

# Function Fitting Two-Step BDF Algorithms for ODEs

Liviu G. Ixaru<sup>1</sup> and Beatrice Paternoster<sup>2</sup>

<sup>1</sup> Institute of Physics and Nuclear Engineering, Bucharest, Romania

<sup>2</sup> Dipartimento di Matematica e Informatica, Università di Salerno, Italy

**Abstract.** We investigate the problem of how big would be the additional accuracy gain from a two-step bdf algorithm for ordinary differential equations if its weights are constructed via function fitting. We find that (i) the order of the algorithm is increased by three units (from two to five), (ii) this enhancement can be achieved not only in the frame of the traditional exponential fitting but also in the frame of a new, more general approach, which results more flexible and safer than the other one.

## 1 Introduction

It is well known that the quality of a multistep algorithm to solve first order ODEs depends on the set of reference functions chosen for the determination of the algorithm weights. The classical form of these algorithms is constructed in terms of power functions but other functions may be used as well, such as exponential functions or mixtures of power and exponential functions, see e.g. [3], [1], [8], [11], and the references therein.

The old problem of how the frequencies of the exponential functions should be tuned in order to obtain a maximal gain in accuracy received a pertinent answer only recently, [6]. (According to the usual terminology in the field, the frequency is the parameter  $\lambda$  in the function  $\exp(\lambda x)$ .) In this paper our main interest consists in searching for alternatives to the exponential fitting. Multistep algorithms based on other functions than the power and/or exponential functions were published before, mainly in connection with the Schrödinger equation (see [2] for functions of the form  $x^{l+k}$ ,  $l > 0$ ,  $k = 1, 2, \dots$  (i.e. an ad hoc subset of power functions) or [7,9,10] for the Bessel and Neumann functions). Now we choose these functions in terms of the behaviour of the derivatives of the solution, implementing the function fitting technique to derive a fifth order version of the method.

## 2 Preliminaries

Let  $X$  be some point on the real axis and  $[X - h, X + h]$  an interval around it,  $[X - h, X + h]$ . We consider the third order linear and homogeneous differential equation, where  $\alpha_m(x)$ ,  $m = 0, 1, 2$ , are low degree polynomials:

$$y^{(3)} + \alpha_2(x)y'' + \alpha_1(x)y' + \alpha_0(x)y = 0, \quad X - h \leq x \leq X + h, \quad (1)$$

We denote the three linear independent solutions of equation (1) as  $\phi_1(x), \phi_2(x)$  and  $\phi_3(x)$ . Each of them is constructed by solving the initial value problem for a specified set of initial conditions at  $x = X$ . We take:

$$\begin{aligned} y(X) &= 1, \quad y'(X) = 0, \quad y''(X) = 0 \quad \text{for } \phi_1(x), \\ y(X) &= 0, \quad y'(X) = 1, \quad y''(X) = 0 \quad \text{for } \phi_2(x), \\ y(X) &= 0, \quad y'(X) = 0, \quad y''(X) = 2 \quad \text{for } \phi_3(x). \end{aligned}$$

The solution of eq.(1) can be constructed through a power series expansion around  $X$ :

$$y(x) = \sum_{n=0} y_n \Delta^n, \quad \text{with } \Delta = x - X, \quad \Delta \in [-h, h]. \quad (2)$$

We are interested only in five particular cases; each of these cases leads to a specific set of weights in the two-step bdf algorithm. The case abbreviated below as C0 leads to the classical algorithm, C1, C2 and C3 provide three versions which form together the new, flexible fitting algorithm while C4 leads to a typical exponential fitting algorithm.

**C0** . This is the particular case of eq.(1) when  $\alpha_0(x) = \alpha_1(x) = \alpha_2(x) = 0$ . In this case the three solutions are simply  $\phi_1(x) = 1$ ,  $\phi_2(x) = x - X$  and  $\phi_3(x) = (x - X)^2$ .

**C1** . This is  $\alpha_2(x) = p_0 + p_1\Delta + p_2\Delta^2$ ,  $\alpha_1(x) = 0$ ,  $\alpha_0(x) = 0$ , where  $p_0$ ,  $p_1$  and  $p_2$  are real constants. Eq.(1) reads

$$y^{(3)} + (p_0 + p_1\Delta + p_2\Delta^2)y'' = 0. \quad (3)$$

Upon inserting (2) in (3) and conveniently organizing the terms, the following recurrence relation is obtained:

$$\begin{aligned} y_{n+3} &= -\frac{1}{(n+3)(n+2)(n+1)} [(n+2)(n+1)p_0 y_{n+2} \\ &\quad + (n+1)n p_1 y_{n+1} + n(n-1)p_2 y_n], \quad n = 0, 1, 2, \dots \end{aligned}$$

Since in this case we have  $\phi_1(x) = 1$  and  $\phi_2(x) = x - X$  directly, the recurrence relation has to be used only for the construction of  $\phi_3(x)$ . The starting values consistent with the mentioned initial value set for  $\phi_3(x)$  are:

$$y_0 = 0, \quad y_1 = 0, \quad y_2 = 1.$$

**C2** . This corresponds to  $\alpha_2(x) = 0$ ,  $\alpha_1(x) = p_0 + p_1\Delta + p_2\Delta^2$ ,  $\alpha_0(x) = 0$ , where again  $p_0$ ,  $p_1$  and  $p_2$  are real constants. Eq.(1) now becomes

$$y^{(3)} + (p_0 + p_1\Delta + p_2\Delta^2)y' = 0.$$

We have  $\phi_1(x) = 1$  directly, while for  $\phi_2(x)$  and  $\phi_3(x)$  we use the recurrence formula

$$y_{n+3} = -\frac{1}{(n+3)(n+2)(n+1)} [(n+1)p_0 y_{n+1} + n p_1 y_n + (n-1)p_2 y_{n-1}], \quad n = 0, 1, \dots$$

with the starting values

$$\begin{aligned} y_0 &= 0, \quad y_1 = 1, \quad y_2 = 0 \quad \text{for } \phi_2(x), \\ y_0 &= 0, \quad y_1 = 0, \quad y_2 = 1 \quad \text{for } \phi_3(x), \end{aligned}$$

and  $y_{-1} = 0$  by default.

**C3**. This corresponds to  $\alpha_2(x) = 0$ ,  $\alpha_1(x) = 0$ ,  $\alpha_0(x) = p_0 + p_1\Delta + p_2\Delta^2$ , where, as before,  $p_0$ ,  $p_1$  and  $p_2$  are real constants. Eq.(1) reads

$$y^{(3)} + (p_0 + p_1\Delta + p_2\Delta^2)y = 0.$$

with the following recurrence relation for  $\phi_1(x)$ ,  $\phi_2(x)$  and  $\phi_3(x)$ :

$$y_{n+3} = -\frac{1}{(n+3)(n+2)(n+1)}[p_0y_n + np_1y_{n-1} + n(n-1)p_2y_{n-2}], \quad n = 0, 1, 2, \dots$$

with the starting values

$$\begin{aligned} y_0 &= 1, \quad y_1 = 0, \quad y_2 = 0, \quad \text{for } \phi_1(x), \\ y_0 &= 0, \quad y_1 = 1, \quad y_2 = 0, \quad \text{for } \phi_2(x), \\ y_0 &= 0, \quad y_1 = 0, \quad y_2 = 1, \quad \text{for } \phi_3(x), \end{aligned} \tag{4}$$

and with the default values  $y_{-2} = y_{-1} = 0$  in all these.

**C4**. Here  $\alpha_0(x) = q_0$ ,  $\alpha_1(x) = q_1$ ,  $\alpha_2(x) = q_2$ , where  $q_0$ ,  $q_1$  and  $q_2$  are real constants. The solution of

$$y^{(3)} + q_2y'' + q_1y' + q_0y = 0 \tag{5}$$

is given by the recurrence relation

$$y_{n+3} = -\frac{1}{(n+3)(n+2)(n+1)}[(n+2)(n+1)q_2y_{n+2} + (n+1)q_1y_{n+1} + q_0y_n],$$

$n = 0, 1, 2, \dots$  with the starting values (4).

This case admits also an analytic solution. Three linear independent solutions are  $\exp(\lambda_1x)$ ,  $\exp(\lambda_2x)$  and  $\exp(\lambda_3x)$ , where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are the roots of the polynomial  $P(\lambda) = \lambda^3 + q_2\lambda^2 + q_1\lambda + q_0$ . Our  $\phi_1(x)$ ,  $\phi_2(x)$  and  $\phi_3(x)$  are some linear combinations of these exponential functions. For all cases C1, C2, C3 and C4 the number of terms to be retained in the series (2) in order to reach some predetermined accuracy in the results depends on  $h$  and on the numerical values of the parameters  $p_0$ ,  $p_1$ ,  $p_2$  or  $q_0$ ,  $q_1$ ,  $q_2$ . For given  $p$ -s or  $q$ -s this number decreases with  $h$ .

#### Weights of the two-step bdf algorithm

We consider the initial value problem

$$y' = f(x, y), \quad x \in [a, b], \quad y(a) = y_0, \tag{6}$$

and its solution by a two-step bdf algorithm,

$$a_0\bar{y}_k + a_1\bar{y}_{k+1} + \bar{y}_{k+2} = hb_2f(x_{k+2}, \bar{y}_{k+2}), \quad k = 0, 1, 2, \dots, \tag{7}$$

on an equidistant partition with the stepsize  $h$ .  $\bar{y}_j$  is an approximation to  $y(a + jh)$ . The weights  $a_0$ ,  $a_1$  and  $b_2$  will differ from one interval to another. Their

construction on the current interval centered at  $X = x_{k+1}$  (and then  $x_k = X - h$ ,  $x_{k+2} = X + h$ ) is done as it follows.

The linear functional

$$\mathcal{L}[y(x); X, h, a_0, a_1, b_2] := a_0 y(X - h) + a_1 y(X) + y(X + h) - h b_2 y'(X + h)$$

is associated to algorithm (7) and we require that this functional is identically vanishing when  $y(x) = \phi_1(x), \phi_2(x), \phi_3(x)$ . We have

$$a_0 \phi_i(X - h) + a_1 \delta_{i1} + \phi_i(X + h) - b_2 h \phi'_i(X + h) = 0, \quad i = 1, 2, 3. \quad (8)$$

Here  $\delta_{ij}$  is the usual Kronecker symbol and it appears because, by the very construction, the three functions satisfy  $\phi_1(X) = 1$  and  $\phi_2(X) = \phi_3(X) = 0$ . The linear system (8) has the solution

$$\begin{aligned} D &= h[-\phi_2(X - h)\phi'_3(X + h) + \phi_3(X - h)\phi'_2(X + h)], \\ a_0 &= h[\phi_2(X + h)\phi'_3(X + h) - \phi_3(X + h)\phi'_2(X + h)]/D, \\ b_2 &= [-\phi_2(X - h)\phi_3(X + h) + \phi_3(X - h)\phi_2(X + h)]/D, \\ a_1 &= b_2 h \phi'_1(X + h) - a_0 \phi_1(X - h) - \phi_1(X + h). \end{aligned}$$

When  $\phi_1(x) = 1$ ,  $\phi_2(x) = x - X$  and  $\phi_3(x) = (x - X)^2$ , as for C0, we get

$$a_0 = \frac{1}{3}, \quad a_1 = -\frac{4}{3} \quad \text{and} \quad b_2 = \frac{2}{3}, \quad (9)$$

that is the classical weights.

### 3 Error Analysis

For the error analysis it is convenient to express  $\mathcal{L}$  in terms of the moments. The moment  $L_m$  is the expression of  $\mathcal{L}$  when  $y(x)$  is the power function  $(x - X)^m$ , i.e.

$$L_m =: \mathcal{L}[(x - X)^m; X, h, a_0, a_1, b_2].$$

In fact, the knowledge of the moments allow writing

$$\mathcal{L}[y(x); X, h, a_0, a_1, b_2] = \sum_{m=0}^{\infty} \frac{1}{m!} L_m y^{(m)}(X) \quad (10)$$

for any function  $y(x)$  which admits a series expansion in power functions.

If  $y(x)$  is the solution of (6), the series (10) furnishes the local error of the method. Omitting the details, it is possible to draw two conclusions:

1. If  $p_0 = p_1 = p_2$ , as for the classical algorithm, the first nonvanishing moment is  $L_3$ . The leading term of the error then reads

$$lte = \frac{1}{3!} L_3 y^{(3)}(X) = -\frac{2}{9} h^3 y^{(3)}(X),$$

a well known result. The order of this version is therefore two.

2. The order is still two also when  $p_0$ ,  $p_1$  and/or  $p_2$  are different from zero. This is because the free term in  $L_2$  is absent i.e.  $L_2 \sim h^3$ .

$p_0$ ,  $p_1$  and  $p_2$  should be chosen in order to obtain an increased accuracy. To this purpose we introduce the function

$$s(x) =: \frac{y^3(x)}{y''(x)}, \quad (11)$$

where  $y(x)$  is the exact solution of eq.(6). We have

$$\begin{aligned} y^{(3)}(x) &= -s(x)y''(x), \quad y^{(4)}(x) = (s^2(x) - s'(x))y''(x), \\ y^{(5)}(x) &= (s^3(x) - 3s(x)s'(x) + s''(x))y''(x), \dots \end{aligned}$$

An analysis of  $\mathcal{L}$  suggests searching for a determination of the  $p$ -s in terms of  $s(x)$ . We just require that the parabola  $\alpha_2(x) = p_0 + p_1\Delta + p_2\Delta^2$  is interpolating  $s(x)$  at the three knots  $X$  and  $X \pm h$  to obtain

$$p_0 = s_0, \quad p_1 = \frac{1}{2h}(s_1 - s_{-1}), \quad p_2 = \frac{1}{2h^2}(s_1 - 2s_0 + s_{-1})$$

where  $s_j = s(X + jh)$ . With this determination, finally,  $\mathcal{L} \sim h^6$ .

To summarize, we have obtained the following result: *The C1 version of the two step bdf algorithm (7) is in general of the second order. However, if the coefficients  $p_0$ ,  $p_1$  and  $p_2$  are fixed by the parabolic interpolation of function  $s(x)$  defined by eq.(11), the order becomes five.*

The above error analysis can be repeated for versions C2 and C3, as well. The result remains the same, i.e. each of these is in general of the second order, but if the  $p$ -s are determined by interpolating the functions

$$s(x) =: \frac{y^3(x)}{y'(x)},$$

for C2, and

$$s(x) =: \frac{y^3(x)}{y(x)},$$

for C3, then these versions become of the fifth order.

Seen from practical point of view each of these three versions exhibits its own limitation. For instance, C1 version has to be avoided when the second derivative of the solution of (6) has a zero inside the quoted interval. Likewise, the use of optimal C2 (or C3) versions should be avoided on the intervals where the first derivative of the solution (or the solution itself) has a zero. For the (exponential fitting) C4-based algorithm, the interpolation procedure to obtain a fifth order algorithmis is necessarily replaced by that of solving the linear algebraic system

$$q_2y''(X + jh) + q_1y'(X + jh) + q_0y(X + jh) = -y^{(3)}(X + jh), \quad j = -1, 0, 1 \quad (12)$$

for the unknowns  $q_0$ ,  $q_1$  and  $q_2$ , which becomes increasingly ill-conditioned and therefore the determination of the parameters is less and less accurate.

## 4 The Flexible Function Fitting Algorithm

In spite of the fact that each of these version has its own limitation, the very existence of three function fitting versions of the same fifth order makes a choice possible in terms of a safety criterion. The rule of choosing between various versions in the flexible fitting algorithm is in the following.

Let us assume that we know the exact values at the points  $X - h$ ,  $X$  and  $X + h$  for the solution of the differential equation (6) and for its derivatives.

We introduce

$$\begin{aligned} s^1 &= -\frac{\sum_{j=-1}^1 y^{(3)}(X + jh)}{\sum_{j=-1}^1 y''(X + jh)}, \quad s^2 = -\frac{\sum_{j=-1}^1 y^{(3)}(X + jh)}{\sum_{j=-1}^1 y'(X + jh)}, \\ s^3 &= -\frac{\sum_{j=-1}^1 y^{(3)}(X + jh)}{\sum_{j=-1}^1 y(X + jh)}, \end{aligned}$$

and evaluate the maximal deviation of the input data for each version,

$$\Delta s^m = \max[|s^m - s_{-1}^m|, |s^m - s_0^m|, |s^m - s_1^m|], \quad m = 1, 2, 3.$$

We compare  $\Delta s^1$ ,  $\Delta s^2$  and  $\Delta s^3$  and let  $\bar{m}$  be that  $m$  such that  $\Delta s^{\bar{m}}$  is the smallest of the three. The version  $C\bar{m}$  is selected for application in that step. In the case of systems of differential equations the selection of the optimal version is operated on each component separately because only in this way the method remains of the fifth order.

In a real run the exact values of the solution and of its derivatives are obviously not known but we just rely on the numerical values calculated up to that interval, that is on  $\bar{y}_0$ ,  $\bar{y}_1$ ,  $\bar{y}_2, \dots$ ,  $\bar{y}_k$ ,  $\bar{y}_{k+1}$ . On effectively using the last three of them, an extrapolated value at the new point  $x_{k+2} = X + h$ , denoted  $y_{k+2}^*$ , is generated by the four point Milne–Simpson formula

$$-\bar{y}_{k-1} + \frac{27}{11}(-\bar{y}_k + \bar{y}_{k+1}) + y_{k+2}^* = h[\frac{3}{11}(f_{k-1} + f(x_{k+2}, y_{k+2}^*)) + \frac{27}{11}(f_k + f_{k+1})],$$

which is sufficiently accurate for this purpose; its error behaves as  $h^7$ .  $\bar{y}_k$ ,  $\bar{y}_{k+1}$  and  $y_{k+2}^*$  are then accepted as sufficiently reliable representations of the exact values of the solution at the three mesh points of the current interval. The values of the derivatives at the same points (these are needed for the interpolation) are generated via the analytic expressions of  $f$  and of its total first and second derivative with respect to  $x$ .

## 5 A Numerical Illustration

We present one test case, whose exact solution is known. We compare three algorithms: (i) The classical two-step bdf algorithm (C0); it is based on the three solutions of eq.(1) with  $\alpha_0(x) = \alpha_1(x) = \alpha_2(x) = 0$  and it has the weights in (9). (ii) The optimal exponential fitting algorithm (CEF); this is based on the

solutions of eq.(5) where the three parameters  $q_0$ ,  $q_1$  and  $q_2$  are calculated by solving the system (12) in each step. (iii) The optimal flexible fitting algorithm (CFF); in each step this chooses between versions C1, C2 and C3 in terms of the safety criterion explained before. We assume that  $y(a)$  and  $y(a + h)$  are given for C0 while three starting values,  $y(a)$ ,  $y(a + h)$  and  $y(a + 2h)$  are available for CEF and CFF. The third data is needed for the activation of the Milne–Simpson extrapolation.

Let us consider the system of two equations

$$\begin{aligned} y'_1 &= -2y_1 + y_2 + 2 \sin(x), \\ y'_2 &= y_1 - 2[y_2 + \sin(x) - \cos(x)], \quad x \in [0, 2], \quad y_1(0) = 2, \quad y_2(0) = 1. \end{aligned} \quad (13)$$

Its solution is

$$\begin{aligned} y_1(x) &= \exp(-x) + \exp(-3x) + \sin(x), \\ y_2(x) &= \exp(-x) - \exp(-3x) + \cos(x). \end{aligned}$$

**Table 1.** Absolute errors  $\Delta y_i(x) = y_i(x) - y_i^{comput}(x)$  at  $x = 1.5$  and  $2.0$  from the classical two-point bdf algorithm C0, its optimal exponential fitting extension and its optimal flexible fitting extension for the two components of system (13).

$h$	$x = 1.5$		$x = 2.0$	
	$\Delta y_1$	$\Delta y_2$	$\Delta y_1$	$\Delta y_2$
C0				
0.0500	0.822(-03)	-0.353(-03)	0.260(-03)	-0.230(-03)
0.0250	0.198(-03)	-0.853(-04)	0.604(-04)	-0.579(-04)
0.0125	0.485(-04)	-0.210(-04)	0.145(-04)	-0.145(-04)
CEF				
0.0500	0.131(-07)	0.151(-07)	0.115(-07)	0.143(-07)
0.0250	0.409(-09)	0.482(-09)	0.329(-09)	0.326(-09)
0.0125	0.128(-10)	0.152(-10)	0.163(-06)	0.522(-06)
CFF				
0.0500	-0.121(-07)	-0.154(-07)	-0.400(-08)	-0.113(-07)
0.0250	-0.832(-09)	-0.884(-09)	-0.393(-09)	-0.597(-09)
0.0125	-0.286(-10)	-0.301(-10)	-0.139(-10)	-0.201(-10)

Each of the two components is a linear combination of four exponential functions, with the frequencies  $-1$ ,  $-3$  and  $\pm i$ , respectively. For this reason, one may be tempted to admit that the exponential fitting version is the method of choice. The reality is however different. At each step, and for each of the two components of the equation, the version CEF determines its optimal parameters by first solving the linear system (12) but the accuracy of this evaluation depends on whether the system is well or badly conditioned which, at its turn, depends of the magnitude of  $h$ . When  $h$  is still big the effect is negligible but when  $h$  is

further decreased it becomes more and more important. In the case of system (13) this effect is negligible for both components of the solution when  $h = 0.05$  or  $h = 0.025$  but when  $h = 0.0125$  it becomes important around  $x = 1.7$  for the second component. Note also that  $y_2(x)$  changes the sign in that region.

The consequences are clearly seen in table 2. The errors from CEF are conforming the theoretical fifth order when  $x = 1.5$  but when  $x = 2$  the errors at  $h = 0.0125$  are abnormally big and they will remain so at any bigger  $x$ . This does not happen with CFF because this algorithm is flexible, and also because the parameters are calculated by interpolation. In the interval around  $x = 1.7$  the version C3 is excluded for the second component but the two others are still available. As a matter of fact, in that region, it activates the version C3 for the first component but C2 for the second.

The result is very encouraging but more investigations are needed for a complete understanding of the properties of the new versions and for improving the quality of the present code, together with analysis of the stability properties of the CFF algorithm.

## References

1. L. Gr. Ixaru, Numerical Methods for Differential Equations and Applications, Reidel, Dordrecht-Boston-Lancaster, 1984.
2. L. Gr. Ixaru, The Numerov method and singular potentials, *J. Comput. Phys.* **72**, 270–274, 1987.
3. L. Gr. Ixaru, Operations on oscillatory functions, *Comput. Phys. Commun.* **105**, 1–19, 1997.
4. L. Gr. Ixaru and B. Paternoster, A Gauss quadrature rule for oscillatory integrands, *Comput. Phys. Commun.* **133**, 177–188, 2001.
5. L. Gr. Ixaru, M. Rizea, H. De Meyer and G. Vanden Berghe, Weights of the exponential fitting multistep algorithms for first order ODEs, *J. Comput. Appl. Math.* **132**, 83–93, 2001.
6. L. Gr. Ixaru, G. Vanden Berghe and H. De Meyer, Frequency evaluation in exponential fitting multistep algorithms for ODEs, *J. Comput. Appl. Math.* **140**, 423–434, 2002.
7. T. E. Simos, A sixth order Bessel and Neumann fitted method for the numerical solution of the Schrödinger equation, *Molecular Simulation* **21**, 191–204, 1999.
8. T. E. Simos, An exponentially fitted eight-order method for the numerical solution of the Schrödinger equation, *J. Comput. Appl. Math.* **108**, 177–194, 1999.
9. T. E. Simos and A. D. Raptis, A fourth order Bessel fitting method for the numerical solution of the Schrödinger equation, *J. Comput. Appl. Math.* **43**, 313–322, 1992.
10. T. E. Simos and P. S. Williams, Bessel and Neumann fitted methods for the numerical solution of the radial Schrödinger equation, *Computers and Chemistry* **21**, 175–179, 1997.
11. G. Vanden Berghe, H. De Meyer, M. Van Daele and T. Van Hecke, Exponentially-fitted explicit Runge-Kutta methods, *Comput. Phys. Commun.* **123**, 7–15, 1997.