An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 45 (2003) 123–129
www.elsevier.com/locate/camwa

An Algorithm for Starting Multistep Methods

R. TIRANI
Department of Biotechnology and Bioscience
University of Milano – Bicocca
P.zza della Scienza 2, 20126 Milano, Italy

C. PARACELLI
Department of Mathematics, University of Lecce
Via per Arnesano, 73100 Lecce, Italy

Abstract—The present paper deals with the problem of starting multistep methods. We take into consideration an Adams-Bashforth-Moulton PECE pair, with the predictor of order q and the corrector of order $q+1$. To start this method, $q-1$ starting values are necessary, in addition to y_0. A well-known result from theory says that the order of convergence of the whole integration is $q+1$, if all those starting values are accurate of that same order. Present production codes start with a predictor of order 1 and a corrector of order 2 at the first step, and then proceed step by step, each time raising the order by 1, until all the necessary starting values have been obtained. But, in this manner, all the starting errors keep of order 3, and so the whole integration converges no faster than that order. This drawback is normally compensated for, by taking very small step sizes in the starting phase. The general algorithm we propose furnishes, at a reasonably low cost, the necessary number $q-1$ of starting values, each of the appropriate order $q+1$, whatever q might be; it is independent of the particular multistep formula considered, and is mainly designed to be used for high values of q ($q \geq 10$), where the alternative strategies are too expensive or do not exist at all. The numerical results reported show the validity of our approach. © 2003 Elsevier Science Ltd. All rights reserved.

Keywords—Ordinary differential equations, Initial value problems, Multistep methods, Starting values.

1. INTRODUCTION

As it is well known, the mathematical formulation of many problems in science, engineering, and economics, leads to the necessity of solving a system of N first-order ODEs

$$y'(x) = f[x, y(x)], \quad y(x_0) = y_0 \qquad \left(x_0 \leq x \leq x_F, y \in \mathbb{R}^N\right).$$

Among the various numerical methods developed to this end, multistep ones are especially useful and widely used in production codes. We consider an Adam-Bashforth-Moulton PECE pair, with the predictor of order q and the corrector of order $q+1$. (See, for example, the popular code STEP [1], but what we shall say can be trivially adapted to the case when both the predictor and the corrector are of the same order.)

To start this method, $q-1$ starting values y_{n+1} ($n = 0, 1, \dots, q-2$) are necessary in addition to y_0, at the points $x_{n+1} = x_n + h$. Now, a well-known result from theory says that the order of convergence of the whole integration on $[x_0, x_F]$ is $q+1$, if all these starting values are accurate of order $q+1$, that is if $y(x_{n+1}) - y_{n+1} = \mathcal{O}(h^{q+1})$.

This work was supported by Italian MURST.

Present production codes (see again, for example, the already mentioned STEP [1]), start with a predictor of order 1 and a corrector of order 2 at the first step, and then proceed step by step, each time raising the order by 1, until all the necessary starting values y_{n+1} $(n = 0, 1, \dots, q-2)$ have been obtained. But, in this manner, all the starting errors $y(x_{n+1}) - y_{n+1}$ keep of order 3, and so the integration converges no faster than this order all over $[x_0, x_F]$ [2, p. 228]. This drawback is normally compensated for, by taking smaller step sizes in the starting phase [1, p. 52].

Alternatively, a discrete Runge-Kutta method could be used. This procedure is found in the early codes, but nowadays it is usually no longer employed, since it is expensive [2, p. 199].

Another more economical way is to resort to a continuous Runge-Kutta method (CRK method), that is a polynomial $p(x)$, such that $y(x) - p(x) = \mathcal{O}(h^{q+1})$, $(x_0 \le x \le x_{q-1},\ h \to 0)$. The $q-1$ starting values $y_{n+1}(n = 0, 1, \dots, q-2)$ are then given by $y_{n+1} = p(x_{n+1})$. This approach is efficient, but when employing Adams methods, the value of q can be very high. (For example, in the above-mentioned code STEP, it can arrive up to 12 [1, p. 178].) Now, as far as we know, no CRK methods of order > 9 has been published [3].

In this paper, we propose an algorithm, based on a cycle of operations, that, at reasonably low cost, furnishes the necessary number $q-1$ of starting values, each of the appropriate order $q+1$. Such an algorithm is mainly designed to be used when q is very high $(q \ge 10)$, where the alternative methods are too expensive or do not exist at all; its implementation is very simple and its cost (in terms of derivative evaluations) is always known, whichever the value of q (see (6)). (We observe that similar starting schemes can also be found in [4, p. 48], for low values of q.)

2. THE ALGORITHM

Such an algorithm essentially consists in the construction of a sequence

$$S = \{p_j(x),\ j = 1, 2, \dots, q-1\}, \tag{1}$$

of polynomials (whose degree does not exceed the corresponding subscript j), each obtained from the previous one and such that

$$y'(x) - p_j(x) = \mathcal{O}\left(h^{j+1}\right), \tag{2}$$

$(x_0 \le x \le x_{q-1},\ h \to 0)$.

Once the last element $p_{q-1}(x)$ of the above sequence (1) is generated, element for which estimate (2) yields

$$y'(x) - p_{q-1}(x) = \mathcal{O}(h^q), \tag{3}$$

we can consider the polynomial $p_q(x)$, of degree $\le q$,

$$p_q(x) = y_0 + \int_{x_0}^{x} p_{q-1}(t)\, dt, \tag{4}$$

for which it is immediately seen that

$$y(x) - p_q(x) = \mathcal{O}\left(h^{q+1}\right), \tag{5}$$

$(x_0 \le x \le x_{q-1},\ h \to 0)$. (In fact, $y(x) - p_q(x) = \int_{x_0}^{x} [y'(t) - p_{q-1}(t)]\, dt = \mathcal{O}(h^{q+1})$, recalling (3).)

The $q-1$ starting values y_{n+1} $(n = 0, 1, \dots, q-2)$, which we have mentioned in the previous section, are then given by

$$y_{n+1} = p_q(x_{n+1}), \tag{6}$$

and are of order $q+1$, by virtue of (5).

We have therefore to see how to produce the sequence (1) and then that its elements satisfy (2).

The first element $p_1(x)$ is the interpolant at the points (x_0, f_0) $[f_0 = f(x_0, y_0)]$ and $(x_{q-1}, f_{q-1}^{(2)})$, where $f_{q-1}^{(2)} = f(x_{q-1}, y_{q-1}^{(2)})$, with $y_{q-1}^{(2)} = y_0 + (q-1)hf_0$. But, $y(x_{q-1}) - y_{q-1}^{(2)} = \mathcal{O}(h^2)$ as well as $y'(x_{q-1}) - f_{q-1}^{(2)}$, provided that the function f satisfies a Lipschitz condition in y. It is then trivial to see that

$$y'(x) - p_1(x) = \mathcal{O}\left(h^2\right), \tag{7}$$

$(x_0 \le x \le x_{q-1},\ h \to 0)$.

As to the subsequent polynomials, $p_i(x)$ $(i = 2, 3, \dots, q-1)$ is obtained from $p_{i-1}(x)$, by interpolating the $i+1$ points $(x_k, f_k^{(i+1)})$, where

$$f_k^{(i+1)} = f\left(x_k, y_k^{(i+1)}\right) \tag{8}$$

with

$$y_k^{(i+1)} = y_0 + \int_{x_0}^{x_k} p_{i-1}(x)\,dx, \tag{9}$$

$(k = 0, \dots, i/2, q - i/2, \dots, q-1$, when i is even; $k = 0, \dots, (i-1)/2,\ q - (i+1)/2, \dots, q-1$, when i is odd). The values $f_0^{(i+1)}$ $(i = 2, 3, \dots, q-1)$ are to be set equal to f_0 for all i.

Now, move on to the verification of (2). We have already seen (7). If we now show that

$$y'(x) - p_2(x) = \mathcal{O}\left(h^3\right), \tag{10}$$

$(x_0 \le x \le x_{q-1},\ h \to 0)$, the validity of (2) for the subsequent polynomials $p_\nu(x)$ $(\nu = 3, 4, \dots, q-1)$ follows by induction.

The polynomial $p_2(x)$ interpolates the points $(x_k, f_k^{(3)})$ $(k = 0, 1, \dots, q-1;\ f_0^{(3)} = f_0)$, where $f_k^{(3)} = f(x_k, y_k^{(3)})$, with $y_k^{(3)} = y_0 + \int_{x_0}^{x_k} p_1(x)\,dx$ (see (8) and (9)).

It is now trivial to see that $y(x_k) - y_k^{(3)} = \mathcal{O}(h^3)$, and so $f_k^{(3)}$, provided that the function f satisfies a Lipschitz condition in y. (In fact, recalling (7), $y(x_k) - y_k^{(3)} = \int_{x_0}^{x_k} [y'(x) - p_1(x)]\,dx = \mathcal{O}(h^3)$.) The validity of (10) then follows easily.

As to the total cost (in terms of derivative evaluations) necessary to obtain all the $q-1$ starting values y_{n+1} $(n = 0, 1, \dots, q-2)$ given by (6), a straightforward calculation shows that it is

$$1 + \frac{1}{2}(q-1)q. \tag{11}$$

If we now compare such a cost with that of the other two alternative strategies mentioned in Section 1, we see that it is clearly lower than that required by a discrete Runge-Kutta method for $q \ge 3$, while for a continuous one we remember that Owren and Zennaro have determined the minimum number CEN (q) of stages needed by a CRK method of order q, for $q \le 5$ [5]. It results that CEN(1) = 1, CEN(2) = 2, CEN(3) = 4, CEN(4) = 6, and CEN(5) = 8. For $6 \le q \le 9$, we refer to Tables 1a and 1b, respectively, on [3, p. 28,33].

In these cases, we see that our method is slightly more expensive. But, when $q > 9$ (values of q for which our algorithm is mainly designed to be employed, as we have already said), no formula, such as (11), has yet been provided [6] in order to compute the relevant cost. And this makes it impossible a comparison with our method, in this last case.

3. NUMERICAL RESULTS

In this section, we show the performance of our method in the case of the two AB4-AM5 and AB10-AM11 pairs. As we have seen in Section 1, in addition to y_0 we need three starting values accurate to order 5 in the first case, and nine starting values accurate to order 11 in the second. According to (11), the total cost (in terms of derivative evaluations) is 7 and 46, respectively.

The formulae relevant to the AB4-AM5 pair are given in the Appendix to the paper, while those relevat to the AB10-AM11 pair are not reported for space reasons. They can be requested to the authors.

In both cases, we give the results of our algorithm for the problems D1, D3, and D5 proposed by Enright *et al.* in [7]; for the mildly stiff problem

$$\text{(S)} \qquad y' = -100\left(y - \frac{1}{x+1}\right) + \frac{1}{(x+1)^2}, \qquad y(0) = 0,$$

used by Decker *et al.* in [8], and finally for the problem

$$\text{(H)} \qquad \begin{aligned} y_1' &= 2xy_1 \log\left[\max\left(y_2, 10^{-3}\right)\right], & y_1(0) &= 1, \\ y_2' &= -2xy_2 \log\left[\max\left(y_1, 10^{-3}\right)\right], & y_2(0) &= e, \end{aligned}$$

quoted by Hairer *et al.* in [9, p. 174].

In the case of the AB4-AM5 pair, for each of the above problems, we have calculated, at the points $x_{n+1} = x_n + h$ ($n = 0, 1, 2$; $h = 0.1, 0.01, 0.001$), the values $\|e_{n+1}\|_\infty$, with $e_{n+1} = y(x_{n+1}) - y_{n+1}$, where y_{n+1} are the approximations given by formulae (12a)–(12c) in the Appendix.

Such values of $\|e_{n+1}\|_\infty$ are reported in Tables 1–3, respectively, for the values $h = 0.1$, 0.01, and 0.001.

Table 1. ($h = 0.1$).

	x_1	x_2	x_3
S	0.93D+00	0.15D+02	0.77D+02
H	0.60D−03	0.18D−03	0.15D−02
D1	0.30D−05	0.12D−03	0.55D−03
D3	0.15D−02	0.77D−02	0.26D−01
D5	0.26D+01	0.23D+01	0.26D+01

Table 2. ($h = 0.01$).

	x_1	x_2	x_3
S	0.12D−04	0.19D−03	0.97D−03
H	0.47D−09	0.15D−09	0.11D−08
D1	0.51D−10	0.14D−08	0.65D−08
D3	0.65D−08	0.19D−06	0.87D−06
D5	0.94D−02	0.25D−01	0.74D−01

Table 3. ($h = 0.001$).

	x_1	x_2	x_3
S	0.11D−09	0.19D−08	0.99D−08
H	0.00D+00	0.00D+00	0.00D+00
D1	0.00D+00	0.13D−13	0.64D−13
D3	0.64D−13	0.18D−11	0.87D−11
D5	0.10D−06	0.85D−06	0.39D−05

A comparison has also been made with the values given by the fourth-order continuous Runge-Kutta method on [10, p. 205] (Tables 4–6) and with those given by the fourth-order formula of the classical (discrete) RKF4(5) pair (Tables 7–9).

As we have anticipated at the beginning of this section, we have tested our method also for the AB10-AM11 pair. We recall that, in this connection, we need nine starting values y_{n+1}, accurate to order 11, at the points x_{n+1} ($n = 0, 1, \ldots, 8$), where the max-norm $\|e_{n+1}\|_\infty$ of the true errors $y(x_{n+1}) - y_{n+1}$ has been computed for $h = 0.01$ and 0.001 and for each of the above-cited problems (Tables 10 and 11).

Here, we have not compared our algorithm with a tenth-order discrete Runge-Kutta method, because its use would require too high a cost. For example, the 17-stage formula by [9, p. 190], would require 153 derivative evaluations, instead of the 46 ones required by our technique.

Neither we could make a comparison with a tenth-order continuous Runge-Kutta method, since, as we have already said in Section 1, none is nowadays available.

Table 4. ($h = 0.1$).

	x_1	x_2	x_3
S	0.13D+03	0.59D+03	0.17D+03
H	0.12D−02	0.29D−04	0.36D−03
D1	0.25D−04	0.24D−05	0.17D−04
D3	0.23D−02	0.11D−02	0.76D−03
D5	0.30D+01	0.55D+00	0.63D−01

Table 5. ($h = 0.01$).

	x_1	x_2	x_3
S	0.37D−04	0.20D−03	0.23D−03
H	0.88D−09	0.66D−10	0.19D−09
D1	0.25D−09	0.20D−10	0.24D−09
D3	0.66D−07	0.24D−07	0.50D−07
D5	0.11D−01	0.68D−02	0.60D−02

Table 6. ($h = 0.001$).

	x_1	x_2	x_3
S	0.65D−10	0.45D−09	0.76D−09
H	0.00D+00	0.00D+00	0.00D+00
D1	0.00D+00	0.00D+00	0.00D+00
D3	0.68D−12	0.25D−12	0.52D−12
D5	0.45D−06	0.22D−06	0.30D−06

Table 7. ($h = 0.1$).

	x_1	x_2	x_3
S	0.26D+00	0.17D+03	0.12D+06
H	0.28D−06	0.14D−05	0.46D−05
D1	0.27D−07	0.25D−07	0.23D−07
D3	0.85D−05	0.51D−05	0.23D−05
D5	0.92D+00	0.64D−03	0.15D−04

Table 8. ($h = 0.01$).

	x_1	x_2	x_3
S	0.48D−06	0.64D−06	0.69D−06
H	0.26D−12	0.95D−12	0.21D−11
D1	0.28D−12	0.28D−12	0.28D−12
D3	0.39D−10	0.39D−10	0.38D−10
D5	0.61D−04	0.27D−04	0.63D−05

Table 9. ($h = 0.001$).

	x_1	x_2	x_3
S	0.26D−11	0.50D−11	0.72D−11
H	0.00D+00	0.00D+00	0.00D+00
D1	0.00D+00	0.00D+00	0.00D+00
D3	0.00D+00	0.00D+00	0.00D+00
D5	0.61D−04	0.27D−04	0.63D−05

Table 10. ($h = 0.01$).

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
S	0.21D−09	0.22D−06	0.13D−04	0.23D−03	0.22D−02	0.13D−01	0.63D−01	0.24D+00	0.78D+00
H	0.31D−02	0.27D−02	0.33D−02	0.55D−02	0.15D−01	0.39D−01	0.88D−01	0.17D+00	0.30D+00
D1	0.11D−01	0.22D−01	0.33D−01	0.44D−01	0.55D−01	0.67D−01	0.78D−01	0.89D−01	0.99D−01
D3	0.36D−01	0.72D−01	0.10D−01	0.14D+00	0.18D+00	0.21D+00	0.24D+00	0.28D+00	0.31D+00
D5	0.84D+00	0.14D+01	0.18D+01	0.18D+01	0.19D+01	0.22D+01	0.25D+01	0.26D+01	0.27D+01

Table 11. ($h = 0.001$).

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
S	0.32D−14	0.24D−14	0.17D−14	0.43D−14	0.34D−13	0.18D−12	0.78D−12	0.24D−11	0.90D−10
H	0.00D+00	0.00D+00	0.00D+00	0.00D+00	0.00D+00	0.00D+00	0.00D+00	0.00D+00	0.17D−14
D1	0.41D−14	0.32D−14	0.34D−14	0.27D−14	0.38D−14	0.10D−14	0.23D−13	0.47D−13	0.16D−12
D3	0.31D−13	0.10D−13	0.12D−13	0.13D−13	0.28D−13	0.67D−13	0.12D−12	0.23D−12	0.41D−12
D5	0.81D−11	0.72D−11	0.83D−11	0.90D−11	0.10D−10	0.12D−10	0.18D−10	0.60D−10	0.16D−09

4. CONCLUSIONS

A general algorithm, mainly designed for high values of q ($q \geq 10$), has been presented, which furnishes, in addition to y_0, at the points $x_{n+1} = x_n + h$, ($n = 0, 1, \ldots, q-2$), the $q-1$ values y_{n+1} necessary to start a PECE method, with the predictor of order q and the corrector of order

$q+1$. Such values y_{n+1} are accurate to order $q+1$, which is appropriate to assure that the PECE scheme be uniformly convergent of that same order on the whole of the interval of integration.

The bulk of such an algorithm consists in the construction of a polynomial $p_{q-1}(x)$, such that $y'(x) - p_{q-1}(x) = \mathcal{O}(h^q)$, $(x_0 \le x \le x_{q-1},\ h \to 0)$. The values y_{n+1} are then given by integration (see (4) and (6)). The relevant overall cost (in terms of derivative evaluations) is given by (11).

APPENDIX

Here, we give the formulae relevant to our technique in the case of an AB4-AM5 pair. As we have already said in Section 3, we need three starting values y_{n+1} $(n = 0, 1, 2)$, in addition to y_0, all of which accurate to order 5.

To this purpose, having constructed the polynomial $p_3(x)$ such that $y'(x) - p_3(x) = \mathcal{O}(h^4)$ (see (7)), relations (4)–(6) give, with $q = 4$, $y_{n+1} = p_4(x_{n+1}) = y_0 + \int_{x_0}^{x_{n+1}} p_3(x)\,dx$. Applying the algorithm of Section 2 and employing the normalized variable $\theta \in [0,1]$, defined by $x = x_0 + 3h\theta$, we successively have

$$y_3^{(2)} = y_0 + 3hf_0,$$

$$f_3^{(2)} = f\left(x_3, y_3^{(2)}\right),$$

$$p_1(\theta) = (1-\theta)f_0 + \theta f_3^{(2)},$$

$$y_1^{(3)} = y_0 + 3h\int_0^{1/3} p_1(\theta)\,d\theta = y_0 + \frac{h}{6}\left(5f_0 + f_3^{(2)}\right),$$

$$y_3^{(3)} = y_0 + 3h\int_0^{1} p_1(\theta)\,d\theta = y_0 + \frac{3h}{2}\left(f_0 + f_3^{(2)}\right),$$

$$f_1^{(3)} = f\left(x_1, y_1^{(3)}\right), \qquad f_3^{(3)} = f\left(x_3, y_3^{(3)}\right),$$

$$p_2(\theta) = \left(3\theta^2 - 4\theta + 1\right) f_0 - \frac{9}{2}\left(\theta^2 - \theta\right) f_1^{(3)} + \frac{1}{2}\left(3\theta^3 - \theta\right) f_3^{(3)},$$

$$y_1^{(4)} = y_0 + 3h\int_0^{1/3} p_2(\theta)\,d\theta = y_0 + \frac{h}{3}\left(\frac{4}{3}f_0 + \frac{7}{4}f_1^{(3)} - \frac{1}{12}f_3^{(3)}\right),$$

$$y_2^{(4)} = y_0 + 3h\int_0^{2/3} p_2(\theta)\,d\theta = y_0 + \frac{h}{3}\left(\frac{2}{3}f_0 + 5f_1^{(3)} + \frac{1}{3}f_3^{(3)}\right),$$

$$y_3^{(4)} = y_0 + 3h\int_0^{1} p_2(\theta)\,d\theta = y_0 + \frac{3h}{4}\left(3f_1^{(3)} + f_3^{(3)}\right),$$

$$f_1^{(4)} = f\left(x_1, y_1^{(4)}\right), \qquad f_2^{(4)} = f\left(x_2, y_2^{(4)}\right), \qquad f_3^{(4)} = f\left(x_3, y_3^{(4)}\right),$$

$$p_3(\theta) = -\frac{1}{2}\left(9\theta^3 - 18\theta^2 + 11\theta - 2\right) f_0 + \frac{9}{2}\left(3\theta^3 - 5\theta^2 + 2\theta\right) f_1^{(4)}$$
$$- \frac{9}{2}\left(3\theta^3 - 4\theta^2 + \theta\right) f_2^{(4)} + \frac{1}{2}\left(9\theta^3 - 9\theta^2 + 2\theta\right) f_3^{(4)}.$$

The three starting values y_{n+1} $(n = 0, 1, 2)$ accurate to order 5, are then given by

$$y_1 = y_0 + 3h\int_0^{1/3} p_3(\theta)\,d\theta = y_0 + \frac{h}{24}\left(9f_0 + 19f_1^{(4)} - 5f_2^{(4)} + f_3^{(4)}\right), \tag{12a}$$

$$y_2 = y_0 + 3h\int_0^{2/3} p_3(\theta)\,d\theta = y_0 + \frac{h}{3}\left(f_0 + 4f_1^{(4)} + 5f_2^{(4)}\right), \tag{12b}$$

$$y_3 = y_0 + 3h\int_0^{1} p_3(\theta)\,d\theta = y_0 + \frac{3h}{8}\left(f_0 + 3f_1^{(4)} + 3f_2^{(4)} + f_3^{(4)}\right), \tag{12c}$$

REFERENCES

1. L.F. Shampine and M.K. Gordon, *Computer Solution of Ordinary Differential Equations*, Freeman and Co., San Francisco, CA, (1975).
2. L.F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, New York, (1994).
3. J.H. Verner, *Differentiable Interpolants for High-Order Runge-Kutta Methods*, Mathematical Preprint #1990-9, Queen's University at Kingston, Kinsgton, Canada, (1990).
4. W.E. Milne, *Numerical Solution of Differential Equations*, J. Wiley and Sons, New York, (1953).
5. B. Owren and M. Zennaro, Order barriers for continuous explicit Runge-Kutta methods, *Math. Comp.* **56**, 645–661 (1991).
6. J.H. Verner, Private communication.
7. W.H. Enright and J.D. Pryce, Two FORTRAN packages for assessing initial value methods, *ACM Trans. Math. Software* **13**, 1–27 (1987).
8. K. Dekker and J.G. Verwer, Estimating the global error of Runge-Kutta approximations for ordinary differential equations, *Differential-Difference Equations,* (Edited by L. Collatz *et al.*), ISNM Series, Basel **62**. 55–71 (1983).
9. E. Hairer, S.P. Nørsett and J. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin, (1987).
10. W.H. Enright, K.R. Jackson, S.P. Nørsett and P.G. Thomsen, Interpolants for Runge-Kutta formulae, *ACM Trans. Math. Software* **12**, 193–218 (1986).
11. L.F. Shampine and S. Baca, Global error estimation for ODE's based on extrapolation methods, *SIAM J. Sci. Stat. Comput.* **6**, 1–14 (1985).