Journal of Computational Mathematics
Vol.29, No.2, 2011, 199–214.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1009-m3137

A PREDICTOR MODIFICATION TO THE EBDF METHOD FOR STIFF SYSTEMS*

Elisabete Alberdi
Lea Artibai Ikastetxea, Xemein Avenue 19, 48270 Markina-Xemein (Vizcaya), Spain
Email: ealberdi@leartik.com
Juan José Anza
Department of Applied Mathematics, ETS de Ingeniería de Bilbao, Universidad del País Vasco,
Alameda de Urquijo s/n, 48013 Bilbao, Spain
Email: juanjose.anza@ehu.es

Abstract

In this paper we modify the EBDF method using the NDFs as predictors instead of BDFs. This modification, that we call ENDF, implies the local truncation error being smaller than in the EBDF method without losing too much stability. We will also introduce two more changes, called ENBDF and EBNDF methods. In the first one, the NDF method is used as the first predictor and the BDF as the second predictor. In the EBNDF, the BDF is the first predictor and the NDF is the second one. In both modifications the local truncation error is smaller than in the EBDF. Moreover, the EBNDF method has a larger stability region than the EBDF.

Mathematics subject classification: 65L05.
Key words: Backward differentiation formula (BDF), EBDF, Predictor, Stability, Stiff Systems.

1. Introduction

We will consider the following initial value problem (IVP):

$$y'(x) = f\left(x, y(x)\right), \quad y\left(x_0\right) = y_0 \tag{1.1}$$

where $T = [x_0, x_n]$ is a finite interval and y: $[x_0, x_n] \to \mathbb{R}^m$ and f: $[x_0, x_n] \times \mathbb{R}^m \to \mathbb{R}^m$ are continuous functions.

When we are working with a stiff problem, the numerical method used must be accurate and it needs an extensive stability region too [4]. Because of the latter reason, in the recent years many researches have been focused on developing convenient numerical methods for stiff problems and a lot of improvements have been made on the basis of the backward differentiation formula (BDF) introduced by Gear [6], due to its good stability properties.

One of the modifications done to the BDFs are the NDFs (Numerical Differentiation formulae). It is a computationally cheap modification that consists of anticipating a difference of order $(k+1)$ multiplied by a constant $\kappa\gamma_k$ in the BDF formula of order k. This term has a positive effect on the local truncation error, making the NDFs more accurate than the BDFs and not much less stable. This modification was proposed by Shampine [10] but only for orders $k = 1, 2, 3, 4$, because it is inefficient for orders greater than 4.

* Received June 11, 2009 / Revised version received July 18, 2010 / Accepted July 20, 2010 /
Published online November 20, 2010 /

In [1] and [2] Cash introduces methods using superfuture points to solve stiff IVPs. These methods are known as extended BDF (EBDF) and modified extended BDF (MEBDF). They consist of applying the BDF predictors twice and one implicit multistep corrector. Both methods use superfuture points to gain stability and they are A-stable up to order 4 and $A(\alpha)$-stable up to order 9. In [3] a code based on the MEBDF is described and in [8] Matrix free MEBDF (MF-MEBDF) methods are introduced to optimize the computations of the EBDF. A different variation of the BDFs was introduced by Fredebeul [5], the A-BDF method. In this method the implicit and explicit BDF are used in the same formula, with a free parameter, being $A(\alpha)$-stable up to order 7.

In this paper, we follow the EBDF scheme but substituting the BDF predictors by the NDF formulae. In the ENDF method we will use the NDFs as predictors maintaining the last corrector of the EBDF. The result of this application will be a smaller local truncation error and a not too much smaller stability region than in the EBDF. Next, we introduce two modifications more, the EBNDF and ENBDF maintaining the corrector of the EBDF scheme. In EBNDF the first predictor is the BDF and the second one the NDF. In ENBDF, the first predictor is the NDF and the second BDF. Both of them have a smaller local truncation error than EBDF, and in the case of EBNDF also the stability region is bigger than the one of the EBDF.

The article is organised as follows: in Section 2 we give details about modifications introduced in EBDF, such as ENDF, ENBDF and EBNDF. In Section 3 the stability analysis is developed and we include some computational aspects as well as numerical examples of ODEs with different stiffness ratios in Section 4.

2. Using NDFs as Predictors in the EBDF Scheme

In this Section we will start analysing the properties of the NDF and EBDF and finally we will derive the ENDF, ENBDF and EBNDF algorithms.

2.1. NDF scheme

Since they were introduced by Gear [6], the Backward differentiation formulae have been widely used due to their good stability properties for solving stiff problems. The BDF of order k can be expressed as follows:

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j y_{n+k} = h f_{n+k}. \tag{2.1}$$

Developing the backward differences of expression (2.1) we get the well-known expression for the BDF:

$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j} = h f_{n+k}. \tag{2.2}$$

The local truncation error (LTE) of the BDF of order k is given by the following expression

$$LTE_k = C_1 h^{k+1} y^{(k+1)}(x_n) + \mathcal{O}\left(h^{k+2}\right), \tag{2.3}$$

where

$$\gamma_k=\sum_{j=1}^{k}\frac{1}{j}=\begin{cases}1, & k=1,\\ 3/2, & k=2,\\ 11/6, & k=3,\\ 25/12, & k=4,\\ 137/60, & k=5.\end{cases} \tag{2.4}$$

Shampine introduces a modification to the BDFs in [10] called NDF, which consists of adding the difference of order $(k+1)$ multiplied by the term $\kappa\gamma_k$ in the BDF formula of order k. The expression of this new method is given by

$$\sum_{j=1}^{k}\frac{1}{j}\nabla^j y_{n+k}=hf_{n+k}+\kappa\gamma_k\nabla^{k+1}y_{n+k}. \tag{2.5}$$

In the same way that we have done for the BDF, an equivalent expression for NDFs can be written as

$$\sum_{j=0}^{k}\hat{\alpha}_j y_{n+j}=hf_{n+k}+\kappa\gamma_k\nabla^{k+1}y_{n+k}. \tag{2.6}$$

The coefficient κ was introduced by Klopfenstein and Shampine, so that the angle of $A(\alpha)$-stability was maximized at the same time that the error was reduced. The NDFs are more precise than the BDF but no more stable. The local truncation error of the NDF is this one:

$$LTE_k=C_2h^{k+1}y^{(k+1)}\left(x_n\right)+\mathcal{O}\left(h^{k+2}\right). \tag{2.7}$$

We will call C_1 and C_2 to the error constants of the LTE of BDF (2.3) and NDF (2.7) methods respectively:

$$C_1=\frac{-1/\gamma_k}{k+1}, \tag{2.8}$$

$$C_2=\frac{-1/\gamma_k}{k+1}-\kappa. \tag{2.9}$$

Table 2.1: The Klopfenstein-Shampine NDFs and their efficiency and $A(\alpha)$-stability relative to the BDFs [10].

k	NDF coefficient κ	Step ratio percent	Stability angle BDF	Stability angle NDF
1	-0.1850	26%	90	90
2	-1/9	26%	90	90
3	-0.0823	26%	86	80
4	-0.0415	12%	73	66

Table 2.2: Values of the error constants C_1, C_2 of the methods BDF and NDF.

k	C_1	C_2
1	-0.5	-0.315
2	-0.222222222	-0.111111111
3	-0.136363636	-0.054063636
4	-0.096	-0.0545

The better accuracy of the NDFs implies that they can achieve the same accuracy as BDFs with a bigger step size. More properties of BDF and NDF methods are shown in Table 2.1. The values of C_1, C_2 for $k = 1, 2, 3, 4$ are in Table 2.2.

2.2. EBDF scheme

With the aim of increasing the stability of the BDF methods, Cash extended these methods by introducing superfuture points (see [1] for more information). The method was called EBDF (extended backward differentiation formula) and its formula is the following:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k f_{n+k} + h\beta_{k+1}\bar{f}_{n+k+1}, \tag{2.10}$$

where the coefficients are adjusted in order to achieve $(k+1)$ order in the formula (2.10). So the coefficients are obtained by solving the following linear system of equations and the normalization $\alpha_k = 1$:

$$\sum_{j=0}^{k} \alpha_j j^q = q\sum_{j=0}^{k} \beta_j j^{q-1}, \qquad \text{for } q = 0, 1, \cdots, k+1.$$

Assuming that the solutions $y_n, y_{n+1}, \cdots, y_{n+k-1}$ are available, the formula (2.10) is used as follows:

1. Compute the first predictor $\bar{y}_{n+k}$ as the solution of the conventional BDF step:
$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j} = hf_{n+k}\ , \quad (y_{n+k} := \bar{y}_{n+k}). \tag{2.11}$$
2. Compute the second predictor $\bar{y}_{n+k+1}$ advancing a new step with the same BDF formula:
$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j+1} = hf_{n+k+1}\ , \quad (y_{n+k+1} := \bar{y}_{n+k+1}). \tag{2.12}$$
3. Evaluate $\bar{f}_{n+k+1} = f(x_{n+k+1}, \bar{y}_{n+k+1})$.
4. Insert $\bar{f}_{n+k+1}$ in (2.10) and solve for a new y_{n+k} which will be the numerical solution of the EBDF method.

The local truncation error of the EBDF method is given by (see [7]):

$$LTE_k = h^{k+2}\left(\beta_{k+1}C_1\left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k}\right)\frac{\partial f}{\partial y}y^{(k+1)} + C_3 y^{(k+2)}\right)(x_n) + \mathcal{O}\left(h^{k+3}\right), \tag{2.13}$$

where C_1 is the error constant of the local truncation error of the BDF method given by (2.8), and C_3 is the error constant of the local truncation error for the formula (2.10).

Lemma 2.1. *If the formula* (2.10) *is of order* $(k+1)$ *and the BDF/NDF used in* (2.11) *and* (2.12) *are of order* k*, then the predictor-corrector algorithm* $(1)-(4)$ *is of order* $(k+1)$.

The demonstration of this lemma can be found in reference [7].

2.3. Scheme related to ENDF, ENBDF, EBNDF methods

2.3.1. ENDF

This method consists of applying the NDF method as the predictor in the EBDF method. Next, we follow the same steps as in the EBDF scheme: we evaluate $\bar{f}_{n+k+1}=f\left(x_{n+k+1}, \bar{y}_{n+k+1}\right)$ and we insert the term $\bar{f}_{n+k+1}$ in the expression (2.10).

First predictor: The first time we apply the predictor NDF, the value $\bar{y}_{n+k}$ is obtained. The difference between the exact value and the calculated is given by this expression:

$$y(x_{n+k}) - \bar{y}_{n+k} = C_2 h^{k+1} y^{(k+1)}\left(x_n\right) + \mathcal{O}\left(h^{k+2}\right), \tag{2.14}$$

where C_2 is the error constant of the method NDF given by (2.9).

Second predictor: The second time we use the predictor NDF we get the value $\bar{y}_{n+k+1}$, and the difference between the exact and the calculated value is this:

$$y(x_{n+k+1}) - \bar{y}_{n+k+1} = C_2\left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha_k}} - \frac{\kappa\gamma_k(k+1)}{\hat{\alpha}_k}\right) h^{k+1} y^{(k+1)}\left(x_n\right) + \mathcal{O}\left(h^{k+2}\right), \tag{2.15}$$

Corrector: Eventually, if we apply the corrector (expression (2.10)), the local truncation error is this one:

$$LTE_k = h^{k+2}\left[\beta_{k+1} C_2\left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha_k}} - \frac{\kappa\gamma_k(k+1)}{\hat{\alpha}_k}\right)\frac{\partial f}{\partial y} y^{(k+1)} + C_3 y^{(k+2)}\right]\left(x_n\right) + \mathcal{O}\left(h^{k+3}\right). \tag{2.16}$$

The first term of the principal local truncation error of the EBDF is affected by the fact of using BDF predictors in the EBDF algorithm (hence the second term of the local truncation error of the EBDF only depends on the corrector expression of the EBDF and this one has not been modified). Comparing the first terms of the local truncation errors of EBDF, expression (2.13), and ENDF, expression (2.16), we see that the latter is smaller, so we have gained efficiency (see Table 2.3).

We will verify for $k = 2$ that the local truncation error of the method after applying the second NDF is the one proposed by expression (2.15). We will apply the first predictor NDF given by (2.6) in order to calculate the value of $\bar{y}_{n+2}$:

$$\hat{\alpha}_0 y_n + \hat{\alpha}_1 y_{n+1} + \hat{\alpha}_2 \bar{y}_{n+2} = h\bar{f}_{n+2} + \kappa\gamma_2\nabla^3\bar{y}_{n+2}.$$

According to (2.7) the difference between the exact solution and the approximated is:

$$y(x_{n+2}) - \bar{y}_{n+2} = C_2 h^3 y'''\left(x_n\right) + \mathcal{O}\left(h^4\right), \qquad \text{where} \qquad C_2 = -2/9 - \kappa. \tag{2.17}$$

Table 2.3: Constant A_k of the local truncation error of EBDF, ENDF, ENBDF, EBNDF.

k	A_k of EBDF	A_k of ENDF	A_k of ENBDF	A_k of EBNDF
1	-1	-0.74655	-0.815	-1
2	-0.518518519	-0.296296296	-0.37037037	-0.481481481
3	-0.359504132	-0.160329154	-0.224831405	-0.322095041
4	-0.28032	-0.17044875	-0.20064	-0.25874

Next, we apply (2.6) again for the second predictor NDF to obtain the value $\bar{y}_{n+3}$:

$$\hat{\alpha}_0 y_{n+1} + \hat{\alpha}_1 \bar{y}_{n+2} + \hat{\alpha}_2 \bar{y}_{n+3} = h\bar{f}_{n+3} + \kappa\gamma_2 \nabla^3 \bar{y}_{n+3}.$$

Developing $\nabla^3 \bar{y}_{n+3}$, we can workout the value of $\bar{y}_{n+3}$ from the last equation. This gives that

$$\bar{y}_{n+3} = \frac{1}{\hat{\alpha}_2}\Big(- \hat{\alpha}_0 y_{n+1} - \hat{\alpha}_1 \bar{y}_{n+2} + h\bar{f}_{n+3} + \kappa\gamma_2 \left(\bar{y}_{n+3} - 3\bar{y}_{n+2} + 3y_{n+1} - y_n\right)\Big).$$

Moreover, the expression of the local truncation error after the second predictor is given by

$$\begin{aligned} &y(x_{n+3}) - \bar{y}_{n+3} \\ =&y(x_{n+3}) - \frac{1}{\hat{\alpha}_2}\Big(- \hat{\alpha}_0 y_{n+1} - \hat{\alpha}_1 \bar{y}_{n+2} + h\bar{f}_{n+3} + \kappa\gamma_2 \left(\bar{y}_{n+3} - 3\bar{y}_{n+2} + 3y_{n+1} - y_n\right)\Big). \end{aligned} \tag{2.18}$$

If we use the expression (2.17) to work out $\bar{y}_{n+2}$ we have:

$$\bar{y}_{n+2} = y(x_{n+2}) - C_2 h^3 y'''(x_n) + \mathcal{O}\left(h^4\right). \tag{2.19}$$

By substituting (2.19) into (2.18), we obtain a new expression of the local truncation error after applying the two predictors:

$$\begin{aligned} y(x_{n+3}) - \frac{1}{\hat{\alpha}_2}\Big(&- \hat{\alpha}_0 y_{n+1} - \hat{\alpha}_1 \left(y(x_{n+2}) - C_2 h^3 y'''(x_n)\right) + h\bar{f}_{n+3}\Big) \\ &- \frac{1}{\hat{\alpha}_2}\Big(\kappa\gamma_2 \left(\bar{y}_{n+3} - 3\left(y(x_{n+2}) - C_2 h^3 y'''(x_n)\right) + 3y_{n+1} - y_n\right)\Big) + \mathcal{O}\left(h^4\right). \end{aligned}$$

By regrouping terms in an appropriate way we have

$$\begin{aligned} y(x_{n+3}) - \frac{1}{\hat{\alpha}_2}\Big(&- \hat{\alpha}_0 y_{n+1} - \hat{\alpha}_1 y(x_{n+2}) + h\bar{f}_{n+3} + \kappa\gamma_2 \left(\bar{y}_{n+3} - 3y(x_{n+2}) + 3y_{n+1} - y_n\right)\Big) \\ &- \frac{\hat{\alpha}_1}{\hat{\alpha}_2} C_2 h^3 y'''(x_n) - 3\kappa\gamma_2 \frac{C_2}{\hat{\alpha}_2} h^3 y'''(x_n) + \mathcal{O}\left(h^4\right). \end{aligned} \tag{2.20}$$

The first two terms in (2.20) are the expression of the local truncation error using the NDF method and from (2.7) we conclude that:

$$\begin{aligned} &y(x_{n+3}) - \frac{1}{\hat{\alpha}_2}\Big(- \hat{\alpha}_0 y_{n+1} - \hat{\alpha}_1 y(x_{n+2}) + h\bar{f}_{n+3} + \kappa\gamma_2 \left(\bar{y}_{n+3} - 3y(x_{n+2}) + 3y_{n+1} - y_n\right)\Big) \\ =&C_2 h^3 y'''(x_n) + \mathcal{O}\left(h^4\right). \end{aligned} \tag{2.21}$$

By substituting (2.21) into (2.20), we finally obtain the local truncation error after applying the two NDF predictors:

$$\begin{aligned} y(x_{n+3}) - \bar{y}_{n+3} =& C_2 h^3 y'''(x_n) - \frac{\hat{\alpha}_1}{\hat{\alpha}_2} C_2 h^3 y'''(x_n) - 3\kappa\gamma_2 \frac{C_2}{\hat{\alpha}_2} h^3 y'''(x_n) + \mathcal{O}\left(h^4\right) \\ =& C_2 \left(1 - \frac{\hat{\alpha}_1}{\hat{\alpha}_2} - \frac{3\kappa\gamma_2}{\hat{\alpha}_2}\right) h^3 y'''(x_n) + \mathcal{O}(h^4). \end{aligned}$$

2.3.2. ENBDF, EBNDF

In the previous Section we have used twice NDF as the predictor of the EBDF algorithm. But the option of using NDF as the predictor in the EBDF is not unique. We can also apply NDF as the first predictor and BDF as the second one or the BDF as the first predictor and NDF as the second one. Hence, the options available for the predictors are the following: BDF-BDF, NDF-NDF, NDF-BDF, BDF-NDF. In all the cases the local truncation error can be expressed in this way:

$$h^{k+2}\left(\beta_{k+1}A_k\frac{\partial f}{\partial y}y^{(k+1)}+C_3y^{(k+2)}\right)(x_n)+\mathcal{O}\left(h^{k+3}\right). \tag{2.22}$$

Where the value of the constant A_k is different depending on the predictors:

- Case EBDF (BDF-BDF-EBDF):

$$A_k=C_1\left(-\hat{\alpha}_{k-1}/\hat{\alpha}_k+1\right); \tag{2.23}$$

- Case ENDF (NDF-NDF-EBDF):

$$A_k=C_2\Big(-\hat{\alpha}_{k-1}/\hat{\alpha}_k-\kappa(k+1)\gamma_k/\hat{\alpha}_k+1\Big); \tag{2.24}$$

- Case ENBDF (NDF-BDF-EBDF):

$$A_k=\left(-C_2\hat{\alpha}_{k-1}/\hat{\alpha}_k+C_1\right); \tag{2.25}$$

- Case EBNDF (BDF-NDF-EBDF):

$$A_k=\left(-C_1\hat{\alpha}_{k-1}/\hat{\alpha}_k-C_1\kappa(k+1)\gamma_k1/\hat{\alpha}_k+C_2\right). \tag{2.26}$$

We can sum up all the cases as follows. The general expression of A_k is this one:

$$A_k=-C_i\left(\hat{\alpha}_{k-1}/\hat{\alpha}_k+\kappa(k+1)\gamma_k1/\hat{\alpha}_k\right)+C_j. \tag{2.27}$$

- When the first predictor is the BDF: C_i=C_1;
- When the first predictor is the NDF: C_i=C_2;
- When the second predictor is the BDF: C_j=C_1, κ=0;
- When the second predictor is the NDF: C_j=C_2 and the values of κ are in Table 2.1.

3. Stability Analysis

3.1. Stability function of ENDF

The expression of the method NDF is given by (2.6):

$$\sum_{j=0}^{k}\hat{\alpha}_jy_{n+j}=hf_{n+k}+\kappa\gamma_k\nabla^{k+1}y_{n+k}.$$

Expression (2.6) can be simplified by substituting $M_k = \kappa\gamma_k$. Therefore, the expression of NDF methods which we are going to work with is of the form

$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j} = hf_{n+k} + M_k \nabla^{k+1} y_{n+k}. \tag{3.1}$$

The region of absolute stability of the overall method ENDF is found using Schur's theorem (see [9]). To do this, we will apply the method ENDF to the test equation $y' = \lambda y$. That is to say, $hf_j = h\lambda y_j$ introduced in expression (2.10) and expression (3.1) is used as the first and second predictor. We will set y_{n-1}=1,. . . ,y_{n+k-1}=r^k and the algorithm will be computed in order to obtain y_{n+k}=r^{k+1}. As a result, the characteristic equation is achieved, being $\hat{h} = h\lambda$:

$$A\hat{h}^3 + B\hat{h}^2 + C\hat{h} + D = 0, \tag{3.2}$$

where

$$A = -\,\beta_k r^{k+1}, \qquad B = 2\left(\hat{\alpha}_k - M_k\right) + T - \beta_{k+1} S, \tag{3.3a}$$

$$\begin{aligned} C = &-\beta_k \left(\hat{\alpha}_k - M_k\right)^2 r^{k+1} - 2\left(\hat{\alpha}_k - M_k\right) T + \left(\hat{\alpha}_k - M_k\right)\beta_{k+1} S \\ &- \beta_{k+1}\left(-\hat{\alpha}_{k-1} - (k+1)M_k\right) R, \end{aligned} \tag{3.3b}$$

$$D = \left(\hat{\alpha}_k - M_k\right)^2 T. \tag{3.3c}$$

In (3.3), some parameters are given below:

$$R = (-1)^{k+1}\binom{k+1}{0} M_k + \sum_{j=1}^{k} r^j \left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j}\binom{k+1}{j} M_k\right), \tag{3.4a}$$

$$S = (-1)^k r M_k - \sum_{j=1}^{k-1} r^{j+1}\left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j}\binom{k+1}{j} M_k\right), \tag{3.4b}$$

$$T = \sum_{j=0}^{k} \alpha_j r^{j+1}. \tag{3.4c}$$

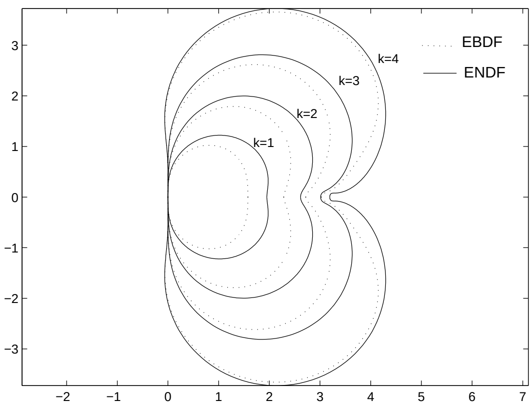

Fig. 3.1. Regions of stability of the methods ENDF and EBDF. The region of stability is the outside of the plotted curve.

Fig. 3.1 shows the stability regions of the ENDF method. We will include the calculations done for the case $k = 2$:

$$\sum_{j=0}^{2} \hat{\alpha}_j y_{n+j} = h\bar{f}_{n+2} + M_k \nabla^3 \bar{y}_{n+2}$$
$$\Rightarrow \hat{\alpha}_0 y_n + \hat{\alpha}_1 y_{n+1} + \hat{\alpha}_2 \bar{y}_{n+2} = h\bar{f}_{n+2} + M_k \nabla^3 \bar{y}_{n+2}. \tag{3.5}$$

Developing $\nabla^3 \bar{y}_{n+2}$ and applying the method given by (3.5) to the test equation, we can obtain

$$\bar{y}_{n+2} = \frac{y_{n+1}\left(-\hat{\alpha}_1 - 3M_k\right) + y_n\left(-\hat{\alpha}_0 + 3M_k\right) - M_k y_{n-1}}{\hat{\alpha}_2 - M_k - \hat{h}}. \tag{3.6}$$

NDF is used again as the second predictor to get $\bar{y}_{n+3}$:

$$\bar{y}_{n+3} = \frac{\bar{y}_{n+2}\left(-\hat{\alpha}_1 - 3M_k\right) + y_{n+1}\left(-\hat{\alpha}_0 + 3M_k\right) - M_k y_n}{\hat{\alpha}_2 - M_k - \hat{h}}. \tag{3.7}$$

Substituting (3.6) into (3.7) we have $\bar{y}_{n+3}$:

$$\begin{aligned}\bar{y}_{n+3} =& \frac{\big(y_{n+1}\left(-\hat{\alpha}_1 - 3M_k\right) + y_n\left(-\hat{\alpha}_0 + 3M_k\right) - M_k y_{n-1}\big)\left(-\hat{\alpha}_1 - 3M_k\right)}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2} \\ &+ \frac{\big(y_{n+1}\left(-\hat{\alpha}_0 + 3M_k\right) - M_k y_n\big)\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2}.\end{aligned}$$

The derivative of $\bar{y}_{n+3}$ is calculated:

$$\begin{aligned}f\left(\bar{y}_{n+3}\right) =& \lambda\left(\frac{\big(y_{n+1}(-\hat{\alpha}_1 - 3M_k) + y_n(-\hat{\alpha}_0 + 3M_k) - M_k y_{n-1}\big)(-\hat{\alpha}_1 - 3M_k)}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2}\right. \\ &\left.+ \frac{\big(y_{n+1}\big(-\hat{\alpha}_0 + 3M_k\big) - M_k y_n\big)\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2}\right).\end{aligned} \tag{3.8}$$

Finally y_{n+2} is obtained using expression (2.10):

$$\begin{aligned}&\alpha_0 y_n + \alpha_1 y_{n+1} + \alpha_2 y_{n+2} - \hat{h}\beta_k y_{n+2} \\ &\quad - \hat{h}\beta_{k+1}\left(\frac{\big(y_{n+1}(-\hat{\alpha}_0 + 3M_k) - M_k y_n\big)(-\hat{\alpha}_2 - M_k - \hat{h})}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2}\right. \\ &\quad \left.+ \frac{\big(y_{n+1}(-\hat{\alpha}_1 - 3M_k) + y_n(-\hat{\alpha}_0 + 3M_k) - M_k y_{n-1}\big)(-\hat{\alpha}_1 - 3M_k)}{\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2}\right) = 0.\end{aligned}$$

Substituting $y_{n+j} = r^j$ the following equation is obtained:

$$\begin{aligned}&\big(\alpha_0 r + \alpha_1 r^2 + \alpha_2 r^3 - \hat{h}\beta_k r^3\big)\big(-\hat{\alpha}_2 - M_k - \hat{h}\big)^2 \\ &\quad + \big(r^2(-\hat{\alpha}_0 + 3M_k) - M_k r\big)\big(\hat{\alpha}_2 - M_k - \hat{h}\big) \\ &\quad - \hat{h}\beta_{k+1}\Big(\big(r^2(-\hat{\alpha}_1 - 3M_k) + r(-\hat{\alpha}_0 + 3M_k) - M_k\big)(-\hat{\alpha}_1 - 3M_k)\Big) = 0.\end{aligned}$$

Grouping the coefficients of the polynomial in $\hat{h}^n$ gives

$$\begin{aligned}
\hat{h}^3 &: \quad A = -\beta_k r^3, \\
\hat{h}^2 &: \quad B = 2\left(\hat{\alpha}_2 - M_k\right)\beta_k r^3 + T - \beta_{k+1} S, \\
\hat{h} &: \quad C = -\beta_k\left(\hat{\alpha}_2 - M_k\right)^2 r^3 - 2\left(\hat{\alpha}_2 - M_k\right) T + \left(\hat{\alpha}_2 - M_k\right)\beta_{k+1} S - \beta_{k+1}\left(-\hat{\alpha}_1 - 3M_k\right) R, \\
\hat{h}^0 &: \quad D = \left(\hat{\alpha}_2 - M_k\right)^2 T,
\end{aligned}$$

where:

$$R = r^2\left(-\hat{\alpha}_1 - 3M_k\right) + r\left(-\hat{\alpha}_0 + 3M_k\right) - M_k,$$

$$S = rM_k - r^2\left(-\hat{\alpha}_0 + 3M_k\right), \qquad T = \sum_{j=0}^{2} \alpha_j r^{j+1}.$$

The stability angles of the method ENDF are given in Table 3.1.

Table 3.1: $A(\alpha)$-stability of the methods EBDF, EBNDF, ENDF, ENBDF.

k	p (order)	$A(\alpha)$ EBDF	$A(\alpha)$ EBNDF	$A(\alpha)$ ENDF	$A(\alpha)$ ENBDF
1	2	90	90	90	90
2	3	90	90	90	90
3	4	90	90	90	90
4	5	87.61	87.68	87.54	87.49

3.2. Stability function of ENBDF

We will apply the ENBDF method to the test equation $y' = \lambda y$. We will substitute $hf_j = \hat{h} y_j$ in expression (2.10). The NDF will be used as the first predictor and the BDF as the second one, where $\hat{h} = h\lambda$. Setting y_{n-1}=1, $\cdots$, y_{n+k-1}=r^k and computing the method we will reach the solution y_{n+k}=r^{k+1} as well as the characteristic equation:

$$A\hat{h}^3 + B\hat{h}^2 + C\hat{h} + D = 0. \tag{3.9}$$

Grouping the coefficients of the polynomial in $\hat{h}^n$ gives

$$A = -\,\beta_k r^{k+1}, \tag{3.10a}$$

$$B = \left(2\hat{\alpha}_k - M_k\right)\beta_k r^{k+1} + T - \beta_{k+1} S, \tag{3.10b}$$

$$C = -\,\beta_k \hat{\alpha}_k\left(\hat{\alpha}_k - M_k\right) r^{k+1} - \left(2\hat{\alpha}_k - M_k\right) T + \left(\hat{\alpha}_k - M_k\right)\beta_{k+1} S + \beta_{k+1}\hat{\alpha}_{k-1} R, \tag{3.10c}$$

$$D = \hat{\alpha}_k\left(\hat{\alpha}_k - M_k\right) T, \tag{3.10d}$$

where

$$R = (-1)^{k+1}\binom{k+1}{0} M_k + \sum_{j=1}^{k} r^j\left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j}\binom{k+1}{j} M_k\right), \tag{3.11a}$$

$$S = \sum_{j=0}^{k-2} \hat{\alpha}_j r^{j+2}, \quad T = \sum_{j=0}^{k} \alpha_j r^{j+1}. \tag{3.11b}$$

We include the calculations for ENBDF2: $\bar{y}_{n+2}$ is predicted using NDF, expression (3.1):

$$\bar{y}_{n+2} = \frac{y_{n+1}\left(-\hat{\alpha}_1 - 3M_k\right) + y_n\left(-\hat{\alpha}_0 + 3M_k\right) - M_k y_{n-1}}{\hat{\alpha}_2 - M_k - \hat{h}}; \tag{3.12}$$

and $\bar{y}_{n+3}$ is predicted using NDF for expression (2.2):

$$\hat{\alpha}_0 y_{n+1} + \hat{\alpha}_1 \bar{y}_{n+2} + \hat{\alpha}_2 \bar{y}_{n+3} = h\bar{f}_{n+3} \ \Rightarrow\ \bar{y}_{n+3} = \frac{-\hat{\alpha}_1 \bar{y}_{n+2} - \hat{\alpha}_0 y_{n+1}}{\hat{\alpha}_2 - \hat{h}}. \tag{3.13}$$

By substituting the expression (3.12) of $\bar{y}_{n+2}$ in (3.13) we have:

$$\begin{aligned}\bar{y}_{n+3} =& \frac{-\hat{\alpha}_1\big(y_{n+1}(-\hat{\alpha}_1 - 3M_k) + y_n(-\hat{\alpha}_0 + 3M_k) - M_k y_{n-1}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)} \\ &- \frac{\hat{\alpha}_0 y_{n+1}\big(\hat{\alpha}_2 - M_k - \hat{h}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)}.\end{aligned}$$

We calculate the derivative of $\bar{y}_{n+3}$:

$$\begin{aligned}f\left(\bar{y}_{n+3}\right) =& \lambda\left(\frac{-\hat{\alpha}_0 y_{n+1}\big(\hat{\alpha}_2 - M_k - \hat{h}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)}\right. \\ &\left. - \frac{\hat{\alpha}_1\big(y_{n+1}(-\hat{\alpha}_1 - 3M_k) + y_n(-\hat{\alpha}_0 + 3M_k) - M_k y_{n-1}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)}\right).\end{aligned}$$

Finally we calculate y_{n+2} using expression (2.10):

$$\begin{aligned}&\alpha_0 y_n + \alpha_1 y_{n+1} + \alpha_2 y_{n+2} - h\beta_k f_{n+2} - \hat{h}\beta_{k+1}\left(-\frac{\hat{\alpha}_0 y_{n+1}\big(\hat{\alpha}_2 - M_k - \hat{h}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)}\right. \\ &\qquad\left. - \frac{\hat{\alpha}_1\big(y_{n+1}(-\hat{\alpha}_1 - 3M_k) + y_n(-\hat{\alpha}_0 + 3M_k) - M_k y_{n-1}\big)}{\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big)}\right) = 0.\end{aligned} \tag{3.14}$$

By replacing $y_{n-1} = 1$, $y_n = r$, $y_{n+1} = r^2$, $y_{n+2} = r^3$ in (3.14) we obtain the following:

$$\begin{aligned}&\big(\alpha_0 r + \alpha_1 r^2 + \alpha_2 r^3 - \hat{h}\beta_k r^3\big)\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\big(\hat{\alpha}_2 - \hat{h}\big) \\ &\quad - \hat{h}\beta_{k+1}\Big(-\hat{\alpha}_1\big(r^2(-\hat{\alpha}_1 - 3M_k) + r(-\hat{\alpha}_0 + 3M_k) - M_k\big) - \hat{\alpha}_0 r^2\big(\hat{\alpha}_2 - M_k - \hat{h}\big)\Big) = 0.\end{aligned}$$

The coefficients of the polynomial are given by

$$\begin{aligned}\hat{h}^3:&\quad A = -\beta_k r^3, \\ \hat{h}^2:&\quad B = \left(2\hat{\alpha}_2 - M_k\right)\beta_k r^3 + T - \beta_{k+1}S, \\ \hat{h}:&\quad C = -\beta_k\hat{\alpha}_2\left(\hat{\alpha}_2 - M_k\right)r^3 - \left(2\hat{\alpha}_2 - M_k\right)T + \left(\hat{\alpha}_2 - M_k\right)\beta_{k+1}S + \beta_{k+1}\hat{\alpha}_1 R, \\ \hat{h}^0:&\quad D = \hat{\alpha}_2\left(\hat{\alpha}_2 - M_k\right)T,\end{aligned}$$

where

$$R = r^2\left(-\hat{\alpha}_1 - 3M_k\right) + r\left(-\hat{\alpha}_0 + 3M_k\right) - M_k,$$

$$S = r^2\hat{\alpha}_0, \qquad T = \sum_{j=0}^{2}\alpha_j r^{j+1}.$$

The stability angles of the method are also given in Table 3.1.

3.3. Stability function of EBNDF

Proceeding in the same way as before, the characteristic polynomial is

$$A\hat{h}^3 + B\hat{h}^2 + C\hat{h} + D = 0. \tag{3.15}$$

Grouping the coefficients of the polynomial in $\hat{h}^n$ gives

$$A = -\beta_k r^k, \quad B = (2\hat{\alpha}_k - M_k)\,\beta_k r^k + T - \beta_{k+1} S, \tag{3.16a}$$

$$C = -\beta_k \hat{\alpha}_k\,(\hat{\alpha}_k - M_k)\,r^k - (2\hat{\alpha}_k - M_k)T + \hat{\alpha}_k \beta_{k+1} S + \beta_{k+1}\big(-\hat{\alpha}_{k-1} - (k+1)M_k\big)R, \tag{3.16b}$$

$$D = \hat{\alpha}_k\,(\hat{\alpha}_k - M_k)\,T, \tag{3.16c}$$

where

$$R = \sum_{j=0}^{k-1} \hat{\alpha}_j r^j, \qquad T = \sum_{j=0}^{k} \alpha_j r^j, \tag{3.17a}$$

$$S = (-1)^k M_k - \sum_{j=1}^{k-1} r^j \left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j}\binom{k+1}{j} M_k\right). \tag{3.17b}$$

3.4. Conclusions about the stability regions

As we can see in Table 3.1, the $A(\alpha)$-stability angles of EBDF and EBNDF are larger than the angles corresponding to ENDF and ENBDF, respectively. So the stability properties are better whenever we apply the BDF as the first predictor.

The $A(\alpha)$-stability angles of the EBNDF are bigger than the angles of the EBDF, and the $A(\alpha)$-stability angles of the ENDF are bigger than the angles of ENBDF. We can conclude that after using the BDF as the first predictor, it is better to use the NDF as the second predictor.

Taking these considerations into account, the EBNDF method has a larger stability region than the EBDF. In Section 2 we have seen that the local truncation error of the EBNDF is smaller than the one of the EBDF. So EBNDF is better than EBDF in both aspects, stability and accuracy.

4. Numerical Results

4.1. Computational aspects

The methods BDF and NDF used as predictors can be written using backward differences (expressions (2.1) and (2.5)). We will write the corrector of the EBDF method (expression (2.10)) using backward differences too. In this way we can use the same scheme during the programming. Expression (2.10) can be written as follows using backward differences:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k f_{n+k} + h\beta_{k+1}\bar{f}_{n+k+1}$$

$$\Rightarrow \quad \sum_{j=1}^{k} m_{k,j} \nabla^j y_{n+k} = h\beta_k f_{n+k} + h\beta_{k+1}\bar{f}_{n+k+1}, \tag{4.1}$$

where

$$M=(m_{k,j})=\begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{18}{23} & \frac{5}{23} & 0 & 0 \\ \frac{132}{197} & \frac{48}{197} & \frac{17}{197} & 0 \\ \frac{1500}{2501} & \frac{606}{2501} & \frac{284}{2501} & \frac{111}{2501} \end{pmatrix}. \tag{4.2}$$

The coefficients corresponding to k are in the k-th row of the matrix M.

An alternative way to write the left hand side of the expressions (2.1) and (2.5), expressions corresponding to BDFs and NDFs, is introduced in [10] and the following is the expression used for the predictors:

$$\sum_{j=1}^{k}\frac{1}{j}\nabla^j y_{n+k}=\gamma_k\left(y_{n+k}-y_{n+k}^{(0)}\right)+\sum_{j=1}^{k}\gamma_j\nabla^j y_{n+k-1}, \tag{4.3}$$

where

$$\gamma_j=\sum_{l=1}^{j}\frac{1}{l}, \tag{4.4a}$$

$$y_{n+k}^{(0)}=\sum_{j=0}^{k}\nabla^j y_{n+k-1}=\nabla^0 y_{n+k-1}+\nabla y_{n+k-1}+\cdots+\nabla^k y_{n+k-1}, \tag{4.4b}$$

$$y_{n+k}-y_{n+k}^{(0)}=\nabla^{k+1}y_{n+k}. \tag{4.4c}$$

The identity (4.3) shows that Eqs. (2.1) and (2.5) are equivalent to

$$(1-\kappa)\gamma_k\left(y_{n+k}-y_{n+k}^{(0)}\right)+\sum_{j=1}^{k}\gamma_j\nabla^j y_{n+k-1}=hf_{n+k}, \tag{4.5}$$

where in the case of the BDFs $\kappa=0$ and the values of κ are in Table 2.1 for the NDFs. We have evaluated the implicit formula (4.5) using the Newton method, and the correction to the current iterate $y_{n+k}^{(i+1)}=y_{n+k}^{(i)}+\Delta^{(i)}$ is obtained by solving:

$$\begin{aligned}&\left(I-\frac{h}{(1-\kappa)\gamma_k}J\right)\Delta^{(i)}\\ =&\frac{h}{(1-\kappa)\gamma_k}f\left(x_{n+k},y_{n+k}^{(i)}\right)-\frac{1}{(1-\kappa)\gamma_k}\sum_{j=1}^{k}\gamma_j\nabla^j y_{n+k-1}-\left(y_{n+k}^{(i)}-y_{n+k}^{(0)}\right),\end{aligned} \tag{4.6}$$

where J is the Jacobian of $f(x,y)$. We have used the previous idea to develop and alternative formula of the left hand side of (4.1) and we have obtained the next expression for the corrector of the EBDF:

$$\sum_{j=1}^{k}m_{k,j}\nabla^j y_{n+k}=\tilde{\gamma}_{k,k}\left(y_{n+k}-y_{n+k}^{(0)}\right)+\sum_{j=1}^{k}\tilde{\gamma}_{k,j}\nabla^j y_{n+k-1} \tag{4.7}$$

where:

$$(\tilde{\gamma}_{k,j}) = \left(\sum_{l=1}^{j} m_{k,l}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{18}{23} & 1 & 0 & 0 \\ \frac{132}{197} & \frac{180}{197} & 1 & 0 \\ \frac{1500}{2501} & \frac{2106}{2501} & \frac{2390}{2501} & 1 \end{pmatrix}. \tag{4.8}$$

Taking into account expressions (4.7) and (4.8), expression (4.1) can be written as

$$\left(y_{n+k} - y_{n+k}^{(0)}\right) + \sum_{j=1}^{k} \frac{\tilde{\gamma}_{k,j}}{\tilde{\gamma}_{k,k}} \nabla^j y_{n+k-1} = h\frac{\beta_k}{\tilde{\gamma}_{k,k}} f_{n+k} + h\frac{\beta_{k+1}}{\tilde{\gamma}_{k,k}} \bar{f}_{n+k+1}. \tag{4.9}$$

We will compute y_{n+k} as the solution of the implicit formula (4.9) using Newton's method. The correction of the current iterate $y_{n+k}^{(i+1)} = y_{n+k}^{(i)} + \Delta^{(i)}$, is obtained by solving the following equation

$$\begin{aligned} \left(I - h\frac{\beta_k}{\tilde{\gamma}_{k,k}} J\right) \Delta^{(i)} &= h\frac{\beta_k}{\tilde{\gamma}_{k,k}} f\left(x_{n+k}, y_{n+k}^{(i)}\right) + h\frac{\beta_{k+1}}{\tilde{\gamma}_{k,k}} \bar{f}_{n+k+1} \\ &\quad - \sum_{j=1}^{k} \frac{\tilde{\gamma}_{k,j}}{\tilde{\gamma}_{k,k}} \nabla^j y_{n+k-1} - \left(y_{n+k}^{(i)} - y_{n+k}^{(0)}\right), \end{aligned} \tag{4.10}$$

where J is the Jacobian of $f(x, y)$.

4.2. Numerical results

In this section we show some numerical results as well as we make a comparison between the results obtained using the methods EBDF, ENDF, ENBDF, EBNDF.

Example 4.1. Consider the following stiff system as considered by Cash in [1]:

$$\begin{cases} y_1' = -y_1 - 15y_2 + 15e^{-x}, \\ y_2' = 15y_1 - y_2 - 15e^{-x}, \end{cases}$$

with initial value $y(0) = (1, 1)^T$. Its exact solution is this one: $y_1(x) = y_2(x) = e^{-x}$.

The eigenvalues of the Jacobian matrix are $-1 \pm 15i$, so they lie close to the imaginary axis. First of all, we have compared the 3-step ENDF ($k = 3$) scheme and the 4-step NDF ($k = 4$) scheme, both of order 4, in order to confirm the superior stability performance of the ENDF in relation to the NDF. We have taken 100 steps in both cases and it can be seen in Table 4.1 that the 3-step ENDF remains stable while the 4-step NDF scheme becomes unstable.

In Table 4.2 we tabulate the error results when the system is integrated using four different algorithms. We can see as expected that better accuracy is obtained by EBNDF, ENBDF and ENDF than by the EBDF. We have taken 500 steps and $k = 4$ to integrate Example 4.1. % EBNDF, % ENBDF and % ENDF are the error percentages of EBNDF, ENBDF, ENDF respectively in relation to the error of EBDF.

Table 4.1: Results for integration of Example 4.1 using 4-step NDF and 3-step ENDF.

x	y_i	Exact solution	Error in 4-step NDF	Error in 3-step ENDF
5	y_1	$0.673794699908547e^{-2}$	$0.370441483455641e^{-1}$	$0.188662337274360e^{-6}$
	y_2	$0.673794699908547e^{-2}$	0.166672447133115	$0.214971188146514e^{-6}$
10	y_1	$0.453999297624848e^{-4}$	$0.115006674043339e^{+2}$	$0.720924919432174e^{-9}$
	y_2	$0.453999297624848e^{-4}$	0.701019886357578	$0.732274686539498e^{-9}$
20	y_1	$0.206115362243856e^{-8}$	$0.507771717169461e^{+5}$	$0.325519853141565e^{-13}$
	y_2	$0.206115362243856e^{-8}$	$0.132236107126323e^{+5}$	$0.335357982679398e^{-13}$

Table 4.2: Results for integration of Example 4.1.

x	y_i	Exact solution	Error in EBDF	Error in EBNDF	% EBNDF	Error in ENBDF	% ENBDF	Error in ENDF	% ENDF
5	y_1	$0.673794699908547e^{-2}$	$0.39e^{-5}$	$0.34e^{-5}$	87.1	$0.32e^{-5}$	82.8	$0.26e^{-5}$	67.2
	y_2	$0.673794699908547e^{-2}$	$0.17e^{-5}$	$0.13e^{-5}$	76.0	$0.16e^{-5}$	91.4	$0.11e^{-5}$	65.1
10	y_1	$0.453999297624848e^{-4}$	$0.27e^{-7}$	$0.24e^{-7}$	89.2	$0.22e^{-7}$	79.3	$0.18e^{-7}$	65.9
	y_2	$0.453999297624848e^{-4}$	$0.33e^{-7}$	$0.26e^{-7}$	80.4	$0.27e^{-7}$	82.8	$0.20e^{-7}$	61.8
20	y_1	$0.206115362243856e^{-8}$	$0.97e^{-12}$	$0.59e^{-12}$	61.5	$0.69e^{-12}$	71.9	$0.37e^{-12}$	38.6
	y_2	$0.206115362243856e^{-8}$	$0.42e^{-11}$	$0.35e^{-11}$	82.7	$0.32e^{-11}$	76.3	$0.25e^{-11}$	58.2

Example 4.2. Consider the system of differential equations:

$$\begin{cases} y_1' = -20y_1 - 0.25y_2 + 19.75y_3, \\ y_2' = 20y_1 - 20.25y_2 + 0.25y_3, \\ y_3' = 20y_1 - 19.75y_2 - 0.25y_3, \end{cases}$$

with initial value $y(0) = (1, 0, -1)^T$. The exact solution is

$$\begin{aligned} y_1(x) &= \frac{1}{2}\Big(e^{-0.5x} + e^{-20x}\,(\cos 20x + \sin 20x)\Big), \\ y_2(x) &= \frac{1}{2}\Big(e^{-0.5x} - e^{-20x}\,(\cos 20x - \sin 20x)\Big), \\ y_3(x) &= -\frac{1}{2}\Big(e^{-0.5x} + e^{-20x}\,(\cos 20x - \sin 20x)\Big). \end{aligned}$$

The system has been integrated by EBDF, EBNDF, ENBDF and ENDF and the results are tabulated in Table 4.3. We have taken 50 steps and $k = 3$ to integrate Example 4.2 and we can see that results obtained by the last three methods are superior to that obtained by the EBDF.

Example 4.3. We consider another stiff system as considered by Hosseini and Hojjati in [8]:

$$\begin{cases} y_1' = -0.1y_1 - 49.9y_2, \\ y_2' = -50y_2, \\ y_3' = 70y_2 - 120y_3, \end{cases}$$

with initial value $y(0) = (2, 1, 2)^T$. The stiffness ratio of this problem is 1200 and the exact solution is

$$y_1(x) = e^{-50x} + e^{-0.1x}, \quad y_2(x) = e^{-50x}, \quad y_3(x) = e^{-50x} + e^{-120x}.$$

In Table 4.4 we list the results of the computed solutions. We have taken 50 steps and $k = 4$ to integrate Example 4.3. Again, errors obtained by EBNDF, ENBDF and ENDF are smaller than the ones obtained by the EBDF as implied by expressions (2.23)-(2.26).

Table 4.3: Results for integration of Example 4.2.

x	y_i	Exact solution	Error in EBDF	Error in EBNDF	% EBNDF	Error in ENBDF	% ENBDF	Error in ENDF	% ENDF
	y_1	0.303265331217737	$0.38e^{-3}$	$0.25e^{-3}$	65.9	$0.12e^{-3}$	30.9	$0.47e^{-4}$	12.3
5	y_2	0.303265330376617	$0.14e^{-2}$	$0.12e^{-2}$	86.7	$0.11e^{-4}$	77.1	$0.85e^{-3}$	59.7
	y_3	-0.303265329336016	$0.91e^{-3}$	$0.76e^{-3}$	83.6	$0.62e^{-3}$	68.8	$0.42e^{-3}$	46.8
	y_1	$0.410424993119494e^{-1}$	$0.36e^{-4}$	$0.34e^{-4}$	92.1	$0.33e^{-4}$	91.4	$0.30e^{-4}$	82
10	y_2	$0.410424993119494e^{-1}$	$0.36e^{-4}$	$0.34e^{-4}$	92.1	$0.33e^{-4}$	91.4	$0.30e^{-4}$	82
	y_3	$-0.410424993119494e^{-1}$	$0.36e^{-4}$	$0.34e^{-4}$	92.1	$0.33e^{-4}$	91.4	$0.30e^{-4}$	82
	y_1	$0.336897349954273e^{-2}$	$0.31e^{-5}$	$0.29e^{-5}$	92.2	$0.28e^{-5}$	91.2	$0.25e^{-5}$	81.9
20	y_2	$0.336897349954273e^{-2}$	$0.31e^{-5}$	$0.29e^{-5}$	92.2	$0.28e^{-5}$	91.2	$0.25e^{-5}$	81.9
	y_3	$-0.336897349954273e^{-2}$	$0.31e^{-5}$	$0.29e^{-5}$	92.2	$0.28e^{-5}$	91.2	$0.25e^{-5}$	81.9

Table 4.4: Results for integration of Example 4.3.

x	y_i	Exact solution	Error in EBDF	Error in EBNDF	% EBNDF	Error in ENBDF	% ENBDF	Error in ENDF	% ENDF
	y_1	0.996787780748254	$0.26e^{-2}$	$0.24e^{-2}$	92.9	$0.24e^{-2}$	92.1	$0.22e^{-2}$	83.1
5	y_2	$0.673794699908547e^{-2}$	$0.26e^{-2}$	$0.24e^{-2}$	92.9	$0.24e^{-2}$	92.1	$0.22e^{-2}$	83.1
	y_3	$0.674409121143880e^{-2}$	$0.23e^{-2}$	$0.19e^{-2}$	84.5	$0.20e^{-2}$	88.0	$0.15e^{-2}$	67.5
	y_1	0.951229424514602	$0.87e^{-8}$	$0.80e^{-8}$	91.2	$0.79e^{-8}$	89.9	$0.70e^{-8}$	79.7
10	y_2	$0.138879438649640e^{-10}$	$0.23e^{-9}$	$0.18e^{-9}$	79.8	$0.15e^{-9}$	64.4	$0.10e^{-9}$	44.5
	y_3	$0.138879438649640e^{-10}$	$0.61e^{-9}$	$0.48e^{-9}$	77.8	$0.54e^{-9}$	87.6	$0.38e^{-9}$	61.1
	y_1	0.904837418035960	$0.81e^{-8}$	$0.74e^{-8}$	91.5	$0.73e^{-8}$	90.6	$0.65e^{-8}$	80.7
20	y_2	$0.192874984796392e^{-21}$	$0.56e^{-18}$	$0.38e^{-18}$	68.1	$0.20e^{-18}$	35.2	$0.17e^{-18}$	30.3
	y_3	$0.192874984796392e^{-21}$	$0.15e^{-17}$	$0.95e^{-18}$	63.3	$0.12e^{-17}$	76.8	$0.48e^{-18}$	32.1

Acknowledgments. The author E. Alberdi would like to thank Lea Artibai ikastetxea for the opportunity she has been given to do research on this area.

References

[1] J.R. Cash, On the integration of stiff systems of ODEs using extended backward differentiation formula, *Numer. Math.*, **34**:2 (1980), 235-246.

[2] J.R. Cash, The integration of stiff initial value problems in ODEs using modified extended backward differentiation formula, *Comut. Math. Appl.*, **9**:5 (1983), 645-657.

[3] J.R. Cash, S. Considine, An MEBDF code for initial value problems, *ACM T. Math. Software*, **18** (1992), 142-160.

[4] G. Dahlquist, A special stability problem for linear multistep methods, *BIT*, **3** (1963), 27-43.

[5] C. Fredebeul, A-BDF: a generalization of the backward differentiation formulae, *SIAM J. Numer. Anal.*, **35**:5 (1998), 1917-1938.

[6] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, New Jersey, 1971.

[7] E. Hairer, G. Wanner, Solving ordinary differential equations, II, Stiff and Differential-Algebraic Problems, Springer, Berlin, 1996.

[8] S.M. Hosseini and G. Hojjati, Matrix-free MEBDF method for numerical solution of systems of ODEs, *Math. Comput. Model.*, **29** (1999), 67-77.

[9] J.D. Lambert, Computational Methods in Ordinary Differential Equations, Wiley, New York, 1972.

[10] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, *SIAM J. Sci. Comput.*, **18**:1 (1997), 1-22.