Acta Universitatis Apulensis
ISSN: 1582-5329

No. 30/2012
pp. 171-188

A CLASS OF SECOND DERIVATIVE MULTISTEP METHODS FOR STIFF SYSTEMS

M. Mehdizadeh Khalsaraei, N. Nasehi Oskuyi and G. Hojjati

Abstract. In this paper we present details of a new class of implicit formulas of linear multistep methods to integrate ordinary differential equations (ODE) numerically. The formulas require the knowledge of functions not only at the past and present step-points, but also at the one future point. In order to obtain higher order A-stable, multistep methods, we have used second derivative of the solutions, additional stages and one future point, which give a class of efficient linear multistep methods. This approach allows us to develop L-stable schemes of order up to 8 and L(α)-stable schemes of order up to 12. The stability analysis is discussed and an improvement is obtained in stability region. A good comparison between the results of this class and the results due to others, demonstrate that this class is suitable for stiff systems.

Keywords: Stiff ODEs, Multistep and Multiderivative Methods, Stability aspects.

2000 *Mathematics Subject Classification*: 65Lxx, 65L05.

1. Introduction

In recent years the problems of deriving more advanced and efficient methods for stiff problems has received a great deal of attention, and as a result a wide variety of approaches have been proposed. A potentially good numerical method for the solution of stiff systems of ODEs must have good accuracy and some reasonably wide region of absolute stability [2]. One of the first and most important stability requirement, particulary for linear multistep method, is A-stability which was proposed in [3]. However, the requirement of A-stability puts some limitations on the choice of suitable linear multistep methods. Dahlquist proved that the order of an A-stable linear multistep method ≤ 2 and that an A-stable multistep method must be implicit. This pessimistic result has encouraged researchers to seek other classes of numerical methods for solving stiff equations.

The search for higher order A-stable multistep methods is carried out in the two main directions:

- use higher derivatives of the solutions,
- throw in additional stages, off-step points, super-future points and like. This leads into the large field general linear methods[4].

One successful scheme in this direction was proposed by Enright[3] that used second derivative of solution in his algorithm, Cash[1], Ismail[6], Hojjati[5] and Mehdizadeh[9] introduced second derivative multistep methods that have good stability properties. These methods are A-stable of high orders.

Here we construct a new class of A-stable methods for the numerical solution of the stiff initial value problems. The algorithms which we shall derive will be seen to require more work per step, but to have higher orders of accuracy and better stability characteristics, than other second derivative multistep methods. This extra stability is particularly important when integrating stiff systems of equations having a jacobian with eigenvalues of large modulus lying close to the imaginary axis. In the second section, second derivative multistep method (SDMM) is described . In the third section, we introduce the details of our new method. Accuracy and the stability behavior of our approach is analyzed in the forth section, and a comparison is made with existing methods for A-stability orders. The numerical solutions and a comparison have been shown with some methods for results in the final section.

2.Second derivative multistep methods (SDMM)

Starting with the second page the header should contain the name of the author and a short title of the paper

Let us consider the stiff initial value problem

$$y'(x) = f(x, y(x)), \qquad y(x_0) = y_0,$$

on the finite interval $I = [x_0, x_N]$ where $y : I \to R^m$ and $f : I \times R^m \to R^m$ is continuous and differentiable.

A SDMM can be written in the form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} + h^2 \sum_{j=0}^{k} \gamma_j g_{n+j}, \tag{1}$$

where α_j, β_j and γ_j are parameters to be determined and $g_{n+j} = f^{(1)}_{n+j}$. If either β_k or γ_k is nonzero, the formula will be implicit.

Taylor expansion shows that method (1) is of order p if and only if

$$\sum_{j=0}^{k} \alpha_j j^q = q \sum_{j=0}^{k} \beta_j j^{q-1} + q(q+1) \sum_{j=0}^{k} \gamma_j j^{q-2}, \qquad 0 \leq p \leq q,$$

Some known important SDMM schemes that will be used for comparison are as follows:

• The Enright[3] k-step formulas of order $k+2$ which takes the following form:

$$y_{n+1} - y_n = h \sum_{j=0}^{k} \beta_j f_{n+j-k+1} + h^2 \gamma_k g_{n+1}.$$

• Second derivative extended backward differentiation formulas (E2BD), that was introduced by Cash[1] with the following form:

class 1 :

Predictor: $y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k} \beta_j f_{n+j} + h^2 \gamma_k g_{n+k},$

Corrector: $y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k+1} \bar{\beta}_j f_{n+j} + h^2 (\bar{\gamma}_k g_{n+k} + \bar{\gamma}_{k+1} g_{n+k+1}),$

class 2 :

Predictor: $y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k} \beta_j f_{n+j} + h^2 \gamma_k g_{n+k},$

Corrector: $y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k+1} \bar{\beta}_j f_{n+j} + h^2 \bar{\gamma}_k g_{n+k}.$

These are of order $k+2$.

• Special class of SDMM, introduced by Ismail and Ibrahim [6] of the form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \beta_k (f_{n+k} - \beta^* f_{n+k-1}) + h^2 \gamma_k (g_{n+k} - \gamma^* g_{n+k-1}).$$

For $\beta^* = 0$, $\gamma^* = 0$ this is the same as the SDBDF [4] method.

- Hojjati[5] introduced second derivative multistep method as follows:

Predictor: $\sum_{j=0}^{k} \alpha_j y_{n+j+1} = h\beta_k f_{n+k+1} + h^2\gamma_k g_{n+k+1}$,

Corrector: $\sum_{j=0}^{k} \bar{\alpha}_j y_{n+j} = h\bar{\beta}_k f_{n+k} + h^2(\bar{\gamma}_k g_{n+k} - \bar{\gamma}_{k+1} g_{n+k+1})$.

This method is of order $k+2$. The regions of A(α)-stability of above methods are given in Table 3.

3.The new method

Let us consider the new SDMM of the form:

$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j} = h(\hat{\beta}_k f_{n+k} + \hat{\beta}_{k+1} f_{n+k+1}) + h^2(\hat{\gamma}_k g_{n+k} + \hat{\gamma}_{k+1} g_{n+k+1}), \qquad (2)$$

where $g(x,y) = y'' = f_x + f_y f$, $\hat{\alpha}_k = 1$ and the other coefficients are chosen so that (2) has order $k+3$.

The coefficients of k-step methods of class (2) are given in table 1, for $k = 1, 2, \ldots, 6$. It has used one super-future point technique and designed so that to have good stability properties with high order of accuracy. Starting from given data $y_n, y_{n+1}, \ldots, y_{n+k-1}$, a predictor is first used to predict y_{n+k+1}, the derivative approximations y'_{n+k+1}, y''_{n+k+1} are then computed and finally y_{n+k} is computed from $y_n, y_{n+1}, \ldots, y_{n+k-1}, y'_{n+k+1}, y''_{n+k+1}$. The way in which (2) is used in practice is by carry out the following computations:

stage 1 : Compute $\bar{y}_{n+k}$ as the solution of

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k f_{n+k} + h^2\gamma_k g_{n+k}, \qquad (3)$$

where $\alpha_k = 1$ and the other coefficients are chosen so that (3) has order $k+1$. The coefficients of these methods are represented in table 2, for $k = 1, 2, \ldots, 6$.

stage 2 : Compute $\bar{y}_{n+k+1}$ as the solution of

$$\sum_{j=0}^{k} \alpha_j y_{n+j+1} = h\beta_k f_{n+k+1} + h^2\gamma_k g_{n+k+1}. \qquad (4)$$

stage 3 : Evaluate

$$\bar{g}_{n+k+1} = g(x_{n+k+1}, \bar{y}_{n+k+1}). \tag{5}$$

stage 4 : Compute y_{n+k} as the solution of

$$\sum_{j=0}^{k} \hat{\alpha}_j y_{n+j} = h(\hat{\beta}_k f_{n+k} + \hat{\beta}_{k+1}\bar{f}_{n+k+1}) + h^2(\hat{\gamma}_k g_{n+k} + \hat{\gamma}_{k+1}\bar{g}_{n+k+1}). \tag{6}$$

Note that at each stages 1,2 and 4 a system of nonlinear equation must be solved in order that the desired approximation can be computed. Usually, to solve these nonlinear systems, a modified Newton method is used. Then a direct method is used to solve any resulting system of linear equations. Hence, in each stage, it is necessary to obtain the Jacobian matrix, the related LU factorization matrices and a forward elimination and back substitution to solve a linear system. In order to save computational effort, the approach described above can be modified such that the Jacobian matrix to be the same in each of 3 steps. We observed that the solution of system of ODEs (1) reduced to the solution of the following system of (generally) nonlinear equations:

$$y_{n+k} - h\beta_k f(x_{n+k}, y_{n+k}) - h^2\gamma_k g(x_{n+k}, y_{n+k}) - a_{n+k} = 0,$$

where $a_{n+k} = -\sum_{j=0}^{k-1} a_j y_{n+j}$. If we let

$$X_{n+k} = h\beta_k f(x_{n+k}, y_{n+k}) + h^2\gamma_k g(x_{n+k}, y_{n+k}) = y_{n+k} - a_{n+k},$$

then we have the following system of nonlinear equations to be solved:

$$F(X_{n+k}) = X_{n+k} - h\beta_k f(x_{n+k}, a_{n+k} + X_{n+k}) - h^2\gamma_k g(x_{n+k}, a_{n+k} + X_{n+k}) = 0.$$

After applying a modified Newton method, we have

$$\begin{aligned} &\big(I - h\beta_k \tfrac{\partial f}{\partial y}(x_{n+k}, y_{n+k}^{(m)})\big)\Delta y_{n+k}^{(m)} = h\beta_k f(x_{n+k}, y_{n+k}^{(m)}) + h^2\gamma_k g(x_{n+k}, y_{n+k}^{(m)}) - X_{n+k}, \\ &X_{n+k}^{(m+1)} = \Delta y_{n+k}^{(m)} + X_{n+k}^{(m)}. \end{aligned} \tag{7}$$

In each step, we predict a value $y_{n+k}^{(0)}$ using a suitable one-step method say, one of the Runge-Kutta methods, and then using $X_{n+k}^{(0)} = y_{n+k}^{(0)} - a_{n+k}$, we predict $X_{n+k}^{(0)}$. Hence, the first system of linear equations to be solved in the n^{th} step is $AX = b$ where

$$A = F'(X_{n+k}^{(0)}) = I - h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, y_{n+k}^{(0)}) - h^2\gamma_k g(x_{n+k}, y_{n+k}^{(0)}),$$

$$b = -F(X^{(0)}_{n+k}) = h\beta_k f(x_{n+k}, y^{(0)}_{n+k}) + h^2\gamma_k g(x_{n+k}, y^{(0)}_{n+k}) - X^{(0)}_{n+k}.$$

In stages 1 and 2 the Jacobian matrix is $I - h\beta_k \frac{\partial f}{\partial y} - h^2\gamma_k \frac{\partial g}{\partial y}$ and for step 4 the Jacobian matrix is $I - h\hat{\beta}_k \frac{\partial f}{\partial y} - h^2\hat{\gamma}_k \frac{\partial g}{\partial y}$. By changing stage 4 to

Stage 4* :

$$\begin{aligned} y_{n+k} - h\beta_k f_{n+k} - h^2\gamma_k g_{n+k} = &-\textstyle\sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} + h(\hat{\beta}_k - \beta_k)\bar{f}_{n+k} \\ &+h(\bar{f}_{n+k+1} + h\hat{\gamma}_{k+1}\bar{g}_{n+k+1}) + h^2(\hat{\gamma}_k - \gamma_k)\bar{g}_{n+k}, \end{aligned}$$

the Jacobian matrix in each of 3 steps 1,2 and 4* is the same as $I - h\beta_k \frac{\partial f}{\partial y} - h^2\gamma_k \frac{\partial g}{\partial y}$. Therefore, when using an iteration to compute y_{n+k} we use the same coefficient matrix as was used with stage 1 and use the initial approximation $\bar{y}_{n+k}$ to y_{n+k}. Practical experience has shown that this iteration scheme converges very fast. Fortunately, this modification improves the computational efficiency of this approach. Another important computational aspect which we consider is that variable stepsize. This is the idea of estimating of the error in $\bar{y}_{n+k}$ used by Cash[1]. It has used the quantity

$$\eta_{n+k} = \|y_{n+k} - \bar{y}_{n+k}\|_\infty, \tag{8}$$

as an estimate of the local truncation error in $\bar{y}_{n+k}$, and control the steplength of integration on the basis of this estimate. If a local error tolerance, Tol, is imposed at each step, the relationships between the new step h' and the old step h are as follows:

1) if $\eta_{n+k} > tol$, reject y_{n+k} and set $h' = h/2$;

2) if $tol > \eta_{n+k} > tol/\mu$ where $\mu = 2.5 \times 2^{k+1}$, accept y_{n+k} and put $h' = h$;

3) if $\eta_{n+k} < \mu^i tol$, $i = 1, 2, 3, 4$, accept y_{n+k}. If h has not been changed for at least $k + 2$ steps, set $h' = 2^i h$. Otherwise, keep h fixed.

k	1	2	3	4	5	6
d	12	481	27703	3852793	123941911	7439022169
$\hat{\beta}_k$	-6/d	178/d	16014/d	2506548/d	84099180/d	5119979220/d
$\hat{\beta}_{k+1}$	18/d	272/d	8586/d	771552/d	17616000/d	797544000/d
$\hat{\gamma}_k$	-17/d	-374/d	-15462/d	-1716408/d	-46636200/d	-2448145800/d
$\hat{\gamma}_{k+1}$	-7/d	-92/d	-2646/d	-222048/d	-4806000/d	-208332000/d
$\hat{\alpha}_0$	-1	31/d	-325/d	13023/d	-157036/d	4192900/d
$\hat{\alpha}_1$		-512/d	3753/d	-141616/d	1742625/d	-48845544/d
$\hat{\alpha}_2$			-31131/d	818856/d	-9481000/d	271110375/d
$\hat{\alpha}_3$				-4543056/d	36589000/d	-983858000/d
$\hat{\alpha}_4$					-152635500/d	2850301500/d
$\hat{\alpha}_5$						-9531923400/d

Table 1: Coefficients in (2)

k	1	2	3	4	5	6
d	2	7	85	415	12019	13489
β_k	1	6/d	66/d	300/d	8220/d	8820/d
γ_k	-1/d	-2/d	-18/d	-72/d	-1800/d	-1800/d
α_0	-1	1/d	-4/d	9/d	-144/d	100/d
α_1		-8/d	27/d	-64/d	1125/d	-864/d
α_2			-108/d	216/d	-4000/d	3375/d
α_3				-576/d	9000/d	-8000/d
α_4					-18000/d	13500/d
α_5						-21600/d

Table 2: Coefficients in (3)

3.Accuracy and stability analysis

We now prove the following lemma regarding the order of accuracy of (2) used in the way described by stages (1)-(4).

Lemma 1. *Let*

(i) *formula (3) is of order $k+1$,*

(ii) *formula (2) is of order $k+3$,*

(iii) *the implicit algebra equations defining $\bar{y}_{n+k}$ and $\bar{y}_{n+k+1}$ are solved using an iteration scheme iterated to convergence,*
then scheme (2) has order $k+2$.

Proof: We prove this lemma on the assumption that y_{n+j} is exact for $0 \leq j \leq k-1$, i.e., $y_{n+j} \equiv y(x_{n+j})$. Now $\bar{y}_{n+k}$ satisfies the equation

$$y_{n+k} - h\beta_k f_{n+k} - h^2\gamma_k g_{n+k} = \sum_{j=0}^{k-1} \alpha_j y_{n+j}.$$

The analytic solution $y(x)$ of the original differential equation satisfies

$$y(x_{n+k}) - h\beta_k f(x_{n+k}, y(x_{n+k})) - h^2\gamma_k \frac{df}{dx}(x_{n+k}, y(x_{n+k})) = \sum_{j=0}^{k-1} \alpha_j y_{n+j} + T_{n+k},$$

where T_{n+k} is the local truncation error associated with (3) at the point x_{n+k}. Setting $\varepsilon_{n+k} \equiv y(x_{n+k}) - y_{n+k}$ and subtracting the above two relations, we have

$$\varepsilon_{n+k} - h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, \eta_1)\varepsilon_{n+k} - h^2\gamma_k \frac{d}{dx}\frac{\partial f}{\partial y}(x_{n+k}, \eta_2)\varepsilon_{n+k} = T_{n+k},$$

where $\eta_1, \eta_2 \in [y(x_{n+k}), y_{n+k}]$. From this relation, it follows immediately that

$$\varepsilon_{n+k} = \big[I - h\beta_k\big(\frac{\partial f}{\partial y}\big)_1 - h^2\gamma_k \frac{d}{dx}\big(\frac{\partial f}{\partial y}\big)_2\big]^{-1} T_{n+k},$$

where $(\frac{\partial f}{\partial y})_i$, $i = 1, 2$, are Jacobian matrices evaluated at approximate points. Similarly,

$$\varepsilon_{n+k+1} = \big[I - h\beta_k\big(\tfrac{\partial f}{\partial y}\big)_3 - h^2\gamma_k \tfrac{d}{dx}\big(\tfrac{\partial f}{\partial y}\big)_4\big]^{-1}$$

$$\times\Big\{ - \alpha_{k-1}\big[I - h\beta_k\big(\tfrac{\partial f}{\partial y}\big)_1 - h^2\gamma_k \tfrac{d}{dx}\big(\tfrac{\partial f}{\partial y}\big)_2\big]^{-1} T_{n+k} + T_{n+k+1}\Big\}.$$

Thus, we see immediately that $\varepsilon_{n+k+1} = \mathcal{O}(h^{k+2})$. Now the local truncation error associated with (2) is

$$\begin{aligned}
&\textstyle\sum_{j=0}^{k} \hat{\alpha}_j y(x_{n+j}) - h\big(\hat{\beta}_k f(x_{n+k}, y(x_{n+k})) + \hat{\beta}_{k+1} f(x_{n+1}, \bar{y}_{n+k+1})\big) \\
&\quad -h^2\big(\hat{\gamma}_k g(x_{n+k}, y(x_{n+k})) + \hat{\gamma}_{k+1} g(x_{n+k+1}, \bar{y}_{n+k+1})\big) \\
&\qquad = \textstyle\sum_{j=0}^{k} \hat{\alpha}_j y(x_{n+j}) - h\big(\hat{\beta}_k f(x_{n+k}, y(x_{n+k})) + \hat{\beta}_{k+1} f(x_{n+k+1}, y_{n+k+1})\big) \\
&\qquad -h^2\big(\hat{\gamma}_k g(x_{n+k}, y(x_{n+k})) + \hat{\gamma}_{k+1} g(x_{n+k+1}, y_{n+k+1})\big) \\
&\qquad +h\hat{\beta}_{k+1}\big(\tfrac{\partial f}{\partial y}\big)_5 \varepsilon_{n+k+1} + h^2\hat{\gamma}_{k+1}\tfrac{d}{dx}\big(\tfrac{\partial f}{\partial y}\big)_6 \varepsilon_{n+k+1}. \\
&\qquad = \hat{C}_{k+4} h^{k+4} y^{k+4}(x_n) + (1-\alpha_{k-1})\hat{\beta}_{k+1} C_{k+2} h^{k+3}\big(\tfrac{\partial f}{\partial y}\big)_5 y^{(k+2)}(x_n) + \mathcal{O}(h^{k+4}),
\end{aligned}$$

where $\hat{C}_{k+4}$ is the principal error constant associated with (2) and C_{k+2} is the principal error constant with (3). Values for the coefficients $\hat{C}_{k+4}, C_{k+2}$ have been given in Table 4. This completes the proof of the lemma. Lemma 1 shows the form taken by the local truncation errors of formulas (2) and, in particular, shows that they all have order $k+2$.

The new algorithm given in section 3 displays good stability characteristics. If we apply (3) and (6) to the test problem $y' = \lambda y$ for which $y'' = \lambda^2 y$, we get

$$\sum_{j=0}^{k} c_j(\bar{h}) y_{n+j} = 0, \tag{9}$$

where

$$c_k = 1 - \bar{h}\hat{\beta}_k - \bar{h}^2\hat{\gamma}_k,$$

$$c_j = \hat{\alpha}_j - \bar{h}\hat{\beta}_{k+1} d_j - \bar{h}^2\hat{\gamma}_{k+1} d_j, \quad j = 0, 1, \ldots, k-1,$$

$$d_0 = \tfrac{\alpha_0 \alpha_{k-1}}{A^2}, \qquad d_j = \tfrac{\alpha_j \alpha_{k-1}}{A^2} - \tfrac{\alpha_{j-1}}{A}, \qquad j = 1, 2, \ldots, k-1,$$

and

$$\bar{h} = \lambda h, \qquad A = 1 - \bar{h}\beta_k - \bar{h}^2\gamma_k.$$

Therefore, the corresponding characteristic equation of kth order difference equation of the method is

$$\pi(\xi, \bar{h}) = \sum_{j=0}^{k} c_j \xi^j = 0. \tag{10}$$

To see the zero-stability of this new method, one can easily show that by substituting $\bar{h} = \lambda h = 0$ in (10) the resulting characteristic polynomial satisfies the root condition and so the method is zero-stable [8].

To obtain the region of absolute stability we use the boundary locus method. Thus, the stability regions given are not exact but are those which have been found using a numerical search. By collecting coefficients of different powers of $\bar{h}$ in (10), we obtain

$$A_6\bar{h}^6 + A_5\bar{h}^5 + A_4\bar{h}^4 + A_3\bar{h}^3 + A_2\bar{h}^2 + A_1\bar{h} + A_0 = 0, \tag{11}$$

where $A_0, A_1, \ldots, A_6$ are functions of ξ. Inserting $\xi = e^{i\theta}$, (11) gives us six roots $\bar{h}(\theta), i = 1, 2, \ldots, 6$ which describe the stability domain. The corresponding (approximation) regions of A(α)-stability are given in Table 4.

Comparison of Tables 3 and 4 shows that regions of A(α)-stability for our new method is larger than those of the other mentioned methods. Also the new formulas of stepnumber k have higher order of accuracy and smaller error constants than mentioned above second derivative formulas and they are A-stable of order 8. If we consider values of $k > 6$ these formulas are highly A(α)-stable. However, when solving stiff systems of equations, it may not be desirable to have too many different order formulas, as quite often a change in order results in the coefficient matrix of the modified Newton scheme having to be re-evaluated an LU-decomposed and this is generally very expensive.

L-stability : From (8) we have

$$c_0 y_n + c_1 y_{n+1} + \cdots + c_k y_{n+k} = 0.$$

That is

$$y_{n+k} = -\frac{c_{k-1}}{c_k} y_{n+k-1} - \frac{c_{k-2}}{c_k} y_{n+k-2} - \cdots - \frac{c_1}{c_k} y_{n+1} - \frac{c_0}{c_k} y_n.$$

We observe that $y_{n+k} \to 0$ as $\bar{h} \to \infty$ that means the method is L-stable or L(α)-stable, according to whether the method is A-stable or A(α)-stable, respectively.

	Enright method		E2BDF1		E2BDF2		Ismail method		Hojjati method	
k	p	$\alpha(^\circ)$	p	$\alpha(^\circ)$	p	$\alpha(^\circ)$	p	$\alpha(^\circ)$	p	$\alpha_{max}(^\circ)$
1	3	90	4	90	4	90	2	90	3	90
2	4	90	5	90	5	90	3	90	4	90
3	5	87.88	6	90	6	90	4	90	5	90
4	6	82.03	7	90	7	89	5	89.9	6	90
5	7	73.10	8	90	8	87	6	87.3	7	89.8
6	8	59.95	9	89	9	83	7	84.2	8	88.3

Table 3: The A(α)-stability of some mentioned methods

k	p	$\alpha(^\circ)$	C_{k+2}	$\hat{C}_{k+4}$	k	p	$\alpha(^\circ)$	C_{k+2}	$\hat{C}_{k+4}$
1	3	90	0.166	0.430E-02	7	9	89.79	0.337E-02	0.173E-03
2	4	90	0.476E-01	0.712E-02	8	10	88.33	0.249E-02	0.705E-04
3	5	90	0.211E-01	-0.651E-02	9	11	85.57	0.191E-02	0.466E-04
4	6	90	0.115E-01	0.238E-03	10	12	81.44	0.149E-02	0.249E-04
5	7	90	0.713E-02	0.402E-03	11	13	75.93	0.120E-02	0.201E-04
6	8	90	0.476E-02	0.208E-03	12	14	68.71	0.816E-03	0.142E-04

Table 4: The A(α)-stability of new method

3.Numerical Results

In this section we present some numerical results to compare the performance of our new class of methods with that of other second derivative multistep methods. What we shall be attempting to do, is to show the superior performance of new method for a given fixed stepsize over some special methods for a small selection of examples. We do not claim that our numerical results demonstrate the superiority of our approach over any of the more second derivative approaches. However, we do feel that our results indicate that a properly implemented version of our algorithm should be useful for the numerical integration of stiff differential systems. We have programmed these methods in MATLAB.

Tol	F^n evals.	$\lvert$Error in $y_1\rvert$	$\lvert$Error in $y_2\rvert$	$\lvert$Error in $y_3\rvert$
1E-03	86	0.436E-07	0.824E-07	0.298E-10
1E-04	102	0.637E-06	0.451E-06	0.232E-10
1E-05	160	0.293E-06	0.169E-06	0.301E-11
1E-06	224	0.524E-07	0.564E-07	0.342E-12

Table 5: Numerical results for Example.1

Example 1. The first test problem which we consider is

$$\begin{array}{lll} y_1' = -0.013y_1 - 1000y_1y_3, & y_1(0) = 1, & \\ y_2' = -2500y_2y_3, & y_2(0) = 1, & 0 \leq t \leq 50, \\ y_3' = -0.013y_1 - 1000y_1y_3 - 2500y_2y_3, & y_3(0) = 0, & \end{array}$$

which is chemistry problem. This problem was integrated using the 1-step new SDMM scheme and the results obtained at the end point of the range of integration. At each step a fixed error tolerance, Tol, was imposed and a maximum of 6 Newton iterations was performed. If the relative accuracy of each iterate (in the sense of relations (8)) was not less than Tol the Jacobian matrix was re-evaluated and if convergence to the required degree of precision still did not occur within six iterations the steplength was halved. An initial step $h_0 = 10^{-3}$ was used. In Table 5 we list the results obtained for this integration. The true solution was obtained using an explicit fourth order Runge-Kutta method with a very small value of h. F^n denotes the number of times the right-hand side of (7). As you can see from Table 5, the high orders of accuracy can be achieved by decreasing Tol.

	x	The new Method
Error (y_1)	3	2.478147E-11
Error (y_2)	3	2.471093E-06
Error (y_1)	5	3.450271E-14
Error (y_2)	5	2.304573E-08
Error (y_1)	10	-3.456372E-18
Error (y_2)	10	-3.150734E-010

Table 6: The error results of Example 2

Example 2. Consider the non-linear system of differential equations:

$$y_1' = \lambda y_1 + y_2^2, \quad y_1(0) = -1/(\lambda + 2),$$

$$y_2' = -y_2, \qquad\quad y_2(0) = 1,$$

where $\lambda = 10000$. The exact solution is:

$$y_1(t) = -e^{-2x}/(\lambda + 2), \quad y_2(t) = e^{-x}$$

and the results are tabulated in Table 6 for $h = 0.0001$ at different values of x. We have obtained slightly better results than those of Ismail's method[7].

x	y_i	The new method
0.4	y_1	9.85172113863285E-1
	y_2	3.38639537890963E-5
	y_3	1.47940221854871E-2
40	y_1	7.15827068718903E-1
	y_2	9.18553476456739E-6
	y_3	2.84163745746394E-1
400	y_1	4.50518668477070E-1
	y_2	3.22290144170159E-6
	y_3	5.49478108624731E-1

Table 7: The results for Example 3

Example 3. Let us consider the following stiff problem

$$\begin{aligned} y_1' &= -0.04y_1 + 10^4 y_2 y_3, \\ y_2' &= 0.04y_1 - 10^4 y_2 y_3 - 3\times 10^7 {y_2}^2, \\ y_3' &= 3\times 10^7 {y_2}^2, \end{aligned}$$

with initial value $y(0) = (1, 0, 0)^T$. This is a chemistry problem suggested by Robertson. The results of the numerical integration at $x = 0.4, 40$ and 400 are presented in Table 7 solving with the method of order four and fixed stepsize $h = 0.001$.

x	y_i	Exact solution	Error in new method	Error in Ismail method	Error in SDBDF
2.0	y_1	-0.3616933169289E-5	0.52E-13	0.82E-10	0.31E-08
	y_2	0.9815029948230	0.19E-08	0.61E-05	0.18E-05
	y_3	01.018493388244	0.63E-08	0.57E-05	0.57E-05

Table 8: Numerical results for Example 4

Example 4. As our fourth numerical experiment, we consider the following stiff problem arose from a chemistry problem

$$\begin{aligned} y_1' &= -0.013y_2 - 1000y_1y_2 - 2500y_1y_3, \\ y_2' &= -0.013y_2 - 1000y_1y_2, \\ y_3' &= -2500y_1y_3, \end{aligned}$$

with initial value $y(0) = (0, 1, 1)^T$. We have solved this problem at $x = 2.0$ and compared the results with those of Ismail's method[6] and SDBDF[4]. A stepsize $h = 0.001$ has been used here and the order of method is four. One can also use the smaller stepsize to get significantly more accurate than this results. For the numerical results, see Table 8.

Example 5. As our fifth example, consider the Van der Pol's equation

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu^2((1 - y_1^2)y_2 - y_1), \end{aligned}$$

with initial value $y(0) = (2, 0)^T$. In Table 9 we have shown the numerical results by choosing $\mu = 500$. It should also be noted that a stepsize $h = 0.001$ has been used here and the order of method is five.

x	y_i	The new method
1	y_1	-1.865095092034
	y_2	0.7524845332331
5	y_1	1.8985234562376
	y_2	-0.7289532569825
10	y_1	1.7865365203279
	y_2	-0.8156276589331
20	y_1	1.5075643289233
	y_2	-1.1911230034538

Table 9: The results for Example 5

Example 6. Finally we present a numerical example which demonstrates the superior stability properties of our formulas. We consider the following stiff ODEs

$$\begin{aligned} y_1' &= -\alpha y_1 - \beta y_2 + (\alpha + \beta - 1)e^{-x}, \\ y_2' &= \beta y_1 - \alpha y_2 - (\alpha - \beta - 1)e^{-x}, \end{aligned}$$

with initial value $y(0) = (1,1)^T$. In order to make this system homogeneous, we introduce an additional variable y_3 such that

$$y_3' = 1, y_3(0) = 0.$$

The eigenvalues of the Jacobian associated with the resulting system are $-\alpha \pm i\beta, 0$ and the required solution is

$$y_1(x) = y_2(x) = e^{-x}.$$

In Table 10 we give the results obtained for the integration of this problem for the case $\alpha = 1$, $\beta = 30$ with $k = 5$. The value of h chosen was 0.09, which is such that the formula of E2BD-Class 2 reported by Cash[1] with $k = 5$ is unstable for this problem. Also comparison with the formula of Class 1, the new formula gives more accurate results for large t.

x	y_i	Error in new method	Error in E2BD1	Error in E2BD2
4.5	y_1	0.3E-11	<0.1E-10	<0.1E-10
	y_2	0.3E-11	<0.1E-10	<0.1E-10
9	y_1	0.3E-14	<0.1E-12	<0.1E-12
	y_2	0.3E-14	<0.1E-12	<0.1E-12
13.5	y_1	0.7E-16	<0.1E-15	0.1E-11
	y_2	0.6E-16	<0.1E-15	0.1E-11
18	y_1	0.1E-19	<0.1E-17	0.1E-11
	y_2	0.2E-19	<0.1E-17	0.1E-11

Table 10: Numerical results of Example 6, for the case $\alpha = 1$, $\beta = 30$

4. Discussion

We have derived a class of methods that, as it has been shown in section 4, has extensive region of stability and in particular is A-stable up to order 8. This Property, let us to apply the new method for numerical solution of stiff systems of ODEs with high accuracy. We do not claim that our numerical results demonstrate the superiority of our approach over any of the more conventional approaches. However, we do feel that our results indicate that a properly implemented version of our algorithm should be useful for the numerical integration of stiff differential systems.

References

[1] Cash J.R., *Second derivative extended backeward differentation formula for the numerical integration of stiff systems*, SIAM J. Numer. Anal., 18 (1981) 21-36.

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

[3] Enright W. H., *Second derivative multistep methods for stiff ordinary differential equation*, SIAM J. Numer.Anal., 11 (1974) 321-331.

[4] Hairer E. and Wanner G., *Solving ordinary differential equation II: Stiff and Differential-Algebraic problems*, Springer, Berlin, (1996).

[5] Hojjati G., Rahimi Ardabili M.Y. and Hosseini S.M., *New second derivative multistep methods for stiff system*, Applied Mathematical Modelling, 30 (2006) 466-476.

[6] Ismail G. and Ibrahim I., *New efficient second derivative multistep methods for stiff systems*, Applied Mathematical Modelling, 23 (1999) 279-288.

[7] Gamal A.F. Ismail. Iman H. Ibrahim, *A new higher order effective P-C methods for stiff systems*, Mathematics and Computers in Simulation 47 (1998) 541-552.

[8] Lambert J. D., *Computational methods in ordinary differential equation*, John Wiley Sons, (1972).

[9] Mehdizadeh khalsaraei M., Rahimi Ardabili M.Y. and Hojjati G., *The New class of super-implicit second derivative multistep methods for stiff systems*, Journal of Applied Functional Analysis, 4(3) (2009) 492-500.

[10] Y. Dai, Z. Wang., D. Wu. *A four-step trigonometric fitted p-stable Oberchkoff method for periodic initial-value problems*, J. Comput. Appl. Math. 187(2006) 192-201.

Mohammad Mehdizadeh Khalsaraei
Department of Mathematics
University of Maragheh
Maragheh, Iran
email: *Muhammad.mehdizadeh@gmail.com*

Nazli Nasehi Oskuei
Department of Mathematics
Tabriz Branch, Islamic Azad University,
Tabriz, Iran
email: *Nazli.Nasehi@gmail.com*

Gholamreza Hojjati
Department of Mathematics
University of Tabriz
Tabriz, Iran
email: *ghojjati@tabrizu.ac.ir*