properties since its sole purpose is to provide an estimate of the local error. Because no extra computational work is needed, we require an embedded formula with "good" stability. By "good" stability we mean at least as large as the corrector's region of absolute stability (but not identical to it).

5 Three different EAS schemes

It now remains to examine the choice of the free parameters $a,\ \bar{a},\ \tilde{a}_1,\ \tilde{a}_2$ and to discuss the step-size and order control. The requirements of high stability and accuracy led us to develop three different families of EAS methods, with each one aiming at a different target, since our investigations have shown that the requirements of accuracy and high stability are often conflicting requirements. Thus, in order to bypass or relax these conflicting requirements, the following classes of EAS methods have been investigated:

1. **EAS1** is developed in order to get the best possible real absolute stability and area but the accuracy requirements (18), (19) and (20) are not strictly followed and thus its accuracy should be poorer compared with Adams formulae of the same order. However, as we will see in the next sections, there is a type of problems, namely mildly stiff problems, where EAS1 should perform better than Adams, due to its much superior regions of absolute stability.
2. **EAS2** is developed in order to have better stability than the Shampine and Gordon P–C scheme and at the same time at least as good accuracy. This class follows

Table 1 EAS schemes and our theoretical accuracy and stability requirements

Accuracy and stability requirements	EAS1	EAS2	EAS3
Accuracy-LTE-1 (18)	In part	✓	✓
Accuracy-LTE-2 (19)	In part	✓	✓
Accuracy-embedded (20)	In part	✓	✓
Stability-real axis (21)	✓	Limited	✓
Stability-area-1 (22a)	✓	✓	✓
Stability-area-2 (22b)	Does not apply	✓	✓

the accuracy requirements (18), (19) and (20) but problems arise with the stability (21). EAS2 is the class that we designed to be directly competitive with the Shampine scheme for general non-stiff problems. The EAS2 scheme has been thoroughly investigated in a separate paper [5].

3. **EAS3** is the class that maintains good accuracy and the best possible stability (better than EAS2). As we will see in a forthcoming article, this method constitutes an even larger departure from the Adams type of methods, compared to EAS1 and EAS2. EAS3 is the class that fulfils all accuracy and stability requirements (i.e. (18) through (22)).

All the above methods are designed to have a superior absolute stability region, ranging from dramatically better to slightly better, compared with Adams methods. If instead of using the real axis of absolute stability, i.e. (21), we do our comparisons using the whole area of the stability region for each order then the outcome of the stability region comparisons is drastically in favour of EAS methods, since even in the "worst" case, the EAS regions are at least double in area compared with Adams methods. If we put all six accuracy and stability requirements in a chart we get Table 1:

6 EAS1 methods

In EAS1 the P–C system (4) is used with the aim of deriving formulae with the largest possible real absolute stability region. The accuracy conditions (18), (19) and (20) are not always met but the stability requirements (21) and (22a) will always be satisfied. We should note that besides deriving formulae with a large real absolute stability region we could also derive formulae with a large imaginary stability interval as well (these would be particularly well suited for solving semi-discretized IVPs arising from hyperbolic partial differential equations). In Table 2 we list the free parameters of the EAS1 formulae, together with the corresponding LTE coefficients, for orders 1–12 that satisfy the restrictions previously described. The choice of the parameters $a, \bar{a}, \tilde{a}_1$ and $\tilde{a}_2$ is by no means straightforward since they have been selected from a large number of possibilities.

Table 2 EAS1 free parameters and error coefficients

Step-number	A	$\bar{a}$	$\tilde{a}_1$	$\tilde{a}_2$	E_1 from (13)	E_2 from (13)	Local error coef. (16)
$k = 1$	6.330	−0.03800	−0.0260	0.4280	0.10340	−0.045330	0.12400
$k = 2$	9.300	−0.02500	−0.0175	0.3910	0.08090	−0.016670	0.07817
$k = 3$	13.950	−0.01400	−0.0101	0.3470	0.07075	−0.012390	0.06840
$k = 4$	15.050	−0.01650	−0.0125	0.3590	0.05572	−0.002250	0.05211
$k = 5$	14.570	−0.02520	−0.0196	0.4090	0.04019	0.010931	0.03846
$k = 6$	22.800	−0.01320	−0.0098	0.3545	0.03145	0.001833	0.02969
$k = 7$	28.900	−0.01040	−0.0082	0.3470	0.02431	0.001043	0.02282
$k = 8$	37.000	−0.00880	−0.0080	0.3560	0.01260	0.000907	0.01087
$k = 9$	43.000	−0.00730	−0.0064	0.3440	0.01164	0.000514	0.00698
$k = 10$	48.000	−0.00640	−0.0056	0.3360	0.01084	0.000476	0.00579
$k = 11$	55.500	−0.00550	−0.0052	0.3350	0.00836	0.000263	0.00167
$k = 12$	62.500	−0.00480	−0.00475	0.3300	0.00729	0.000122	0.00078

6.1 EAS1 stability regions

Before presenting the stability region graphs, it is worth saying something about the method that these graphs were obtained. The way these graphs were obtained was by no means trivial and the difficulty of this task is something usually underestimated or completely overlooked. Let us consider the case for $k = 1$. Making the appropriate algebraic manipulations [4] and using the test equation $y' = \lambda y$ we have:

$$y_{n+1}^{(p)} = y_n + hf_n$$

where $f_n = \lambda y_n$

$$y_{n+2}^{(p)} = y_n + h\left[\left(\frac{3}{2} - \frac{1}{2}a\right) f_{n+1}^{(p)} + \left(\frac{1}{2}a + \frac{1}{2}\right) f_n\right]$$

where $f_{n+1}^{(p)} = (\lambda + h\lambda^2)y_n$

$$y_{n+1}^{(c)} = y_n + h\left[\bar{a} f_{n+2}^{(p)} + \left(\frac{1}{2} - 2\bar{a}\right) f_{n+1}^{(p)} + \left(\frac{1}{2} + \bar{a}\right) f_n\right] y_n$$

where $f_{n+2}^{(p)} = \left[\lambda + h\lambda\left(\frac{3}{2} - \frac{1}{2}a\right) f_{n+1}^{(p)} + \left(\frac{1}{2}a + \frac{1}{2}\right) h\lambda^2\right] y_n$ and we finally get

$$y_{n+1}^{(c)} = \left[1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \bar{a}\left(\frac{3}{2} - \frac{1}{2}a\right)(h\lambda)^3\right] y_n. \tag{23}$$

If we let $h\lambda = x + iy$ and we separate the real from the imaginary part, then (23) becomes:

$$y_{n+1}^{(c)} = \left[\left(1 + x + \frac{1}{2}(x^2 - y^2) + \bar{a} \left(\frac{3}{2} - \frac{1}{2}a \right) (x^3 - 3xy^2) \right) \right.$$
$$\left. + i \left(y + xy + \bar{a} \left(\frac{3}{2} - \frac{1}{2}a \right) (3x^2 y - y^3) \right) \right] y_n.$$

Thus, the general k-step corrector using the customary test equation etc can be written in the following form:

$$y_{n+k}^{(c)} = \sum_{j=0}^{k-1} ([A_j + i B_j] y_{n+j}) \tag{24}$$

where A_j represents the real part and B_j the imaginary part of the coefficients of y_{n+j}.

Since we can write the corrector for each k in the form (24), this means that we can separate the real from the imaginary part in each case. Then we scan for x and y and accept the point (x, y) if and only if the polynomial at hand meets the *Schur criteria* or in other words is a *Schur polynomial*. Consider the nth degree polynomial

$$\Pi(t) = q_n t^n + q_{n-1} t^{n-1} + \cdots + q_1 t + q_0$$

where $q_n \neq 0,\ q_0 \neq 0$. The polynomial $\Pi(t)$ is said to be a Schur polynomial if its roots t_s satisfy $|t_s| < 1$.

Schur's Theorem *A polynomial Π (t) is a Schur polynomial if and only if:*

(1) $|\hat{\Pi}(0)| > |\Pi(0)|$ *and*
(2) Π_1*(t) is a Schur polynomial.*

Where

$$\hat{\Pi}(t) = q_0^* t^n + q_1^* t^{n-1} + \cdots + q_{n-1}^* t + q_n^*$$
$$\Pi_1(t) = \frac{1}{t}\{\hat{\Pi}(0)\Pi(t) - \Pi(0)\hat{\Pi}(t)\}$$

q_i^* is the complex conjugate of q_i and Π_1(t) is a polynomial of degree $\leq t - 1$.

If the conditions given by *Schur's theorem* are applied $n - 1$ times they yield $n - 1$ inequalities and the requirement that a polynomial of degree 1 should be a Schur polynomial. In order to implement these conditions we had to create a special subroutine that applies the Schur criteria for polynomials up to order 12. This technique, which we will call the scanning technique, we believe it has a considerable advantage compared to the boundary locus technique (see [22, p.71–72]) in the case that the region of absolute stability has two or more non-connected parts, such as the Runge–Kutta DOPRI (5,4) method. The disadvantage of this technique is that it becomes very complicated as the degree of the polynomial rises. All the EAS1 scheme graphs in this paper are obtained through this technique.

For the very first time in an article, the graphs of the stability regions for the EAS1 corrector, the embedded and the corresponding well known Adams corrector are given in Figs. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 (three similar but not identical figures to Figs. 1, 2 and 3 appeared in [3]). We observe from these figures that EAS1 formulae have a real stability interval which is about 3–4 times that of Shampine's P–C scheme. This is a safe indication that EAS1 methods would be much more efficient than Adams method in cases where the step-size of integration is restricted by the real stability region, i.e. in the case of mildly stiff problems with real eigenvalues. Each different line shown in the figures below represents one of the formulae.

7 Implementation of the EAS1 scheme

This section deals with the way in which EAS1 formulae are implemented in a variable step/variable order mode. It should be noted that although we investigate the implementation of EAS1 scheme, the implementation logic and the computer code are the same for the EAS2 implementation as well (including some necessary modifications) [5]. The description will be brief since Shampine and Gordon's code has been a great influence on our implementation. For example, algorithms for interpolating solutions at off-step points or decreasing and increasing order are exactly as described in [20]. Even an attempt to briefly describe the features of Shampine and Gordon's code would be totally beyond the scope of this work and for a description of the implementation of Adams formulae the reader is referred to [20]. Needless to say, if any modifications

a) EAS1 Corrector ————————
b) Embedded (EAS1) - - - - - - - - - - - - - - - -
c) Adams Corrector — — — — — — — — — — —

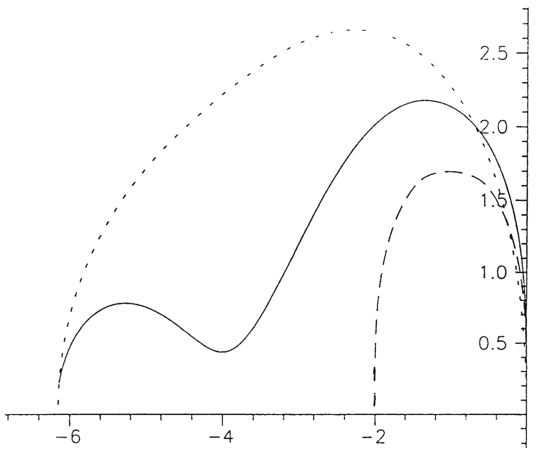

Fig. 1 EAS1 corrector-embedded and Adams corrector stability regions for $k = 1$

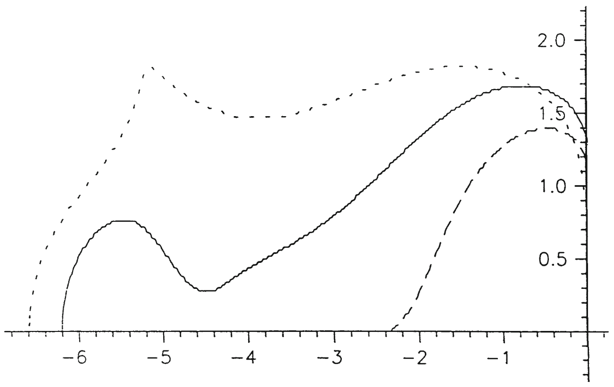

Fig. 2 EAS1 corrector-embedded and Adams corrector stability regions for $k = 2$

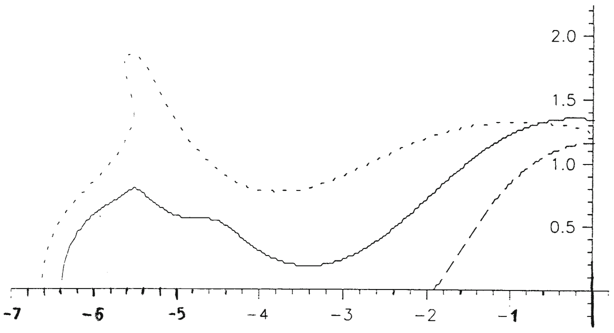

Fig. 3 EAS1 corrector-embedded and Adams corrector stability regions for $k = 3$

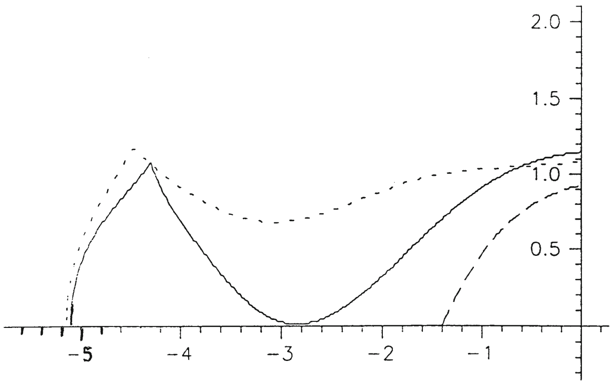

Fig. 4 EAS1 corrector-embedded and Adams corrector stability regions for $k = 4$

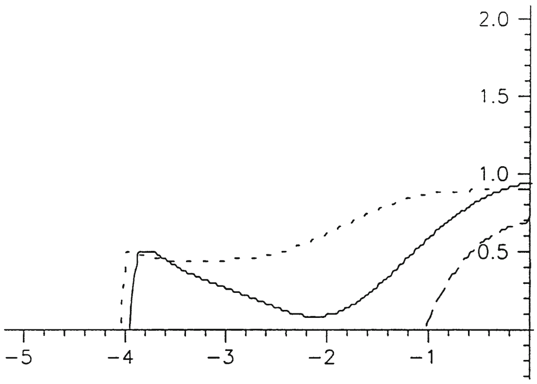

Fig. 5 EAS1 corrector-embedded and Adams corrector stability regions for $k = 5$

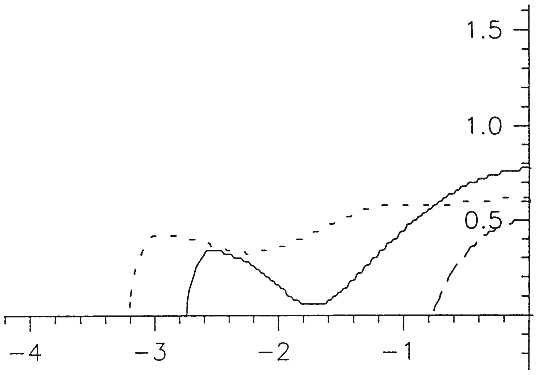

Fig. 6 EAS1 corrector-embedded and Adams corrector stability regions for $k = 6$

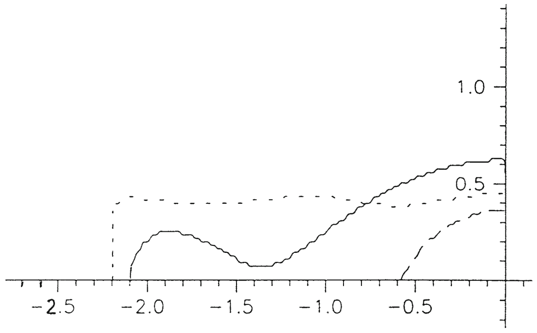

Fig. 7 EAS1 corrector-embedded and Adams corrector stability regions for $k = 7$

have been applied to the heavily reprogrammed Shampine and Gordon code after we obtained our numerical results (sect. 8) some time ago, this is not the concern of this paper. As well as many similarities in the implementation of the Adams and EAS approaches there are also several differences, which will be briefly described below.

Shampine's code uses divided differences for the storing of information. Instead of doing this we chose to use backward differences to store the history array, which

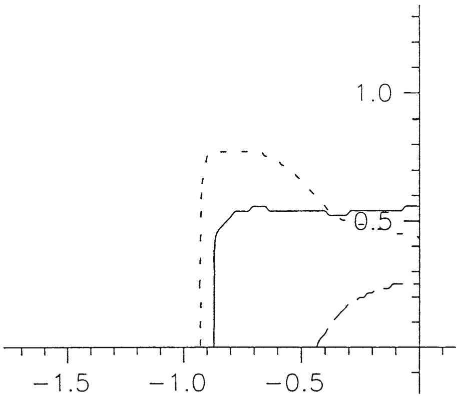

Fig. 8 EAS1 corrector-embedded and Adams corrector stability regions for $k = 8$

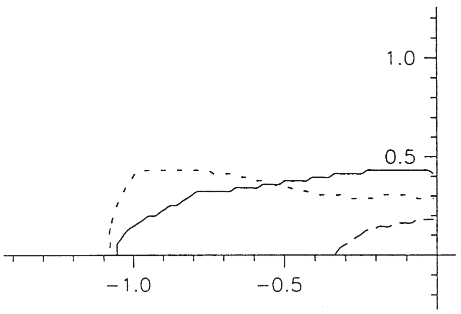

Fig. 9 EAS1 corrector-embedded and Adams corrector stability regions for $k = 9$

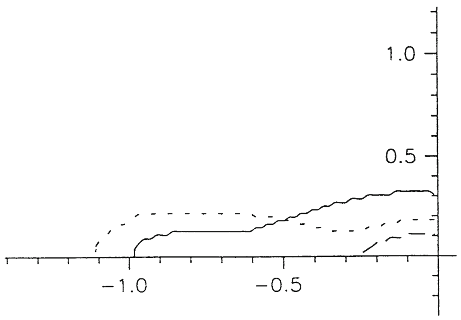

Fig. 10 EAS1 corrector-embedded and Adams corrector stability regions for $k = 10$

represents a rather big departure from the conventional approach. The stored array, when using a k-step formula with a current stepsize h, has the form:

$$\left(y_n,\ hy'_n,\ h\nabla y'_n,\ h\nabla^2 y'_n, \ldots, h\nabla^{k+1} y'_n\right)^T$$

We will use a *quasi-constant stepsize* implementation and backward differences are very efficient for such situations. In order to make clear how the history array is updated let us say that for the information stored at x_{n-1} and x_n, if for example we

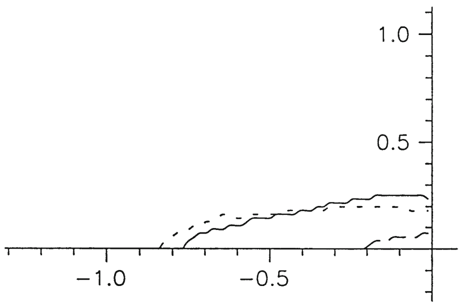

Fig. 11 EAS1 corrector-embedded and Adams corrector stability regions for $k = 11$

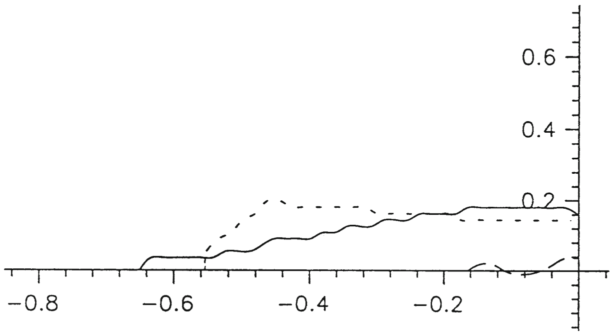

Fig. 12 EAS1 corrector-embedded and Adams corrector stability regions for $k = 12$

want $h\nabla y'_n = h(y'_n - y'_{n-1})$, we just subtract the already computed results, and in general:

$$\nabla^k y'_n = \nabla^{k-1} y'_n - \nabla^{k-1} y'_{n-1}.$$

The local error estimate (16) can be computed using stored information. The quantity $h\nabla^k f^{(p)}_{n+k}$ is stored in the history array, while $h\nabla^{k+1} f^{(p)}_{n+k+1}$ may also be computed from already existing information in the history array:

$$h\nabla^{k+1} f^{(p)}_{n+k+1} \equiv hf^{(p)}_{n+k+1} - h\sum_{i=0}^{k} \nabla^i f^{(p)}_{n+k}.$$

The local error check is done after the corrected solution has been computed.

As far as the starting of the integration is concerned, the ideas of Shampine and Gordon are broadly followed. While being in this initial phase we try to change order and stepsize at every integration step. The difference with Shampine's code lies in the fact that we compute also a stepsize for the next higher order. To explain our starting procedure we suppose that our current order is *RD* with a step $\tilde{h}$. We compute, via our local error estimates, the steps $\bar{h}$ and h that can be used for the next step at orders *RD*+1 and *RD* , respectively. The procedure for choosing order and stepsize is as follows:

```
IF $\bar{h} > h$ or $\bar{h} > 5\tilde{h}$ THEN
  IF $\bar{h} < 2\tilde{h}$ THEN
    $h = \tilde{h}$
    get out of starting phase
  ELSE
  IF RD $\neq$ Maximum order THEN
    RD = RD + 1
    $h = 2\tilde{h}$
  ELSE
    $h = 2\tilde{h}$
  ENDIF
ELSE IF $h > 2\tilde{h}$ THEN
        $h = 2\tilde{h}$
      $RD = RD - 1$
  ELSE
      $h = \tilde{h}$
      get out of starting phase
  ENDIF
ENDIF
```

The integration is continued with step h and order *RD*. We allow step doubling when a step is increased, whereas decreasing the step is permitted by a factor between 0.5 and 1.

EAS formulae are not defined for non-constant stepsize and thus we do not know how the stability of the formulae behaves for truly variable steps. We work therefore with a quasi-uniform step and we use interpolation to change h. This standard approach is often called the *fixed coefficient implementation*. Providing that the norm of (16) is less than a prescribed tolerance:

$$\left\| (\gamma_k - \tilde{a}_2 - (k+1)\,\tilde{a}_1)\, h\nabla^k f_{n+k}^{(p)} + (\bar{a} - \tilde{a}_1)\, h\nabla^{k+1} f_{n+k+1}^{(p)} \right\| < \varepsilon$$

where ε is the prescribed tolerance, using a k-step formula, we allow changes in stepsize to be made after $k+2$ steps and the decisions for changing (or not changing) order and/or stepsize are as described in [20]. If the prescribed tolerance is not satisfied we replace the derivatives by those at the previous step and we scale h by an appropriate factor. If more than 7 successive failures occur we decrease h by a factor of 10 and we start at order 1. When changing from a current stepsize h to a new one αh, we interpolate the existing data so as to get values appropriate for working with the new fixed step αh. As far as change of order is concerned, we compute

$$C_{k-1}\nabla^{k-1} y'_{n+k},\;\; C_k \nabla^k y'_{n+k} \; and \; C_{k+1}\nabla^{k+1} y'_{n+k}$$

where C_{k-1} is the LTE of the predictor of order $k-1$, and we accept the smallest norm of the three. In the process of implementing the EAS1 scheme the code used in [24] was employed as a starting point and among other things, we had to change the interpolation subroutine and the main integration subroutine of that code.

8 Numerical results and comparisons

We would expect EAS1 formulae with their excellent stability to perform better than Adams formulae on mildly stiff problems. In order to investigate to what extent these expectations are true we give the results of appropriate numerical comparisons below.

We should stress that the EAS1 code is in an embryonic stage. This means that it could improve in performance if reprogrammed and in the light of further experience. Until we performed our numerical comparisons a while ago, the Shampine code had been reprogrammed many times and this had led to a much more efficient version than the one first appeared in 1975. A problem that we encountered with the EAS code is that the numerical results do not improve as the order rises above $k = 8$. We attempted to rectify this but had been unable to identify the programming inefficiency. Although the EAS1 scheme have been developed up to and including $k = 12$, as the Adams formulae have, we had to restrict the order of both codes to $k = 8$ in order to be able to obtain fair comparisons. Thus, in what follows the highest permitted order is 8 for both the EAS1 formulae as well as for the Adams formulae.

8.1 The DETEST programme

Enright and Pryce in order to facilitate comparison processes between different codes, developed the DETEST computer programme [25], which provides a set of test problems as well as the facility to compare different methods using the "normalized efficiency results" option.

Different codes are based on different discretization methods and thus they commit different errors. Codes for IVPs control an estimate of the local error and not, in general, an estimate of the global error, the magnitude of which we do not know. Generally it is hoped that the global error will be in the range of the error tolerance specified by the user. That is if we keep local truncation error suitably banded then the global error will also be banded. In practice though, it is normally the case that the global error is larger than the specified local error tolerance. When two different codes are applied to the same problem with the same local tolerance they may, at the same time, have a quite different global error. This difference in the global error makes the comparison of different codes a rather difficult issue and that is why it is not sufficient to make a comparison by counting only the number of function evaluations, the number of steps taken to complete the integration and the time required for the integration. The DETEST program provides us with a reliable estimate of the maximum global error over the whole range of integration. This facility allows DETEST to compute:

$$\text{Maximum Global Error} = \max \| y\,(x_n) - y_n \|, \quad n = 1, 2, \ldots$$

where y_n is the numerical solution generated by the user's code and $y(x_n)$ is the true solution at the point x_n. DETEST is able to compute an accurate approximation to both the maximum global error and the global error at the endpoint by implementing a very accurate IVP integration method that runs at a stricter tolerance than the one prescribed by the user. DETEST has also the ability to display information about the amount of work needed to achieve a specified accuracy, i.e. to keep the maximum global error less or equal to a given bound. Consequently the comparisons between different codes are much fairer because the numerical results correspond to solutions with the same maximum global error. DETEST refers to such results as normalized efficiency results. The problems for EAS1 and Adams were run at local tolerances of 10^{-2}, $10^{-3}, \ldots, 10^{-9}$. DETEST produces two types of result tables after the successful termination of the integration. These tables are given in the numerical comparisons.

The non-normalized results tables, i.e. the odd numbered tables, are not essential for our numerical comparisons and we provide them for the sake of completeness. These tables provide the following information: The local tolerance "LOG10 TOL" required, the time in seconds which was needed to solve the problem at a given tolerance, the overhead "OVHD" in seconds (this is equal to the time taken for solving a problem minus the time taken for the function evaluations), the number of function calls "FCN CALLS", the number of integration steps taken "NO OF STEPS", the end point global error, the maximum global error and the maximum local error.

The normalized efficiency results tables are the only results we need for our comparisons. These tables include information of the time required for the solver to achieve a maximum global error less than or equal to the "EXPECTED ACCURACY", the respective overhead, the number of function evaluations and the number of integration steps taken. The logarithm of the estimate of the local tolerance that the solver should be supplied with in order to keep the maximum global error less than or equal to the "EXPECTED ACCURACY", is given by the column "EQUIV LOG10 TOL". If the DETEST's intrinsic solver fails to complete successfully the integration for a given tolerance then the amount of output displayed for the normalized efficiency tables may vary.

8.2 Mildly stiff problems: EAS1 versus Adams

In this section we obtain numerical results for EAS1 and Adams IVP solvers on two mildly stiff problems. The problems and the numerical results are given below in Tables 3 through 10.

Problem 1 This is an artificial test problem of the form

$$\frac{dy}{dx} = -\lambda y + (\lambda - 1)e^{-x}, \quad y(0) = 1$$

that has the solution

$$y(x) = e^{-x}.$$

The problem becomes increasingly stiff as λ increases. We ran the two codes on this problem with (a) $\lambda = 10$, (b) $\lambda = 100$. The absolute accuracy requirement is for 10^{-2} to 10^{-9} and the range of integration is [0, 20]. Please note that for $\lambda = 1$ Problem 1 is non-stiff.

Table 3 Non-normalized results for the mildly stiff problem 1, $\lambda = 10$

LOG10 TOL	Time	OVHD	FCN CALLS	NO OF STEPS	END PNT GLB ERR	MAXIMUM GLB ERR	MAXIMUM LOC ERR
EAS1							
−2.00	.010	.002	166	50	.71	1.31	1.282
−3.00	.013	.005	169	52	1.18	1.20	1.203
−4.00	.018	.008	206	67	5.16	5.16	5.207
−5.00	.020	.008	241	81	1.65	1.65	1.598
−6.00	.023	.010	266	92	2.20	2.20	2.104
−7.00	.033	.016	355	122	4.71	4.71	4.716
−8.00	.038	.017	405	137	4.35	61.60	64.528
−9.00	.040	.019	426	151	.11	.57	.532
ADAMS							
−2.00	.035	.022	271	127	2.59	3.73	3.687
−3.00	.033	.021	247	115	2.73	5.86	6.741
−4.00	.038	.024	275	128	.17	6.68	7.508
−5.00	.038	.024	284	136	.43	3.59	4.004
−6.00	.047	.030	349	168	.12	6.35	7.303
−7.00	.044	.029	319	152	.50	4.30	5.044
−8.00	.053	.034	382	186	2.02	5.32	6.907
−9.00	.063	.041	456	222	.25	4.65	5.390

Table 4 Normalized efficiency results for problem 1, $\lambda = 10$

EXPECTED ACCURACY	EQUIV LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS
EAS1					
10** −2	−2.23	.011	.002	166	50
10** −3	−3.30	.015	.006	180	56
10** −4	−4.38	.019	.008	219	72
10** −5	−5.46	.022	.009	252	86
10** −6	−6.54	.029	.013	313	108
10** −7	−7.61	.036	.017	385	131
10** −8	−8.69	.040	.019	419	146
ADAMS					
10** −2	−2.69	.034	.021	254	118
10** −3	−3.69	.037	.023	266	123
10** −4	−4.69	.038	.024	281	133
10** −5	−5.69	.044	.028	329	158
10** −6	−6.70	.045	.029	328	156
10** −7	−7.70	.051	.033	362	175
10** −8	−8.70	.060	.039	433	211

Problem 1 for $\lambda = 10$ can be regarded as mildly stiff. Looking at the normalized efficiency results (Table 4) we see that the EAS1 code is clearly faster and also more accurate than the Shampine & Gordon code.

For the value $\lambda = 100$, Problem 1, can be regarded as being stiff. Observing the normalized efficiency results (Table 6) we see that the EAS1 code is much faster and more accurate than the Shampine and Gordon code. We may also see that the "TIME" ratio between EAS1/Adams is much more in favour of EAS1 than it was for $\lambda = 10$.

Problem 2 This is a partial differential equation of the form:

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial y^2}, \quad 0 \leq y \leq 1, \quad x \geq 0,$$
$$u(y, 0) = \cos(y), \;\; u(0, x) = e^{-x}, \;\; u(1, x) = e^{-x}\cos(1).$$

We semidiscretized this problem, using the method of lines [26], in order to obtain a system of IVPs. We took $h = 1/10$ and replaced the second derivative term by the well known central differences, to obtain a system of nine IVPs,

$$\frac{du}{dx} = A\underline{u} + \underline{b}\ ,$$

Table 5 Non-normalized results for the mildly stiff problem 1, $\lambda = 100$

LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS	END PNT GLB ERR	MAXIMUM GLB ERR	MAXIMUM LOC ERR
EAS1							
−2.00	.111	.032	1585	465	.38	2.28	2.284
−3.00	.094	.024	1413	418	.71	2.14	2.100
−4.00	.117	.033	1686	481	.46	2.09	2.053
−5.00	.117	.036	1633	480	.61	3.19	3.166
−6.00	.122	.040	1666	507	.24	2.60	2.582
−7.00	.133	.048	1714	530	.30	2.93	2.887
−8.00	.150	.058	1864	593	1.19	3.33	3.329
−9.00	.150	.060	1815	577	.48	2.98	2.990
ADAMS							
−2.00	.383	.247	2749	1294	1.14	5.94	6.581
−3.00	.367	.232	2715	1274	5.94	6.34	6.524
−4.00	.350	.220	2612	1212	.62	6.11	6.957
−5.00	.333	.204	2605	1209	1.51	6.66	7.317
−6.00	.367	.232	2710	1265	1.53	8.43	9.402
−7.00	.350	.217	2688	1258	.33	6.70	6.904
−8.00	.383	.242	2847	1357	.04	6.59	6.913
−9.00	.417	.273	2903	1378	1.28	7.61	7.828

Table 6 Normalized efficiency results for problem 1, $\lambda = 100$

EXPECTED ACCURACY	EQUIV LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS
EAS1					
10** −2	−2.34	.105	.030	1525	448
10** −3	−3.37	.103	.028	1514	441
10** −4	−4.40	.117	.034	1665	480
10** −5	−5.42	.119	.037	1646	491
10** −6	−6.45	.127	.043	1687	517
10** −7	−7.47	.141	.053	1785	559
10** −8	−8.50	.150	.059	1839	585
ADAMS					
10** −2	−2.80	.370	.235	2721	1278
10** −3	−3.81	.353	.223	2631	1223
10** −4	−4.82	.336	.207	2606	1209
10** −5	−5.83	.361	.228	2692	1255
10** −6	−6.85	.353	.219	2691	1259
10** −7	−7.86	.379	.239	2824	1343
10** −8	−8.87	.412	.269	2895	1375

Table 7 Non-normalized results for the mildly stiff problem 2, $0 \le x \le 5$

LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS	END PNT GLB ERR	MAXIMUM GLB ERR	MAXIMUM LOC ERR
EAS1							
−2.00	.350	.284	1307	391	.05	1.00	.993
−3.00	.383	.315	1358	400	.16	.94	.920
−4.00	.467	.384	1649	466	1.50	1.50	1.281
−5.00	.550	.468	1625	486	.70	1.21	1.193
−6.00	.583	.504	1573	497	1.09	1.42	1.415
−7.00	.717	.625	1817	602	7.42	1.41	1.408
−8.00	.783	.687	1922	648	67.42	1.31	1.284
ADAMS							
−2.00	1.233	1.099	2673	1257	.82	2.93	2.922
−3.00	1.150	1.020	2589	1211	.16	2.88	2.981
−4.00	1.117	.989	2529	1180	.62	3.29	3.513
−5.00	1.150	1.019	2595	1205	.37	3.75	4.186
−6.00	1.167	1.035	2608	1197	1.06	3.11	3.236
−7.00	1.233	1.096	2727	1296	6.87	3.21	3.262
−8.00	1.317	1.174	2838	1350	67.06	4.01	4.416
−9.00	1.417	1.266	2988	1439	674.73	2.73	2.957

where $\underline{u} = (u_1, u_2, \ldots, u_9)^T$, $\underline{b} = (1/h^2)(e^{-x}, 0, 0, \ldots, e^{-x}\cos(1.0))$, and A is the usual tridiagonal matrix. This problem was solved using the two codes in the integration ranges (a) $0 \le x \le 5$ and (b) $0 \le x \le 20$. The results are given in Tables 7, 8, 9 and 10.

Table 8 Normalized efficiency results for problem 2, $0 \le x \le 5$

EXPECTED ACCURACY	EQUIV LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS
EAS1					
10** −2	−2.03	.351	.285	1308	391
10** −3	−3.05	.388	.319	1372	403
10** −4	−4.07	.473	.390	1647	467
10** −5	−5.09	.553	.471	1620	486
10** −6	−6.11	.598	.517	1599	508
10** −7	−7.13	.725	.633	1830	607
10** −8	−8.14	.795	.698	1934	651
ADAMS					
10** −2	−2.49	1.192	1.060	2631	1234
10** −3	−3.50	1.133	1.005	2559	1195
10** −4	−4.50	1.133	1.004	2562	1192
10** −5	−5.51	1.158	1.027	2601	1200
10** −6	−6.51	1.201	1.066	2668	1247
10** −7	−7.52	1.276	1.136	2784	1323
10** −8	−8.52	1.369	1.222	2916	1396

Table 9 Non-normalized results for the mildly stiff problem 2, $0 \le x \le 20$

LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS	END PNT GLB ERR	MAXIMUM GLB ERR	MAXIMUM LOC ERR
EAS1							
−2.00	1.583	1.308	5469	1549	.01	1.00	.993
−3.00	1.367	1.119	4920	1498	.07	.94	.920
−4.00	1.967	1.620	6881	1936	.56	1.09	1.100
−5.00	2.050	1.704	6866	1959	.79	1.21	1.193
−6.00	2.067	1.726	6764	1955	.58	1.42	1.415
−7.00	2.017	1.703	6223	1838	.10	1.41	1.408
−8.00	2.367	2.013	7021	2134	2.08	2.08	1.802
−9.00	2.550	2.204	6879	2149	1.06	1.38	1.382
ADAMS							
−2.00	4.750	4.221	10513	4923	1.26	3.92	4.322
−3.00	4.883	4.359	10405	4874	.16	3.40	3.865
−4.00	4.633	4.119	10225	4787	.00	3.92	4.367
−5.00	4.667	4.147	10323	4820	.28	3.75	4.186
−6.00	4.783	4.258	10428	4873	1.48	3.37	3.740
−7.00	4.867	4.335	10551	4956	1.51	3.65	3.910
−8.00	4.833	4.302	10547	4958	.34	6.63	6.548
−9.00	4.967	4.427	10713	5045	.33	3.80	4.265

Table 10 Normalized efficiency results for problem 2, $0 \leq x \leq 20$

EXPECTED ACCURACY	EQUIV LOG10 TOL	TIME	OVHD	FCN CALLS	NO OF STEPS
EAS1					
10** −2	−1.98	1.588	1.312	5481	1550
10** −3	−3.01	1.376	1.126	4949	1504
10** −4	−4.05	1.971	1.625	6880	1937
10** −5	−5.09	2.052	1.706	6856	1958
10** −6	−6.13	2.060	1.723	6693	1939
10** −7	−7.17	2.076	1.755	6357	1887
10** −8	−8.21	2.405	2.052	6991	2137
ADAMS					
10** −2	−2.56	4.824	4.298	10452	4895
10** −3	−3.57	4.741	4.222	10302	4824
10** −4	−4.58	4.653	4.135	10282	4806
10** −5	−5.60	4.737	4.214	10385	4851
10** −6	−6.61	4.835	4.306	10503	4923
10** −7	−7.63	4.846	4.315	10548	4957
10** −8	−8.64	4.919	4.383	10653	5013

For an integration range between $0 \leq x \leq 5$ we clearly see that the EAS1 code is much faster and also more accurate than the Shampine/Gordon code (Table 8). This is in accordance with the results obtained for Problem 1.

For the larger integration range $0 \leq x \leq 20$ we see that EAS1 performs even better than before compared to Adams (Table 10). This can be seen from the "TIME" ratio EAS1/Adams which is more in favour for EAS1 than it was before.

9 Conclusions

In this work we comprehensively examined, for the very fist time in a paper, the EAS numerical multistep methods designed for the solution of mildly stiff and non-stiff IVPs. Their accuracy was carefully considered and their LTE was expressed both in the conventional form, as well as and in their elementary differential mode. The local error estimation of EAS schemes was also systematically investigated. Expressing the LTE in an elementary differential form proved central in establishing our basis for developing six theoretical accuracy and stability requirements. In turn, the construction of the accuracy and stability requirements led us to develop three distinct EAS schemes, each one targeting different types of problems and/or fulfilling different sets of requirements, and all of them competing with the well established Adams P–C formulae (in the form of the Shampine and Gordon code). Then we studied in some detail the EAS1 scheme and we presented its superior regions of absolute stability. The implementation of the EAS1 formulae in a variable step/variable order code was properly considered and the numerical results and comparisons for mildly stiff problems were meticulously presented. The EAS1 code performs much better than the Shampine code. The numerical results showed (sect. 8.2) that the EAS1 scheme is more accurate and much faster than the Adams formulae in their Shampine and Gordon code implementation. This is not surprising since the EAS1 scheme possesses much larger absolute stability regions than the Adams formulae. In mildly stiff problems, where stability is vital, the EAS1 code performed 3–4 times faster and also more accurately.

Despite the much better performance of EAS1 scheme on mildly stiff problems, we need to keep in mind though that the EAS1 code used in this work is in need of much further development, whereas the Shampine and Gordon code has been reprogrammed and optimised for many years. The major point here is the existence of the EAS1 methods, which could represent a first-class research alternative. We hope that this work may encourage some further research on the EAS schemes and similar type of methods.

References

1. G. Psihoyios, Advanced step-point methods for the solution of initial value problems. PhD Thesis, University of London—Imperial College of Science, (1995)
2. J.R. Cash, [G-]Y. Psihoyios, Advanced step-point methods for initial value problems, in *Proceedings of the Third International Colloquium on Numerical Analysis*, ed. by D. Bainov, V. Covachev, (Book publisher: VSP/Brill-Holland (ISBN 9067641936) 1995), pp. 43–50
3. J.R. Cash, [G-]Y. Psihoyios, An efficient class of general linear methods for non-stiff initial value problems, in *Proceedings of HERMIS 94*, ed. by E.A. Lipitakis, (Book publisher: Hellenic Mathematical Society (ISBN 11059737) 1995), pp. 813–820
4. G. Psihoyios, Some general formulae for the stability function of explicit advanced step-point (EAS) methods. Math. Comput. Model. (Elsevier, ISSN 08957177) **40**(11–12), 1171–1179 (2004)
5. G. Psihoyios, Explicit advanced step-point (EAS) methods and the EAS2 multistep scheme for the solution of non-stiff initial value problems. Appl. Math. Comput. (Elsevier, ISSN 00963003) **209**(1), 106–124 (2009)
6. G. Psihoyios, T.E. Simos, Exponentially & trigonometrically-fitted explicit advanced step-point (EAS) methods for IVPs with oscillating solutions. Int. J. Mod. Phys. C (World Scientific, ISSN 01291831) **14**(2), 175–184 (2003)
7. G. Psihoyios, T.E. Simos, Trigonometrically-fitted Adams–Bashforth–Moulton methods for periodic initial value problems, in *Computational Fluid and Solid Mechanics 2003 (Proceedings of the 2nd MIT Conference on Computational Fluid and Solid Mechanics, June 2003, MIT, USA)*, ed. by K.J. Bathe, (Book publisher: Elsevier Science (ISBN 0080440487) 2003), pp. 2097–2100
8. G. Psihoyios, T.E. Simos, Trigonometrically-fitted predictor–corrector methods for IVPs with oscillating solutions. J. Comput. Appl. Math. (Elsevier, ISSN 03770427) **158**(1), 135–144 (2003)
9. G. Psihoyios, T.E. Simos, A fourth algebraic order trigonometrically fitted predictor–corrector scheme for IVPs with oscillating solutions. J. Comput. Appl. Math. (Elsevier, ISSN 03770427) **175**(1), 137–147 (2005)
10. G. Psihoyios, T.E. Simos, A family of fifth algebraic order trigonometrically fitted P–C schemes for the numerical solution of the radial Schrödinger equation. MATCH—Commun. Math. Comput. Chem. (U. of Kragujevac, ISSN 03406253) **53**(2), 321–346 (2005)
11. G. Psihoyios, T.E. Simos, Sixth algebraic order trigonometrically fitted predictor–corrector methods for the numerical solution of the radial Schrödinger equation. J. Math. Chem. (Springer, ISSN 02599791) **37**(3), 295–316 (2005)
12. G. Psihoyios, T.E. Simos, A new trigonometrically fitted sixth algebraic order P–C algorithm for the numerical solution of the radial Schrödinger equation. Math. Comput. Model. (Elsevier, ISSN 08957177) **42**(7–8), 887–902 (2005)

13. G. Psihoyios, T.E. Simos, The numerical solution of the radial Schrödinger equation via a trigonometrically fitted family of seventh algebraic order predictor–corrector methods. J. Math. Chem. (Springer/Kluwer, ISSN 02599791) **40**(3), 269–293 (2006)
14. G.-Y. Psihoyios, J.R. Cash, A stability result for general linear methods with characteristic function having real poles only. BIT Numer. Math. (Springer, ISSN 00063835) **38**(3), 612–617 (1998)
15. G. Psihoyios, Towards a general formula for the stability functions of a family of implicit multistep methods, in Lecture Series on Computer and Computational Sciences: ICCMSE-2004 (VSP, ISBN 9067644188) vol. 1, pp. 440–443 (2004)
16. G. Psihoyios, A class of implicit advanced step-point methods with a parallel feature for the solution of stiff initial value problems. Math. Comput. Model. (Elsevier, ISSN 08957177) **40**(11–12), 1199–1224 (2004)
17. G. Psihoyios, A multispep block scheme for the solution of ordinary differential equations, in Lecture Series on Computer and Computational Sciences: IeCCS-2005 (VSP/Brill, ISBN 9067644250) vol. 2, pp. 97–101 (2005)
18. G. Psihoyios, A block implicit advanced step-point (BIAS) algorithm for stiff differential systems. Comput. Lett. (VSP/Brill, E-ISSN 15740404) **2**(1–2), 51–58 (2006)
19. G. Psihoyios, A general formula for the stability functions of a group of implicit advanced step-point (IAS) methods. Math. Comput. Model. (Elsevier, ISSN 08957177) **46**(1–2), 214–224 (2007)
20. L.F. Shampine, M.K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem* (W.H. Freeman, San Francisco, CA, 1975)
21. E. Fehlberg, Classical fifth, sixth, seventh and eighth order Runge–Kutta formulas with step size control. Computing **4**, 93–106 (1969)
22. L.F. Shampine, Some practical Runge–Kutta formulae. Math. Comp. **46**, 135–150 (1986)
23. J.D. Lambert, *Numerical methods for ordinary differential systems* (Wiley, London, 1991)
24. J.R. Cash, S. Semnani, A modified Adams method for non-stiff and mildly stiff IVPs. ACM TOMS **19**, 63–80 (1993)
25. W.H. Enright, J.D. Pryce, Two FORTRAN packages for assessing initial value methods. ACM TOMS **13**, 1–27 (1987)
26. G. Psihoyios, Solving time dependent PDEs via an improved modified extended BDF scheme. Appl. Math. Comput. (Elsevier, ISSN 00963003) **184**(1), 104–115 (2007)