Вычислительная математика

Computational mathematics

УДК 519.62

СТАБИЛИЗИРОВАННЫЕ ЯВНЫЕ МЕТОДЫ ТИПА АДАМСА ВЫСОКИХ ПОРЯДКОВ С ДЕМПФИРОВАНИЕМ

А. В. МОЙСА[1)]***, Б. В. ФАЛЕЙЧИК***[1)]***, В. И. РЕПНИКОВ***[1)]

[1)]*Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь*

Продолжается исследование явных методов типа Адамса с расширенным интервалом устойчивости, впервые представленных в предыдущей статье авторов в издании «Журнал Белорусского государственного университета. Математика. Информатика» (2021, № 2). Такие методы требуют только одного вычисления f на каждом шаге, но при этом имеют гораздо более длинные интервалы устойчивости, чем классические аналоги. Целью работы является построение демпфированных модификаций методов с расширенным интервалом устойчивости второго порядка и выше, а также тестирование их пригодности для решения жестких систем обыкновенных дифференциальных уравнений. Для расширения области устойчивости вблизи действительной оси предлагается общая процедура оптимизации, основанная на поиске по сетке с последовательным увеличением демпфирующего параметра. Строятся ряд методов второго, третьего и четвертого порядков, описывается реализация адаптивного выбора шага интегрирования и приводятся результаты сравнительных численных экспериментов.

Ключевые слова: жесткие системы; линейные многошаговые методы; методы типа Адамса; явные методы.

Образец цитирования:
Мойса АВ, Фалейчик БВ, Репников ВИ. Стабилизированные явные методы типа Адамса высоких порядков с демпфированием. *Журнал Белорусского государственного университета. Математика. Информатика.* 2023;1:64–75 (на англ.).
https://doi.org/10.33581/2520-6508-2023-1-64-75

For citation:
Moisa AV, Faleichik BV, Repnikov VI. Higher order stabilised explicit Adams-type methods with damping. *Journal of the Belarusian State University. Mathematics and Informatics.* 2023;1:64–75.
https://doi.org/10.33581/2520-6508-2023-1-64-75

Авторы:
Андрей Владимирович Мойса – аспирант кафедры вычислительной математики факультета прикладной математики и информатики. Научный руководитель – Б. В. Фалейчик.
Борис Викторович Фалейчик – кандидат физико-математических наук, доцент; доцент кафедры вычислительной математики факультета прикладной математики и информатики.
Василий Иванович Репников – кандидат физико-математических наук, доцент; заведующий кафедрой вычислительной математики факультета прикладной математики и информатики.

Authors:
Andrew V. Moisa, postgraduate student at the department of computational mathematics, faculty of applied mathematics and computer science.
moysa@bsu.by
Boris V. Faleichik, PhD (physics and mathematics), docent; associate professor at the department of computational mathematics, faculty of applied mathematics and computer science.
faleichik@bsu.by
Vasily I. Repnikov, PhD (physics and mathematics), docent; head of the department of computational mathematics, faculty of applied mathematics and computer science.
repnikov@bsu.by

HIGHER ORDER STABILISED EXPLICIT ADAMS-TYPE METHODS WITH DAMPING

***A. V. MOISA*[a], *B. V. FALEICHIK*[a], *V. I. REPNIKOV*[a]**

[a]*Belarusian State University, 4 Niezaliežnasci Avenue, Minsk 220030, Belarus*
Corresponding author: A. V. Moisa (moysa@bsu.by)

In this paper we continue the study of explicit Adams-type methods with an extended stability interval represented for the first time in the previous article of the authors in «Journal of the Belarusian State University. Mathematics and Informatics» (2021, No. 2). Such methods require only one calculation of f at each step, but at the same time, they have much longer stability intervals than their classical counterparts. The aim of this work is to construct damped modifications of the methods with an extended stability interval of second order and higher and to test their ability to solve stiff systems of ordinary differential equations. In order to extend the stability regions along the real axis, we propose a general optimisation procedure based on grid search with a progressive increase in the damping parameter. We construct several methods of second, third and fourth orders, describe the realisation of the adaptive choice of the integration step, and represent the results of the comparative numerical experiments.

Keywords: stiff systems; linear multistep methods; Adams-type methods; explicit methods.

Introduction

Let us consider a system of ordinary differential equations

$$y' = f(t, y),\ y(t_0) = y_0,\ y : \mathbb{R} \to \mathbb{R}^n,\ f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n. \tag{1}$$

Methods of the form

$$y_{m+k} = y_{m+k-1} + \tau\left(\beta_0 f_m + \ldots + \beta_{k-1} f_{m+k-1}\right) \tag{2}$$

with an increased length of the stability interval were considered in work [1]. Let p be the order of the method (2). The root locus curves [2, p. 241] of these methods have the form

$$\mathcal{C} = \left\{\mu_\beta\left(e^{i\varphi}\right) \middle| \varphi \in [0, 2\pi)\right\},\ \mu_\beta(\zeta) = \frac{\rho(\zeta)}{\sigma(\zeta)} = \frac{\zeta^k - \zeta^{k-1}}{\sum\limits_{j=0}^{k-1} \beta_j \zeta^j}. \tag{3}$$

These curves for a sufficiently large k touch the real axis at several points (fig. 1). Damping of the method for the numerical solution of the initial value problem (1) [2, p. 32–33] is a qualitative modification of the method in order to increase the distance between the boundary of the stability region and the real axis (see also [3; 4]). This is necessary for the stability of the method in the case of slight deviations of the eigenvalues of the Jacobian matrix from the real axis. When constructing a damped method, we need to introduce a minimal change to the stability interval length while keeping an acceptable distance of the root locus curve from the real axis.

The paper is organised as follows. In section «Damped methods construction» we describe the approach which was used to construct the damped stabilised Adams-type methods of higher orders. The implementation details and the results of numerical experiments are presented in section «Implementation».

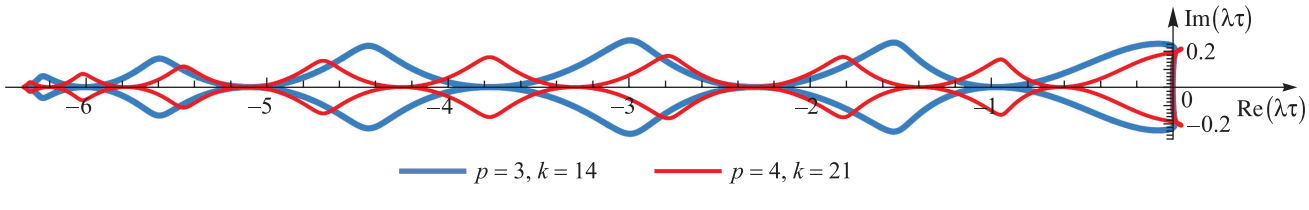

Fig. 1. Root locus curves (3) of the methods with an extended stability interval

Damped methods construction

Problem statement. The optimisation problem for the unknown coefficients of the damped method can be stated as

$$\begin{gathered}\beta^* = \arg\min_{\beta \in \mathcal{F} \cap \mathcal{P}_p} \mu_\beta(-1),\\ \mathcal{F} = \left\{\beta \in \mathbb{R}^k \,\middle|\, \operatorname{Im}\mu_\beta\left(e^{i\varphi}\right) \geq \varepsilon \ \ \forall\varphi \in [\gamma,\, \pi - \gamma]\right\},\\ \mathcal{P}_p = \left\{\beta \in \mathbb{R}^k \,\middle|\, \sum_{j=0}^{k-1} \beta_j = 1,\ \sum_{j=0}^{k-1} (1-k+j)^{q-1}\beta_j = \frac{1}{q},\ q = \overline{2,\, p}\right\},\end{gathered} \tag{4}$$

where ε is a parameter which determines a shift from the real axis, and γ is a small constant which eliminates inequalities in the conditions where $\sin\varphi$ is small. However, it is not possible to solve it in this form for a sufficiently large k, so let's try to get a simpler problem.

It was shown in work [1] that the imaginary part of root locus curve (3) of the method (2) has the form

$$\operatorname{Im}\mu_\beta\left(e^{i\varphi}\right) = Q(\varphi)\sin\varphi,\ Q(\varphi) = \frac{\sum\limits_{j=0}^{k-1} a_j \cos j\varphi}{\sum\limits_{j=0}^{k-1} \delta_j \cos j\varphi},$$

where

$$\begin{gathered}a_{k-1} = 2\beta_0,\ a_{k-j-1} = 2\beta_j - a_{k-j},\ a_0 = \beta_{k-1} - \frac{a_1}{2},\ j = \overline{1,\, k-2},\\ \delta_0 = \sum_{j=0}^{k-1} \beta_j^2,\ \delta_j = 2\sum_{l=0}^{j} \beta_l \beta_{j+1},\ j = \overline{1,\, k-1}.\end{gathered} \tag{5}$$

A way of damping of the first order methods was proposed in work [1]. It is based on the representation of $\hat{Q}(\varphi)$ of the damped method as

$$\hat{Q}(\varphi) = \frac{\sum\limits_{j=0}^{k-1} \hat{a}_j \cos j\varphi}{\sum\limits_{j=0}^{k-1} \hat{\delta}_j \cos j\varphi},\quad \hat{a}_j = C\left(a_j + \varepsilon\delta_j\right),$$

where C is a scaling constant to be determined from only one order condition. The coefficients $\hat{\beta}_j$ and $\hat{\delta}_j$ are related to $\hat{a}_j$ in the same way as β_j and δ_j relate to a_j in formula (5). Thus, $\hat{\beta}_j$ can be found from $\hat{a}_j$ as

$$\hat{\beta}_0 = \frac{\hat{a}_{k-1}}{2},\ \ \hat{\beta}_j = \frac{\hat{a}_{k-j-1} + \hat{a}_{k-j}}{2},\ \ \hat{\beta}_{k-1} = \hat{a}_0 + \frac{\hat{a}_1}{2},\ \ j = \overline{1,\, k-2}. \tag{6}$$

This approach is not applicable in the case when there are more than one order condition ($p > 1$), therefore to construct a higher order damped method we put

$$\hat{a}_j = C\left(a_j + \varepsilon\delta_j + \nu_j\right), \tag{7}$$

where ν_j are constants, $p - 1$ of which are determined from the order conditions, and the rest can be set in such a way that the stability interval of the obtained method is the largest possible, but at the same time the imaginary part of the root locus curve is greater than ε on the segment $[\gamma,\, \pi - \gamma]$. Let's write the corresponding optimisation problem as

$$\begin{gathered}C,\, \nu = \arg\min_{\substack{(C,\, \nu) \in \mathcal{F},\\ \hat{\beta} \in \mathcal{P}_p}} \mu_\beta(-1),\\ \mathcal{F} = \left\{C \in \mathbb{R},\, \nu \in \mathbb{R}^k \,\middle|\, \hat{Q}(\varphi)\sin\varphi \geq \varepsilon\ \forall\varphi \in [\gamma,\, \pi - \gamma]\right\}.\end{gathered} \tag{8}$$

Problem (8) is much easier to handle than problem (4) because, firstly, the functions from the admissible set directly depend on the sought coefficients, and, secondly, it turns out that to obtain an acceptable result, it is enough to immediately fix $\nu_q = \nu_{q+1} = \ldots = \nu_k = 0$, $p - 1 \leq q \leq k$, thus reducing the dimension of the problem. We chose $\varepsilon = 0.05$, $\gamma = 0.15$ in our implementation.

The damped methods construction algorithm. The algorithm below is based on a combination of a brute-force grid search with a successive increase of ε.

The required input values are:

• the coefficients $\{\beta_j\}$ of an original stabilised Adams-type method constructed in work [1];

• the search grid step h, the empirically defined range of values is $h \in \left[10^{-4}, 10^{-3}\right]$;

• the required root locus curve shift from the real axis $\varepsilon > 0$.

The output of the algorithm is the scalar variable *sc* and the vector variable *nu* which store the computed values of C and $\nu = \{\nu_j\}$ respectively. The sought coefficients of the damped method $\{\hat{\beta}_j\}$ are then derived from C and ν using formulas (7) and (6).

Algorithm: damping of the higher order stabilised explicit Adams-type method.

Require: $\beta_j,\ j = \overline{0,\ k-1},\ h > 0,\ \varepsilon > 0$.

Calculate $a_j,\ \delta_j,\ j = \overline{0,\ k-1}$, from β_j by (5)

$q \leftarrow \min\{2 + p,\ k - p\}$

$o \leftarrow \{o_j = 0,\ j = \overline{0,\ q-p}\}$

$nu \leftarrow \{nu_j = 0,\ j = \overline{0,\ k-1}\}$

$eps \leftarrow \dfrac{h}{5}$

while $eps \le \varepsilon$ **do**

    $\ell \leftarrow 0$

    $N \leftarrow [o_0 - h,\ o_0,\ o_0 + h] \times [o_1 - h,\ o_1,\ o_1 + h] \times \ldots \times [o_{q-p} - h,\ o_{q-p},\ o_{q-p} + h]$

    for every unique node n in N **do**

        $\{\nu_0, \ldots, \nu_{q-p}\} \leftarrow \{n_0, \ldots, n_{q-p}\}$

        $\{\nu_q, \ldots, \nu_{k-1}\} \leftarrow \{0, \ldots, 0\}$

        Express $\hat{a}_j \leftarrow C\left(a_j + eps \cdot \delta_j + \nu_j\right)$, see (7)

        Express $\hat{\beta}_j$ from $\hat{a}_j$ as in (6)

        Find $C,\ \nu_j,\ j = \overline{q-p+1,\ q-1}$ from the order conditions $\mathcal{P}_p$ (4)

        Calculate values of $\hat{\beta}_j$

        if $\operatorname{Im}\mu_{\hat{\beta}}\left(e^{i\varphi}\right) \ge eps\ \ \forall\varphi \in [\gamma,\ \pi - \gamma]$

        and $\operatorname{Im}\mu_{\hat{\beta}}\left(e^{i\varphi}\right) \ge 0\ \ \forall\varphi \in [\pi - \gamma,\ \pi]$

        and $\left|\mu_{\hat{\beta}}\left(e^{i\pi}\right)\right| > \ell$ **then**

            $sc \leftarrow C$

            $nu \leftarrow \nu$

            $\ell \leftarrow \left|\mu_{\hat{\beta}}\left(e^{i\pi}\right)\right|$

        else

            continue

        end if

    end for

    if $\ell > 0$ **then**

        $o \leftarrow nu$

        if $eps = \varepsilon$ **then**

            break

        else

            $eps \leftarrow \min\left\{eps + \dfrac{h}{5},\ \varepsilon\right\}$

        end if

    else

        $q \leftarrow q + 1$

        Append nu_{q-p+1} to vector o

    end if

end while

return *sc*, *nu*

The non-linear system of equations which needs to be solved in row 14 is transformed to a linear one by replacing $C \cdot \nu_j$ with x_{j-q+p}.

The inequalities in rows 16 and 17 can be checked for every local minimum of function $\mathrm{Im}\,\mu_{\hat{\beta}}\left(e^{i\varphi}\right)$, taking into account the specific shape of this function (fig. 2).

It is necessary to manually check the shape of the root locus curve of the damped method, because sometimes the algorithm can produce an unsatisfactory result (although it formally satisfies all the requirements). In order for the root locus curve to have a more regular shape, it is necessary to increase the parameter q in row 2 and repeat the algorithm.

Examples of the root locus curves of the damped method for $p = 4$, $k = 21$ and different values of ε are presented in fig. 3, graphs of $Q(\varphi)$ and $\hat{Q}(\varphi)$ of these methods are given in fig. 2. Stability regions of the second order methods with an extended stability interval and their damped versions (ε = 0.05) for $k = 4, \ldots, 13$ are presented in fig. 4.

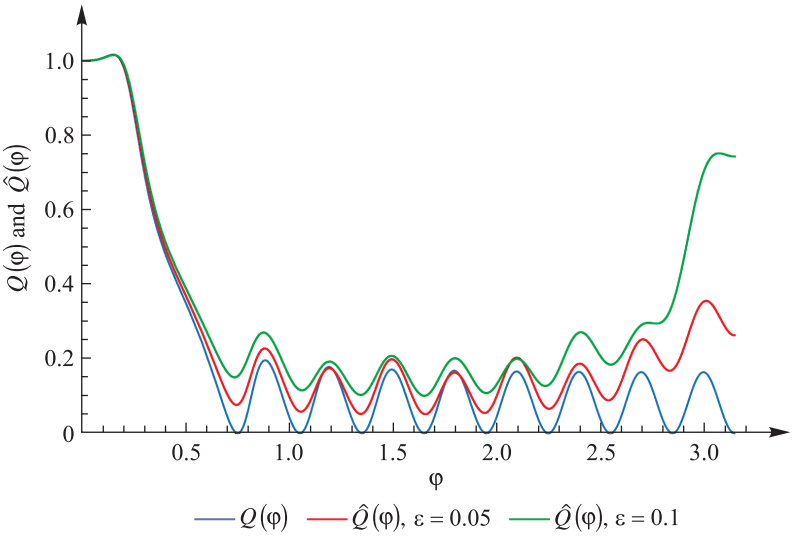

Fig. 2. Graphs of $Q(\varphi)$ and $\hat{Q}(\varphi)$ of the damped methods with $p = 4$, $k = 21$

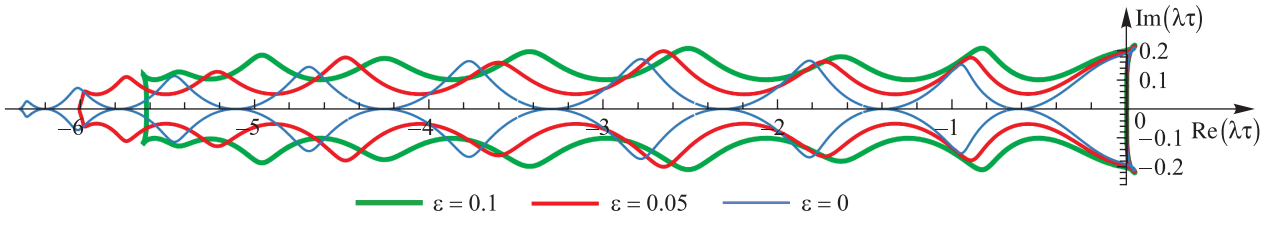

Fig. 3. Root locus curves (3) of the damped methods with $p = 4$, $k = 21$

Implementation

Implementation of methods (2) with a constant step size is straightforward. To implement a variable step size, it is necessary to solve the following main tasks:

1) error estimation;
2) increasing the grid step size;
3) decreasing the grid step size.

Error estimation. We use two methods of orders p (main method) and $p - 1$ (assistant method) to estimate the error. The values y_{m+1} calculated by these methods in formula (2) we denote as $y_{m+k}^{[p]}$ and $y_{m+k}^{[p-1]}$ respectively. Then the absolute error estimate of the $y_{m+k}^{[p]}$ can be found from the well-known formula

$$aerr = \left\| y_{m+k}^{[p]} - y_{m+k}^{[p-1]} \right\|_{\infty}. \tag{9}$$

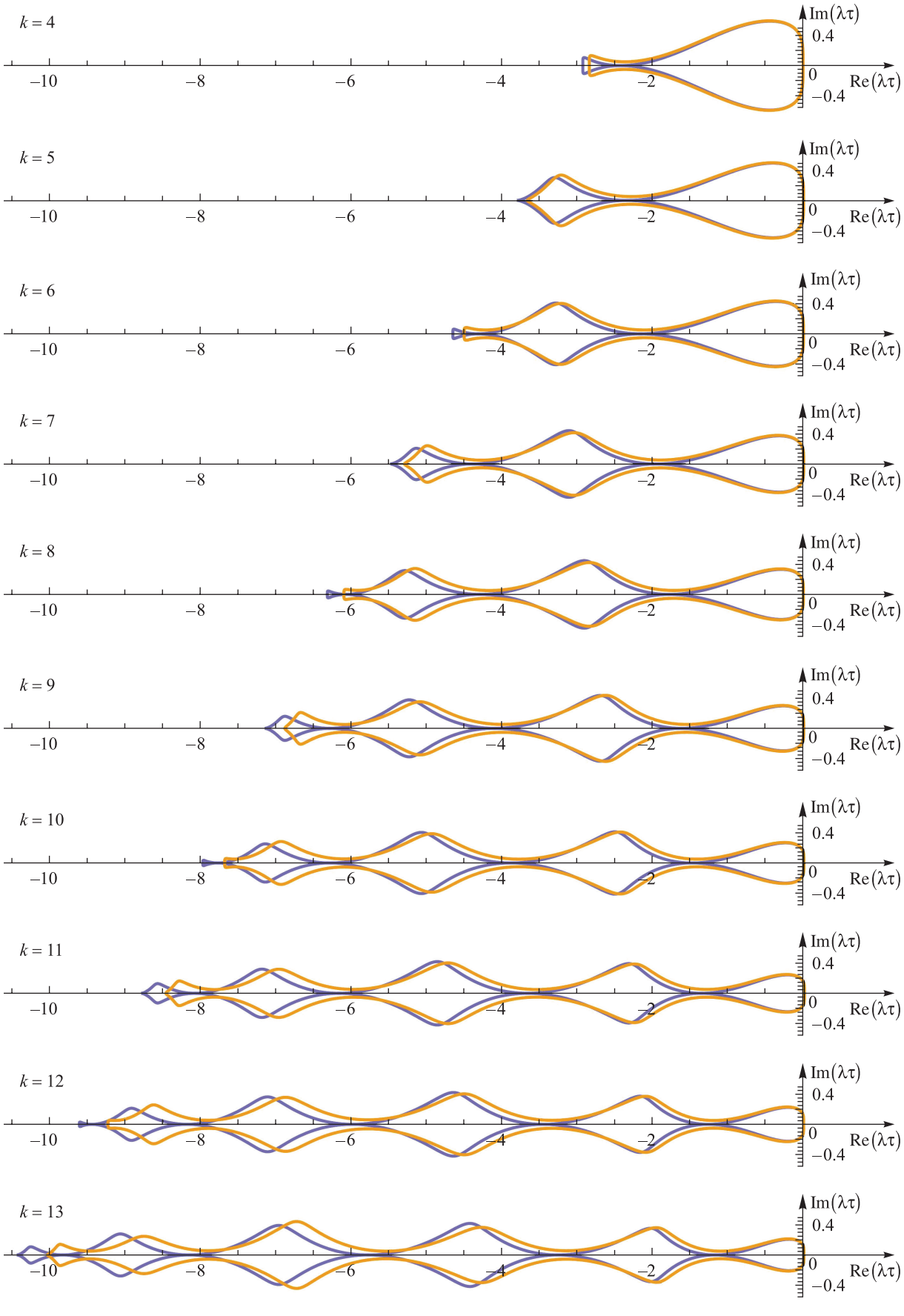

Fig. 4. Stability regions of the second order methods with an extended stability interval and their damped versions (ε = 0.05) for $k = 4, \ldots, 13$

The relative error estimation is defined as

$$rerr = \|r\|_\infty, \; r_j = \frac{\left\{y_{m+k}^{[p]}\right\}_j - \left\{y_{m+k}^{[p-1]}\right\}_j}{\left|\left\{y_{m+k}^{[p]}\right\}_j\right| + atol}, \tag{10}$$

where *atol* is the value of the required absolute error. It is added to the absolute value of the vector components to avoid close to zero values in the denominator.

If the inequalities

$$aerr \le atol, \; rerr \le rtol, \tag{11}$$

where *rtol* is the required relative error, are satisfied, then the value $y_{m+k}^{[p]}$ is considered as accurate enough.

Formulas (9)–(11) can be easily generalised for the case when *atol* and *rtol* are vector values. To do this, in expressions (9) and (10) it is necessary to replace the norm by the vector of the absolute values of the components, and the inequalities (11) should be generalised to the case of the vector values.

We mainly used the damped version of the method with $p = 4$, $k = 21$ as the main method in our implementation (see fig. 2 and 3). This method is quite simple to obtain and damp and it has 20 times larger stability interval than the classical fourth order Adams method. If it is necessary to get a larger stability interval, it might be better to reduce the order p instead of increasing k. The coefficients of this method are presented in table 1. The error constants of some fourth order methods are given in table 2.

Initially, it seems natural to choose an assistant method with approximately the same stability interval as the main method, for example, $p = 3$, $k = 14$ (see fig. 1). But in practice, better results were obtained using the classical explicit Adams method of order $p - 1$ as an assistant one. This can be explained by the fact that the result of the assistant method is not used in the next step, so its instability does not affect the process too much.

Table 1

Coefficients of method with an extended stability interval and its damped version with $p = 4$, $k = 21$

β		C; ν
Initial method	Damped method	
–0.014 543 302 409 352 176	–0.012 505 757 070 276 544	
–0.037 372 767 456 907 95	–0.032 789 411 451 952 875	
–0.043 406 196 086 467 105	–0.039 488 125 649 616 054	0.835 1;
–0.027 486 149 404 601 503	–0.027 107 568 402 238 53	0.038 4
0.008 252 769 527 671 221	0.003 642 176 786 235 481 7	0.050 7
0.054 537 413 741 972 81	0.045 478 507 054 112 57	0.044
0.097 209 195 621 928 01	0.086 121 595 255 923 64	0.015 4
0.121 058 627 899 381 7	0.111 500 997 408 773 63	0.007 5
0.114 285 854 014 006 83	0.109 820 665 237 236 78	–0.004 1
0.072 422 109 196 496 3	0.074 196 216 563 922 67	–0.001 6
0.000 623 683 164 529 862 5	0.009 440 996 312 261 642	–0.001 8
–0.086 396 278 848 252 68	–0.070 033 943 854 504 19	0.002 1
–0.166 978 077 073 466 46	–0.147 429 638 898 536 27	0.001 5
–0.217 219 080 029 73	–0.202 674 306 764 101 2	–0.001 8
–0.216 768 383 496 507 5	–0.212 396 132 756 734 38	–0.002 9
–0.154 368 955 636 348 4	–0.163 055 764 045 559 94	0
–0.031 766 972 270 760 77	–0.049 724 157 172 647 37	⋮
0.135 099 815 930 591 2	0.114 121 239 098 021 19	0
0.318 153 574 581 982 3	0.305 707 490 985 957	
0.482 259 056 075 009 85	0.483 881 120 483 082 2	
0.592 404 062 958 824 4	0.613 293 800 880 640 2	

Table 2

Error constants of the fourth order methods with an extended stability interval and their damped versions

k	Initial method	Damped method
21	94.211 3	88.203 6
17	41.462 4	39.133 3
13	14.899 8	14.196
9	3.878 8	3.730 3

Increasing the grid step size. In the cases of sufficiently small errors *aerr* and *rerr* it makes sense to increase the step size τ of method (2). In contrast to one-step methods [5, chapter II, section II.4], changing the step size τ by an arbitrary factor causes serious computational difficulties (see also [3; 4; 6]). To simplify the process of updating the grid, we introduce the constant

$$irate = \frac{\tau_{new}}{\tau},$$

where τ_{new} is the step size after the increasing (fig. 5).

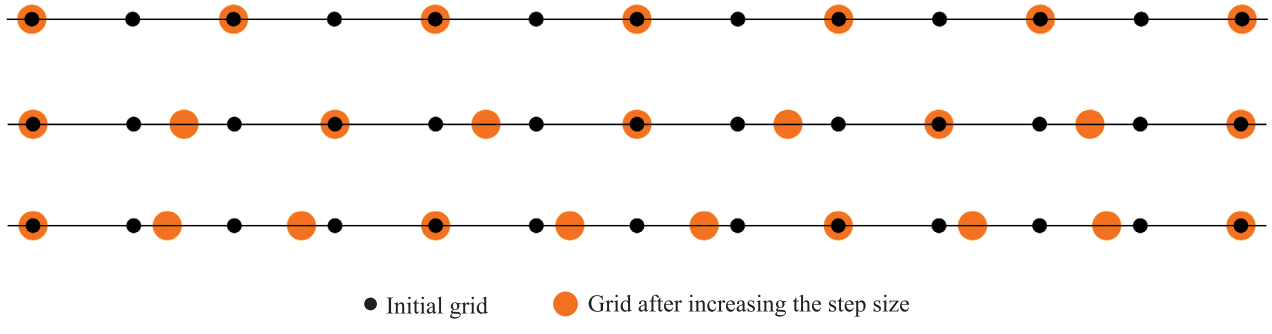

Fig. 5. Increasing the grid step size in the cases $irate = 2$ (top), $irate = \frac{3}{2}$ (middle) and $irate = \frac{4}{3}$ (bottom)

After increasing the grid step size, the density of nodes decreases, but their number must still be at least k to calculate the next value of the solution by the formula (2). We will store a grid of nodes of size

$$gsize = \lceil irate(k-1)+1 \rceil$$

to avoid extrapolation. It allows to calculate k nodes of the increased grid by the interpolation polynomial and continue solving using the formula (2) with the new step size τ_{new}.

We used the Hermite interpolation polynomial in our implementation. For methods with $p \le 3$, the well-known formula

$$y_{l-\frac{3}{2}} = \frac{y_{l-2}+y_{l-1}}{2} + \frac{\tau}{8}\left(f_{l-2}+f_{l-1}\right),\ l = m+k-3j,\ j = 0, 1, \ldots, \left\lfloor \frac{k}{2}-1 \right\rfloor,$$

is enough to find the values of the missing nodes in the case $irate = \frac{3}{2}$. For higher order methods we used the more accurate formula

$$y_{l-\frac{3}{2}} = \frac{13y_{l-3}+243y_{l-2}+243y_{l-1}+13y_l}{512} + \frac{\tau}{512}\left(3f_{l-3}+81f_{l-2}-81f_{l-1}-3f_l\right),$$

$$l = m+k-3j,\ j = 0, 1, \ldots, \left\lfloor \frac{k}{2}-1 \right\rfloor.$$

Naturally, we have to completely fill the grid with solution values before the next increment. It is reasonable to choose a method with $irate \cdot (k-1) \in \mathbb{N}$ to reduce the computational costs when grid values with an increased step size are calculated.

Suppose that the value *irate* is chosen. If at some moment the number of stored solution values is less than *gsize* (for example, at the very beginning or immediately after increasing the grid step size) then it is not possible to increase the grid step size. If the entire grid is completely filled, in addition to inequalities (11) the condition

$$aerr \le \omega \frac{atol}{irate^p}, \quad rtol \le \omega \frac{rtol}{irate^p}, \quad 0 < \omega < 1,$$

should also be checked. If it is satisfied, we calculate the missing grid nodes with an increased step size τ_{new} and store them into a separate array. Then we find the next value of the solution using formula (2) and check its accuracy using inequalities (11). If they are satisfied, we change the old grid with step size τ to the new one with step size τ_{new} and continue calculating with this step. If inequalities (11) are not satisfied, we discard the solution value obtained with step size τ_{new} and continue applying the method with step size τ.

In our implementation we take $irate = \frac{3}{2}$, $\omega = 0.9$.

This implementation already works quite well, but it can be further improved. Consider the situation when the errors $aerr_{m+k}$ and $rerr_{m+k}$ of $y_{m+k}^{[p-1]}$ still satisfy conditions (11) but at least one of them have become larger than $aerr_{m+k-1}$ and $rerr_{m+k-1}$ respectively. This could happen for three reasons:

1) the error of integration of the function f has increased;
2) one or more eigenvalues of the Jacobian matrix went out of the stability region;
3) there were roundoff errors while calculating *aerr* and *rerr*.

In the first two cases, we should avoid the step increase during the next *iprhb* steps. In practice, it turned out that in the case of double precision numbers, a simple check

$$aerr_{m+k} - aerr_{m+k-1} \le 3 \cdot 10^{-15}, \quad rerr_{m+k} - rerr_{m+k-1} \le 3 \cdot 10^{-15} \tag{12}$$

is sufficient to handle the third case. If at least one inequality in formula (12) is not satisfied, we disallow the step size increase for the next $iprhb = 13$ steps.

Decreasing the grid step size. If at some moment at least one inequality in formula (11) is not satisfied, then the grid step size must be reduced. As in the case of increase, we introduce the constant

$$drate = \frac{\tau}{\tau_{new}},$$

which is presented in fig. 6.

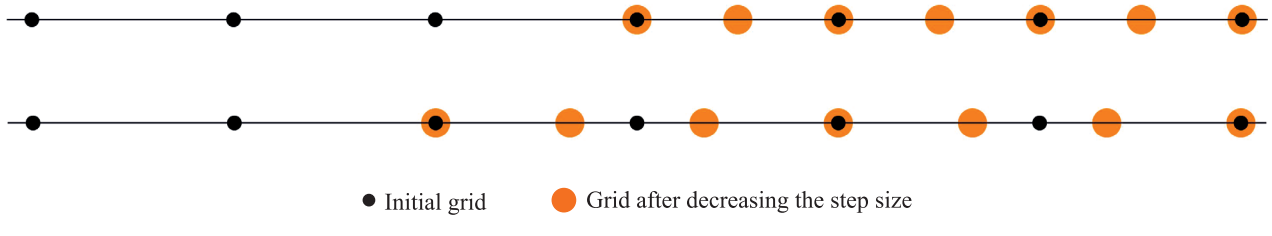

Fig. 6. Decreasing the grid step size in the cases $drate = 2$ (top), $drate = \frac{3}{2}$ (bottom)

In our implementation we take $drate = \frac{3}{2}$.

The values at the missing nodes can be found using the formula

$$y_{l-\frac{2}{3}} = \frac{5}{81} y_{l-2} + \frac{64}{81} y_{l-1} + \frac{4}{27} y_l + \frac{\tau}{243}\left(4 f_{l-2} + 64 f_{l-1} - 8 f_l\right),$$

$$y_{l-\frac{4}{3}} = \frac{4}{27} y_{l-2} + \frac{64}{81} y_{l-1} + \frac{5}{81} y_l + \frac{\tau}{243}\left(8 f_{l-2} - 64 f_{l-1} - 4 f_l\right),$$

$$l = m + k - 2j, \quad j = 0, 1, \ldots, \left\lfloor \frac{k}{2} \right\rfloor.$$

Implementation details. The proposed damped explicit multistep methods were implemented in C. The main features of the implementation have already been described, so here we add only a few details.

The first $k - 1$ grid values were found using the one-step method RADAU5 (see the Internet page *www.unige.ch/~hairer/software.html*). Grid values y_m and f_m of length *gsize* are stored in a circular queue. Pointers

to the oldest element in the queue (`begin`), to the element from which the calculation begins according to formula (2) (`front`), and to the last added element (`back`) are stored separately. There are always exactly k elements between the `front` (inclusive) and `back` (inclusive) elements. Between the `begin` (inclusive) and `back` (inclusive) elements there can be from k (at the very beginning or immediately after increasing the step size) to *gsize* elements (the grid is completely filled). Elements are never completely removed from this queue, currently unnecessary elements are simply moved to the end of the queue (after the `back` element). Elements from the outside are never added to this queue, the usage of free (located between `back` and `begin`) elements is always enough. The projects are compiled with Intel Fortran Compiler Classic 2021.4.0 and Intel C++ Compiler 19.2.

Numerical experiments

All methods implementations to compare with our method were taken from the Internet page *www.unige.ch/~hairer/software.html*. The damped method with $p = 4$, $k = 21$ will be denoted as SA4-21. The parameters *rtol* and *atol* are equal in all cases. The following tables will provide the statistical data about the performance of the considered methods. We use the following notation:

- $rerr_{fin}$ is the final relative error compared to a more accurate model solution;
- $aerr_{fin}$ is the final absolute error compared to a more accurate model solution;
- *fcn* is the number of function evaluations;
- *step* is the number of the computed steps;
- *accpt* is the number of the accepted steps;
- *rejct* is the number of the rejected steps;
- *time* is the elapsed time (ms).

DOPRI5 is an explicit one-step Dormand – Prince method of fifth order [7]. Initially this method was not intended for solving stiff problems. But in work [2, chapter IV, section IV.10] it was shown that it can be successfully applied to middly-stiff systems.

ROCK4 is an explicit one-step fourth order Chebyshev method [4]. This method was created for stiff problems which possess a Jacobian matrix with (possibly large) eigenvalues close to the real negative axis.

RADAU5 is an implicit (unlike all the others) one-step method RadauIIA [2, p. 74]. It is impossible to directly compare the statistics of explicit methods with it, it is listed as a classical method for solving stiff systems.

HIRES. This is a classical midly stiff test system of dimension 8 describing a chemical reaction [8]. The interval of integration is $[0, 321.812\,2]$ (table 3) and $[0, 421.812\,2]$ (table 4).

Table 3

Numerical experiment results for the HIRES problem, $t_{max} = 321.812\,2$

Method	*atol*	$rerr_{fin}$	$aerr_{fin}$	*fcn*	*step*	*accpt*	*rejct*	*time*
SA4-21	$1 \cdot 10^{-6}$	$7.16 \cdot 10^{-6}$	$4.47 \cdot 10^{-8}$	13 766	12 774	12 745	29	3
	$1 \cdot 10^{-8}$	$7.03 \cdot 10^{-8}$	$4.38 \cdot 10^{-10}$	19 080	17 132	17 069	63	4
	$1 \cdot 10^{-10}$	$2.51 \cdot 10^{-9}$	$1.57 \cdot 10^{-11}$	22 517	20 811	20 756	55	4
	$1 \cdot 10^{-12}$	$2.46 \cdot 10^{-10}$	$1.54 \cdot 10^{-12}$	41 523	40 359	40 302	57	8
DOPRI5	$1 \cdot 10^{-6}$	$2.37 \cdot 10^{-3}$	$1.48 \cdot 10^{-7}$	62 504	10 417	10 416	1	2
	$1 \cdot 10^{-8}$	$8.17 \cdot 10^{-7}$	$2.33 \cdot 10^{-9}$	62 840	10 473	10 473	0	2
	$1 \cdot 10^{-10}$	$1.02 \cdot 10^{-8}$	$2.9 \cdot 10^{-11}$	65 366	10 894	10 894	0	2
	$1 \cdot 10^{-12}$	$1.07 \cdot 10^{-10}$	$3.05 \cdot 10^{-13}$	79 340	13 233	13 233	0	3
ROCK4	$1 \cdot 10^{-6}$	$3.23 \cdot 10^{-3}$	$2.01 \cdot 10^{-7}$	11 628	873	812	61	1
	$1 \cdot 10^{-8}$	$9.24 \cdot 10^{-6}$	$2.63 \cdot 10^{-8}$	19 931	2071	1928	143	1
	$1 \cdot 10^{-10}$	$7.82 \cdot 10^{-9}$	$3.09 \cdot 10^{-11}$	35 410	4842	4632	210	1
	$1 \cdot 10^{-12}$	$3.28 \cdot 10^{-11}$	$1.59 \cdot 10^{-13}$	69 873	11 253	11 102	151	2
RADAU5	$1 \cdot 10^{-6}$	$1.01 \cdot 10^{-3}$	$1.48 \cdot 10^{-7}$	491	60	53	7	<1
	$1 \cdot 10^{-8}$	$1.10 \cdot 10^{-5}$	$6.89 \cdot 10^{-8}$	820	100	97	3	1
	$1 \cdot 10^{-10}$	$5.89 \cdot 10^{-8}$	$3.68 \cdot 10^{-10}$	1655	197	197	0	1
	$1 \cdot 10^{-12}$	$4.88 \cdot 10^{-9}$	$1.8 \cdot 10^{-11}$	3291	414	414	0	1

Table 4

Numerical experiment results for the HIRES problem, $t_{max} = 421.8122$

Method	*atol*	$rerr_{fin}$	$aerr_{fin}$	*fcn*	*step*	*accpt*	*rejct*	*time*
SA4-21	$1 \cdot 10^{-6}$	$1.08 \cdot 10^{-10}$	$1.52 \cdot 10^{-13}$	14 290	13 288	13 259	29	3
	$1 \cdot 10^{-8}$	$1.28 \cdot 10^{-9}$	$1.64 \cdot 10^{-12}$	19 962	17 916	17 851	65	4
	$1 \cdot 10^{-10}$	$2.01 \cdot 10^{-10}$	$2.58 \cdot 10^{-13}$	24 602	22 816	22 761	55	5
	$1 \cdot 10^{-12}$	$7.19 \cdot 10^{-12}$	$9.23 \cdot 10^{-15}$	47 226	45 953	45 895	58	9
DOPRI5	$1 \cdot 10^{-6}$	$4.01 \cdot 10^{-3}$	$1.98 \cdot 10^{-7}$	64 406	10 734	10 733	1	2
	$1 \cdot 10^{-8}$	$2.92 \cdot 10^{-5}$	$1.45 \cdot 10^{-9}$	64 736	10 789	10 789	0	2
	$1 \cdot 10^{-10}$	$2.88 \cdot 10^{-7}$	$1.42 \cdot 10^{-11}$	67 436	11 239	11 238	1	3
	$1 \cdot 10^{-12}$	$8.4 \cdot 10^{-9}$	$4.16 \cdot 10^{-13}$	82 142	13 690	13 689	1	3
	$1 \cdot 10^{-14}$	$9.08 \cdot 10^{-11}$	$4.49 \cdot 10^{-15}$	149 588	24 931	24 930	1	5
ROCK4	$1 \cdot 10^{-6}$	$5.52 \cdot 10^{-5}$	$6.22 \cdot 10^{-8}$	11 967	909	844	65	1
	$1 \cdot 10^{-8}$	$1.58 \cdot 10^{-6}$	$2.07 \cdot 10^{-9}$	20 312	2110	1966	144	1
	$1 \cdot 10^{-10}$	$1.57 \cdot 10^{-8}$	$2.04 \cdot 10^{-11}$	36 251	4958	4748	210	1
	$1 \cdot 10^{-12}$	$2.06 \cdot 10^{-10}$	$2.64 \cdot 10^{-13}$	72 180	11 617	11 466	151	2
	$1 \cdot 10^{-14}$	$3.82 \cdot 10^{-12}$	$4.9 \cdot 10^{-15}$	168 592	28 285	27 083	1202	6
RADAU5	$1 \cdot 10^{-6}$	$6.65 \cdot 10^{-7}$	$8.54 \cdot 10^{-10}$	542	66	59	7	<1
	$1 \cdot 10^{-8}$	$2.78 \cdot 10^{-7}$	$3.57 \cdot 10^{-10}$	915	111	108	3	1
	$1 \cdot 10^{-10}$	$4.93 \cdot 10^{-9}$	$6.32 \cdot 10^{-12}$	1839	219	219	0	1
	$1 \cdot 10^{-12}$	$1.81 \cdot 10^{-9}$	$2.33 \cdot 10^{-12}$	3661	460	460	0	1
	$1 \cdot 10^{-14}$	$3.02 \cdot 10^{-10}$	$3.87 \cdot 10^{-13}$	7271	977	977	0	2

It should be noted that the required and actually obtained errors for some methods are very different. This fact should be taken into account when comparing statistics.

Unlike one-step methods, reducing the grid step size always causes *a* rejected step, so in some cases the number of the rejected steps is quite large.

Burgers' equation. The second problem is taken from work [4]. The spatial derivatives are approximated by standard central finite differences, the discretisation step is $\Delta x = \frac{1}{501}$, so the dimension of the resulting ordinary differential equation is 500. We took $\mu = 0.005$ and the integration interval is equal to $[0, 2.5]$ (table 5).

Table 5

Numerical experiment results for the Burgers' equation

Method	*atol*	$rerr_{fin}$	$aerr_{fin}$	*fcn*	*step*	*accpt*	*rejct*	*time*
SA4-21	$1 \cdot 10^{-6}$	$2.82 \cdot 10^{-10}$	$6.08 \cdot 10^{-11}$	4912	4744	4741	3	29
	$1 \cdot 10^{-8}$	$2.54 \cdot 10^{-10}$	$5.65 \cdot 10^{-11}$	4713	4544	4542	2	28
	$1 \cdot 10^{-10}$	$2.69 \cdot 10^{-10}$	$4.55 \cdot 10^{-11}$	4996	4826	4825	1	30
	$1 \cdot 10^{-12}$	$5.88 \cdot 10^{-11}$	$9.16 \cdot 10^{-12}$	9273	9036	9032	4	54
SA3-21	$1 \cdot 10^{-6}$	$4.09 \cdot 10^{-7}$	$6.72 \cdot 10^{-8}$	3204	3045	3043	2	17
	$1 \cdot 10^{-8}$	$1.59 \cdot 10^{-7}$	$2.44 \cdot 10^{-8}$	4305	4144	4144	0	23
DOPRI5	$1 \cdot 10^{-6}$	$1.1 \cdot 10^{-6}$	$1.84 \cdot 10^{-7}$	22 784	3797	3794	1	25
	$1 \cdot 10^{-8}$	$1.1 \cdot 10^{-8}$	$1.84 \cdot 10^{-9}$	22 790	3798	3795	1	25
	$1 \cdot 10^{-10}$	$5.75 \cdot 10^{-10}$	$9.62 \cdot 10^{-11}$	22 814	3802	3798	2	25
	$1 \cdot 10^{-12}$	$3.94 \cdot 10^{-12}$	$6.42 \cdot 10^{-13}$	22 868	3811	3807	2	25

Ending table 5

Method	*atol*	$rerr_{fin}$	$aerr_{fin}$	*fcn*	*step*	*accpt*	*rejct*	*time*
ROCK4	$1 \cdot 10^{-6}$	$1.48 \cdot 10^{-3}$	$7.59 \cdot 10^{-5}$	1748	76	71	5	2
	$1 \cdot 10^{-8}$	$2.96 \cdot 10^{-5}$	$1.38 \cdot 10^{-6}$	2059	105	104	1	2
	$1 \cdot 10^{-10}$	$1.85 \cdot 10^{-8}$	$6.62 \cdot 10^{-10}$	3407	231	231	0	3
	$1 \cdot 10^{-12}$	$4.33 \cdot 10^{-11}$	$2.31 \cdot 10^{-12}$	7537	835	835	0	7
RADAU5	$1 \cdot 10^{-6}$	$2.7 \cdot 10^{-6}$	$3.14 \cdot 10^{-7}$	145	22	22	0	2
	$1 \cdot 10^{-8}$	$1.28 \cdot 10^{-7}$	$1.42 \cdot 10^{-8}$	255	39	39	0	3
	$1 \cdot 10^{-10}$	$2.69 \cdot 10^{-9}$	$3.18 \cdot 10^{-10}$	520	76	76	0	5
	$1 \cdot 10^{-12}$	$8.07 \cdot 10^{-11}$	$1.19 \cdot 10^{-11}$	1104	159	159	0	9

The stability region of the SA4-21 method is not enough in the case of low required accuracy. The SA3-21 method has a 1.5 time larger stability interval and works slightly better in these cases.

Conclusions

Based on the experimental results obtained, we can conclude that the stabilised explicit Adams-type methods of higher orders with damping can be useful in solving middly-stiff differential systems with real (or close to real) eigenvalues of the Jacobian matrix. If it is possible to improve the algorithm by adding to it an adaptive choice of parameters p and k, in some cases it can become the optimal variant, due to the smaller number of the right-hand side evaluations.

References

1. Repnikov VI, Faleichik BV, Moisa AV. Stabilised explicit Adams-type methods. *Journal of the Belarusian State University. Mathematics and Informatics.* 2021;2:82–98. DOI: 10.33581/2520-6508-2021-2-82-98.

2. Hairer E, Wanner G. *Solving ordinary differential equations II: stiff and differential-algebraic problems.* 2nd edition. Berlin: Springer; 1996. XV, 614 p. (Springer series in computational mathematics; volume 14). DOI: 10.1007/978-3-642-05221-7.

3. Abdulle A, Medovikov AA. Second order Chebyshev methods based on orthogonal polynomials. *Numerische Mathematik.* 2001;90(1):1–18. DOI: 10.1007/s002110100292.

4. Abdulle A. Fourth order Chebyshev methods with recurrence relation. *SIAM Journal on Scientific Computing.* 2002;23(6): 2041–2054. DOI: 10.1137/S1064827500379549.

5. Hairer E, Nørsett SP, Wanner G. *Solving ordinary differential equations I: nonstiff problems*. 2nd edition. Berlin: Springer; 1993. XV, 528 p. (Springer series in computational mathematics; volume 8). DOI: 10.1007/978-3-540-78862-1.

6. Lebedev VI. How to solve stiff systems of differential equations by explicit methods. In: Marchuk GI, editor. *Numerical methods and applications.* Boca Raton: CRC Press; 1994. p. 45–80.

7. Dormand JR, Prince PJ. A family of embedded Runge – Kutta formulae. *Journal of Computational and Applied Mathematics.* 1980;6(1):19–26. DOI: 10.1016/0771-050X(80)90013-3.

8. Schäfer E. A new approach to explain the «high irradiance responses» of photomorphogenesis on the basis of phytochrome. *Journal of Mathematical Biology.* 1975;2(1):41–56. DOI: 10.1007/BF00276015.

Received 12.09.2022 / revised 10.02.2023 / accepted 11.02.2023.