Computers and Mathematics with Applications 60 (2010) 3032–3045

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Adams–Bashforth and Adams–Moulton methods for solving differential Riccati equations

J. Peinado[a,*], J. Ibáñez[a], E. Arias[b], V. Hernández[a]

[a] Departamento de Sistemas Informáticos y Computación, Technical University of Valencia, Camino de Vera s/n, 46022-Valencia, Spain

[b] Departamento de Sistemas Informáticos, University of Castilla-La Mancha, Avda. España s/n, 02071-Albacete, Spain

ARTICLE INFO

Article history:
Received 2 April 2009
Received in revised form 1 October 2010
Accepted 1 October 2010

Keywords:
differential matrix Riccati equation (DMRE)
algebraic matrix Riccati equation (AMRE)
algebraic matrix Sylvester equation (AMSE)
Adams–Bashforth methods
Adams–Moulton methods
GMRES methods
fixed-point method

ABSTRACT

Differential Riccati equations play a fundamental role in control theory, for example, optimal control, filtering and estimation, decoupling and order reduction, etc. In this paper several algorithms for solving differential Riccati equations based on Adams–Bashforth and Adams–Moulton methods are described. The Adams–Bashforth methods allow us explicitly to compute the approximate solution at an instant time from the solutions in previous instants. In each step of Adams–Moulton methods an algebraic matrix Riccati equation (AMRE) is obtained, which is solved by means of Newton's method. Nine algorithms are considered for solving the AMRE: a Sylvester algorithm, an iterative generalized minimum residual (GMRES) algorithm, a fixed-point algorithm and six combined algorithms. Since the above algorithms have a similar structure, it is possible to design a general and efficient algorithm that uses one algorithm or another depending on the considered differential matrix Riccati equation.

MATLAB versions of the above algorithms are developed, comparing precision and computational costs, after numerous tests on five case studies.

1. Introduction

In this paper we consider DMREs of the form

$$\begin{aligned}
&\dot{X}(t) = A_{21}(t) + A_{22}(t)X(t) - X(t)A_{11}(t) - X(t)A_{12}(t)X(t), \\
&t_0 \le t \le t_f, \\
&X(t_0) = X_0 \in \mathbb{R}^{m\times n},
\end{aligned} \tag{1}$$

where $A_{11}(t) \in \mathbb{R}^{n\times n}, A_{12}(t) \in \mathbb{R}^{n\times m}, A_{21}(t) \in \mathbb{R}^{m\times n}, A_{22}(t) \in \mathbb{R}^{m\times m}$.

DMREs arises in several applications, in particular in control theory, for example the time-invariant linear quadratic optimal control problem. In this case, the DMRE has the following expression

$$\dot{X}(t) = Q + A^T X(t) + X(t)A - X(t)BR^{-1}B^T X(t), \tag{2}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, Q = Q^T \in \mathbb{R}^{n\times n}$ is positive semidefinite, and $R = R^T \in \mathbb{R}^{m\times m}$ is positive definite, representing respectively, the state matrix, the input matrix, the state weight matrix and the input weight matrix. Another application of the DMRE (1) consists of solving a two point boundary problem by decoupling this problem into two initial

* Corresponding author. Tel.: +34 96 3877798; fax: +34 96 3877359.
E-mail addresses: jpeinado@dsic.upv.es (J. Peinado), jjibanez@dsic.upv.es (J. Ibáñez), earias@dsi.uclm.es (E. Arias), vhernand@dsic.upv.es (V. Hernández).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.10.002

value problems [1]. Since the mid-1970s, many different methods have been proposed: the linearization approach [2–4], Chandrasekhar approach [5], superposition methods [6,7], the BDF approach [1,8–11], the Hamiltonian approach [12], unconventional reflexive numerical methods [13], etc.

This paper is organized as follows. First, Sections 2 and 3 describe an Adams–Bashforth method and a family of Adams–Moulton methods. Section 4 shows the developed algorithms. Section 5 presents the test battery and the experimental results. Finally, the conclusions and future work are outlined in Section 6.

2. Adams–Bashforth method

Let $F(t, X)$ be the right hand side of (1),

$$F(t, X) = A_{21}(t) + A_{22}(t)X(t) - X(t)A_{11}(t) - X(t)A_{12}(t)X(t).$$

If we consider a constant step size Δt and a mesh $t_0 \le t_1 \le t_2 \le \cdots \le t_f$, and we apply a Adams–Bashforth scheme, then the approximate solution X_k at t_k is obtained from the previous values $X_{k-1}, x_{k-2}, \dots, X_{k-r}$ as

$$X_k = X_{k-1} + \Delta t \sum_{j=1}^{r} \beta_j F(t_{k-j}, X_{k-j}),$$

where

$$\gamma_i = (-1)^i \int_0^1 \binom{-s}{i} \mathrm{d}s, \qquad \beta_j = (-1)^{j-1} \sum_{i=j-1}^{k-1} \binom{i}{j-1} \gamma_i.$$

This formula is a k-step method because it uses information at the r points $t_{k-1}, t_{k-2}, \dots, t_{k-r}$. The values of parameters γ_i and β_i can be found in [14, Chapter 5].

3. Adams–Moulton methods

In this section we will describe a family of Adams–Moulton methods for solving DMREs. In an Adams–Moulton scheme, the integration interval $[t_0, t_f]$ is divided so that the approximate solution at t_k, X_k, is obtained by solving an AMRE. Several methods have been implemented for solving AMREs, however one of the better choices for solving the associated AMRE is to apply implicit schemes based on Newton's or quasi-Newton's methods.

If we consider a mesh $t_0 \le t_1 \le t_2 \le \cdots \le t_f$ and we apply an Adams–Moulton scheme, then the approximate solution X_k at t_k is obtained by solving the following matrix equation

$$X_k = X_{k-1} + \Delta t \sum_{j=0}^{r} \bar{\beta}_j F(t_{k-j}, X_{k-j}),$$

$$-X_k + \Delta t \bar{\beta}_0 F(t_k, X_k) + X_{k-1} + \Delta t \sum_{j=1}^{r} \bar{\beta}_j F(t_{k-j}, X_{k-j}) = 0, \tag{3}$$

where $\bar{\beta}_j, j = 0, 1, \dots, r$, are the parameters of Adams–Moulton Methods, which can be found in [14, Chapter 5].

Eq. (3) can be expressed as the AMRE

$$\bar{A}_{21} + \bar{A}_{22} X_k + X_k \bar{A}_{11} + X_k \bar{A}_{12} X_k = 0, \tag{4}$$

where

$$\bar{A}_{21} = X_{k-1} + \Delta t \sum_{j=1}^{r} \bar{\beta}_j F(t_{k-j}, X_{k-j}) + \Delta t \bar{\beta}_0 A_{21},$$
$$\bar{A}_{22} = -I_m + \Delta t \bar{\beta}_0 A_{22},$$
$$\bar{A}_{11} = -\Delta t \bar{\beta}_0 A_{11},$$
$$\bar{A}_{12} = -\Delta t \bar{\beta}_0 A_{12}.$$

The methods for solving AMRE (4) are presented in the subsections below.

3.1. A Sylvester method

Eq. (4) can be solved by the Newton's iteration

$$\hat{G}'_{(l)}(X_k^l - X_k^{l-1}) = -G(X_k^{l-1}), \quad l \ge 1, \tag{5}$$
$$X_k^0 = X_{k-1},$$

where $\hat{G}'_{(l)}$ is the Fréchet derivative [15, p. 310] of

$$G(X) = \bar{A}_{21} + \bar{A}_{22}X + X\bar{A}_{11} + X\bar{A}_{12}X.$$

For a fixed l the following AMSE is obtained

$$C_{22}^{l-1}\Delta X_k^{l-1} - \Delta X_k^{l-1}C_{11}^{l-1} = C_{21}^{l-1}, \tag{6}$$

where

$$C_{22}^{l-1} = \bar{A}_{22} + X_k^{l-1}\bar{A}_{12},$$
$$C_{11}^{l-1} = \bar{A}_{11} + \bar{A}_{12}X_k^{l-1},$$
$$C_{21}^{l-1} = -\bar{A}_{21} - \bar{A}_{22}X_k^{l-} - X_k^{l-1}C_{11}^{l-1},$$

and $\Delta X_k^{l-1} = X_k^l - X_k^{l-1}$. X_k^l can be obtained by solving (6) for ΔX_k^{l-1} and computing $X_k^l = \Delta X_k^{l-1} + X_k^{l-1}$. The standard solution process for (6) is the Bartels–Stewart algorithm [16].

3.2. A GMRES method

This method was developed by the authors in [11]. If we apply the `vec` operator [17, p. 244] to (6), then

$$[I_n \otimes C_{22}^{l-1} - (C_{11}^{l-1})^T \otimes I_m]\Delta x_k^{l-1} = \mathtt{vec}(C_{21}^{l-1}).$$

This linear system can be solved efficiently without explicitly building the matrix $I_n \otimes C_{22}^{l-1} - (C_{11}^{l-1})^T \otimes I_m$ by a GMRES method, and then

$$X_k^l = X_k^{l-1} + \mathtt{mat}(\Delta x_k^{l-1}, m, n),$$

where `mat` is an operator [11, p. 617] which converts a vector into a matrix.

3.3. A fixed-point method

This method was developed by the authors in [10]. From (4) we obtain

$$\bar{A}_{21} + X_k\bar{A}_{11} + (\bar{A}_{22} + X_k\bar{A}_{12})X_k = 0,$$
$$(\bar{A}_{22} + X_k\bar{A}_{12})X_k = -(\bar{A}_{21} + X_k\bar{A}_{11})$$

that permits us to define the following fixed-point iteration

$$(\bar{A}_{22} + X_k^{l-1}\bar{A}_{12})X_k^l = -(\bar{A}_{21} + X_k^{l-1}\bar{A}_{11}), \quad l \geqslant 1, \tag{7}$$
$$X_k^0 = X_{k-1}.$$

Similarly, from (4) we obtain the fixed-point iteration

$$X_k^l(\bar{A}_{11} + \bar{A}_{12}X_k^{l-1}) = -(\bar{A}_{21} + \bar{A}_{12}X_k^{l-1}), \quad l \geqslant 1, \tag{8}$$
$$X_k^0 = X_{k-1}.$$

3.4. Other methods

The above methods can be modified to improve speed and convergence. The idea consists of combining two of the above methods applying a first step using one method, and then, applying several steps using a second method making the necessary iterations (steps) to reach convergence. This give us two kinds of methods, 'single' methods, where only one method is used, and 'combined' methods, where two methods are used.

Using the three methods explained before (Sylvester, GMRES and fixed point), it is possible to have nine combinations of single (three) and combined (six) methods. The combined methods are:

(1) Apply a fixed-point iteration and the Sylvester method.
(2) Apply a fixed-point iteration and the GMRES method.
(3) Apply a Sylvester iteration and the fixed-point method.
(4) Apply a Sylvester iteration and the GMRES method.
(5) Apply a GMRES iteration and the Sylvester method.
(6) Apply a GMRES iteration and the fixed-point method.

4. Algorithms for solving DMREs

Two kind of algorithms have been developed: driver algorithms and computational algorithms. Driver algorithms solve DMREs by using Adams–Bashforth or Adams–Moulton methods. Computational algorithms are single or combined algorithms for solving AMREs.

4.1. A nomenclature to describe the algorithms

In this subsection two driver algorithms and nine single and one combined computational algorithms are described. The driver routines have names of the form XYYZZZVVVV where:

- X indicates data type (d = Double).
- YY indicates the type of matrix (ge = GEneral).
- ZZZ indicates the problem solved:
  - `vdr`: time varying differential Riccati equation.
  - `idr`: invariant differential Riccati equation.
  - `are`: algebraic Riccati equation.
- VVVV indicates the method used to solve the problem:
  - For the Adams–Bashforth algorithm VVVV is equal to `adba`.
  - For the Adams–Moulton algorithm family VVVV is equal to `admo`.

For computational algorithms only three characters VVV are used to indicate the method used. If the method is single, VVV indicates the method: `syl` (for Sylvester), `fpo` (for fixed point) and `gmr` (for GMRES). If the method is combined, the first letter indicates the method used, and the second and third, the kind of approximation: `gfp` (GMRES method with a fixed-point approximation, see Algorithm 2), etc.

4.2. Adams–Bashforth algorithm

Algorithm 1 solves DMREs using the Adams–Bashforth method.

Algorithm 1. $\{X_k\}_{k=1}^{l} = \mathtt{dgevdradba}(t, data, X_0, r)$.

Inputs: Time vector $t \in \mathbb{R}^{l+1}$; $\mathtt{data}(\tau)$ function that computes the coefficient matrices of (1); starting guess matrix $X_0 \in \mathbb{R}^{m\times n}$; $r \in \{1, 2, 3, 4, 5\}$ is the order of the Adams–Bashforth method.

Output: Matrices $X_k \in \mathbb{R}^{m\times n}$, $k = 1, \ldots, l$.

1. Initialize β.
2. For $k = 1 : l$.
   2.1. $\Delta t = t_k - t_{k-1}$.
   2.2. $r_1 = \min(r, k)$.
   2.3. $S = 0_{m\times n}$.
   2.4. For $j = r_1$ to 2 step -1.
      2.4.1. $F_j = F_{j-1}$.
      2.4.2. $S = S + \beta_{r_1 j} F_j$.
   2.5. $[A_{11}, A_{12}, A_{21}, A_{22}] = \mathtt{data}(t_k)$.
   2.6. $F_1 = A_{21} + A_{22}X_{k-1} - X_{k-1}(A_{11} + A_{12}X_{k-1})$.
   2.7. $X_k = X_{k-1} + \Delta t(S + \beta_{r_1 1} F_1)$.

4.3. Adams–Moulton algorithm scheme

A general algorithm scheme which allows us easily to choose 'the best method' for each DMRE problem based on the above Adams–Moulton method is presented in this subsection. With this scheme up to nine different Adams–Moulton algorithms can be applied.

There are two kind of algorithms: driver algorithms and computational algorithms. Driver algorithms solve DMREs by using Adams–Moulton methods. Computational algorithms are single or combined approaches for solving AMREs (Section 3.4). All the above possibilities have been implemented, but only one of them is shown. Therefore there is a driver algorithm for the invariant case named `dgeidradmo` (not shown) and another for the time-varying case (Algorithm 2) named `dgevdradmo`. This algorithm solves a time-varying DMRE by calling other algorithms to solve the corresponding AMRE. There are nine algorithms for solving AMREs (see Fig. 1) but only one of them (Algorithm 3) is shown. Other algorithms can be studied in [10].

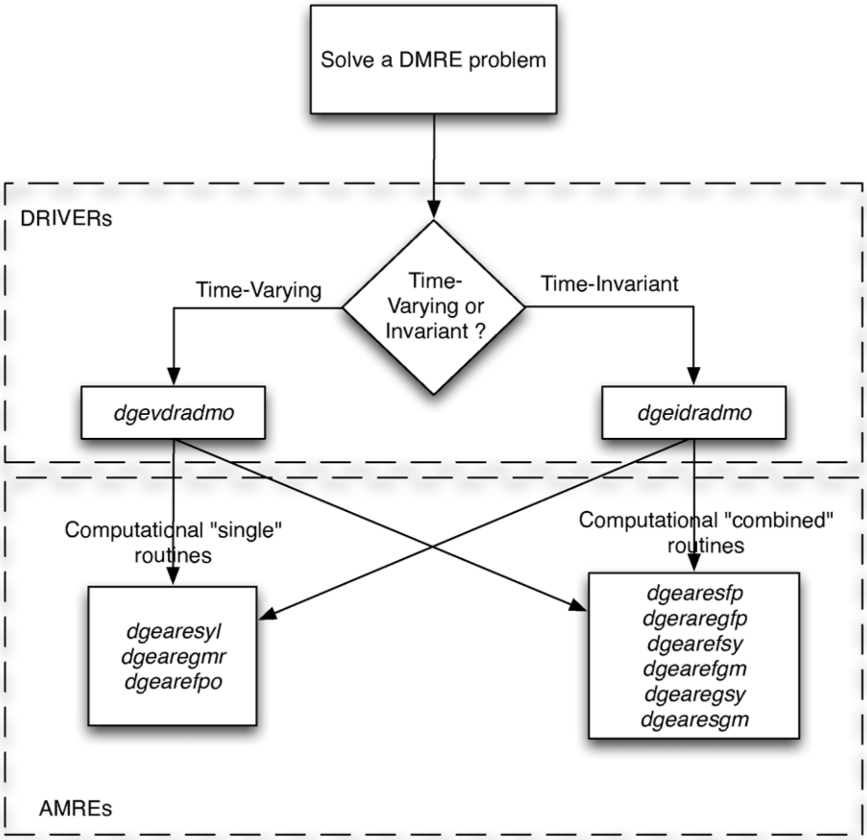

Fig. 1. Implemented implicit algorithms and their inter-dependences.

Algorithm 2. $\{X_k\}_{k=1}^{l} = \mathtt{dgevdradmo}(t, \mathtt{data}, X_0, r, \mathtt{met}, tol, maxiter)$.

Inputs: *Time vector* $t \in \mathbb{R}^{l+1}$; $\mathtt{data}(t)$ is the function that computes the coefficient matrices of (1) at instant t; starting guess matrix $X_0 \in \mathbb{R}^{m\times n}$; $r \in \{1, 2, 3, 4, 5\}$ is the order of the Adams–Bashforth method; met is the method used for solving the AMRE associated with the DMRE; tol $\in \mathbb{R}^+$ is the tolerance of the iterative method used in the Adams–Moulton method; *maxiter* $\in \mathbb{N}$ is the maximum number of the iterative method used in the Adams–Moulton method.

Output: Matrices $X_k \in \mathbb{R}^{m\times n}$, $k = 1, 2, \ldots, l$.

1. Initialize $\bar{\beta}$.
2. For $k = 1 : l$.
   2.1. $\Delta t = t_k - t_{k-1}$.
   2.2. $r_1 = \min(r, k)$.
   2.3. $S = 0_{m\times n}$.
   2.4. For $j = r_1$ to 2 step -1.
      2.4.1. $F_j = F_{j-1}$.
   2.5. $F_1 = A_{21} + A_{22}X_{k-1} - X_{k-1}(A_{11} + A_{12}X_{k-1})$.
   2.6. $[A_{11}, A_{12}, A_{21}, A_{22}] = \mathtt{data}(t_k)$.
   2.7. $A_{21} = X_{k-1} + \Delta t \sum_{j=1}^{r} \bar{\beta}_j F(t_{k-j}, X_{k-j}) + \Delta t \bar{\beta}_0 A_{21}$.
   2.8. $A_{22} = -I_m + \Delta t \bar{\beta}_0 A_{22}$.
   2.9. $A_{11} = -\Delta t \bar{\beta}_0 A_{11}$.
   2.10. $A_{12} = -\Delta t \bar{\beta}_0 A_{12}$.
   2.11. $X_k = X_{k-1} + \Delta t(S + \beta_{r_1 1} F_1)$.
   2.12. Solve $A_{21} + A_{22}X_k + X_k A_{11} + X_k A_{12} X_k = 0$ for X_k by the met method.

Algorithm 3 solves the AMRE (4) using the Newton–GMRES method.

Algorithm 3. $[X_k, l] = \texttt{dgearegmr}(A_{11}, A_{12}, A_{21}, A_{22}, X_{k-1}, tol, maxiter)$.

Inputs: Coefficient matrices A_{11}, A_{12}, A_{21} and A_{22} of AMRE (4); starting guess matrix $X_{k-1} \in \mathbb{R}^{m\times n}$; tolerance $tol \in \mathbb{R}^+$ used in Newton's method; maximum number of Newton iterations $maxiter \in \mathbb{N}$.

Outputs: Solution matrix $X_k \in \mathbb{R}^{m\times n}$; $l \in \mathbb{Z}$ is the number of Newton iterations used to reach convergence, or -1 if convergence is not reached

1. $X_k = X_{k-1}$.
2. $l = 1$.
3. While $l \leqslant maxiter$
   3.1. $C_{22} = A_{22} + X_k A_{12}$.
   3.2. $C_{11} = A_{11} + A_{12} X_k$.
   3.3. $C_{21} = -A_{21} - A_{22} X_k - X_k C_{11}$.
   3.4. Solve $[I_n \otimes C_{22} - (C_{11})^T \otimes I_m]\Delta x = \texttt{vec}(C_{21})$ for Δx by using a GMRES algorithm.
   3.5. $X_k = X_k + \texttt{mat}(\Delta x, m, n)$.
   3.6. If $\|\Delta x\|_\infty < tol$.
      3.6.1. Leave while loop
   3.7. $l = l + 1$.
4. End While
5. If $l > maxiter$.
   5.1. $l = -1$
6. End If

5. Experimental results

The main objective of this section is to compare the algorithms developed in the previous sections. The implementations have been tested on an Intel Centrino Core 2 Duo T5470 at 1.6 GHz with 2 GB main memory, and using MATLAB 7.0. As test battery, five case studies have been considered. For each case study, the characteristic parameters of each method which offer better accuracy and lower execution time have been determined.

For each case study step size, final time, dimension of problem or stiffness have been varied in order to study the particular behaviour of the algorithms developed in this work. The particular values are highlighted in each case study.

The parameters of the methods have been adjusted in each case study to obtain the best execution time (*Te*) with the least relative error (using the analytic solution and the obtained solution). Then, for each test we show results for execution time and relative error. The relative error is computed as

$$E_r = \frac{\|X - X^*\|_\infty}{\|X\|_\infty},$$

where X^* is the computed solution and X is the analytic solution (case studies 2, 3) or an approximate solution (case studies 1, 4 and 5).

The relative errors are shown in tables. Since the errors of functions `dgeidradmo` and `dgevdradmo` are similar for the different AMRE solvers used (see Fig. 1), only two entries are used for each table: the first for `dgeidradba/dgevdradba` and the second for `dgeidradmo/dgevdradmo`.

The execution times in seconds are shown in figures. In each figure appear one line for `dgeidradba/dgevdradba` and nine lines for `dgeidradmo/dgevdradmo` (one for each AMRE solver used).

5.1. Case studies

5.1.1. Case Study 1

The first time-invariant DMRE is a non-stiff case taken from a two-point boundary value problem in [18]. This DMRE is defined for $t \geq 0$ by the coefficient matrices

$$A_{11} = \begin{bmatrix} 0 & 0 \\ -100 & -1 \end{bmatrix}, \qquad A_{12} = \begin{bmatrix} 0 & 1 \\ 100 & 0 \end{bmatrix},$$

$$A_{12} = \begin{bmatrix} 0 & 1 \\ 100 & 0 \end{bmatrix}, \qquad A_{22} = \begin{bmatrix} 0 & 0 \\ -10 & -1 \end{bmatrix},$$

and the initial condition

$$X(0) = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}.$$

If the value of t is large, the solution of the previous DMRE is approximately equal to

$$X = \begin{bmatrix} 1 & 0.110 \\ 0 & -0.1 \end{bmatrix}. \tag{9}$$

In this case study the characteristic parameters are the following:

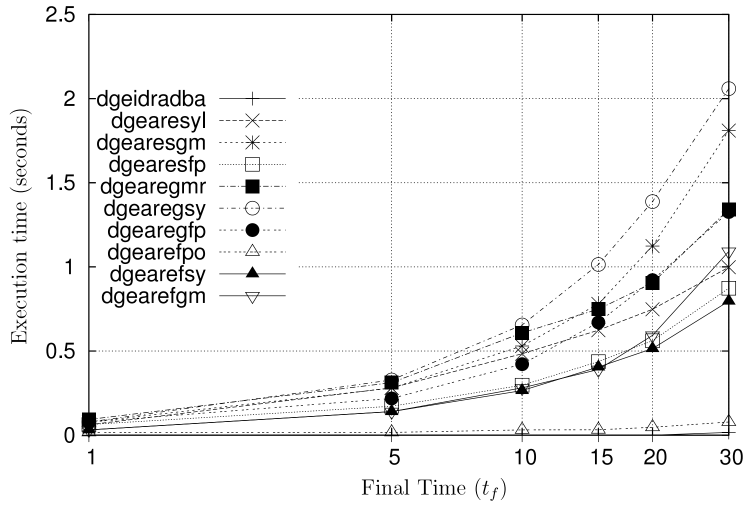

Fig. 2. Execution time (seconds) considering $\Delta t = 0.05$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$ for Case Study 1.

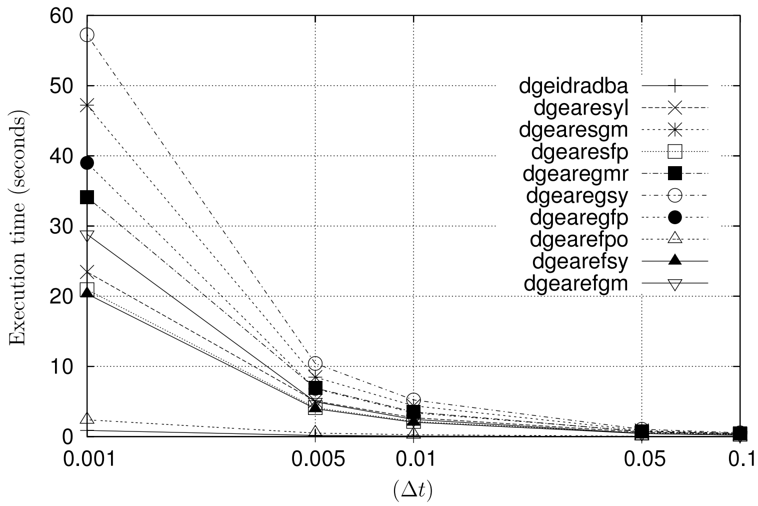

Fig. 3. Execution time (seconds) considering $t_f = 15$, $r = 3$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$ for Case Study 1.

Table 1
Table reporting relative error for the solution of DMRE for Case Study 1 considering $\Delta t = 0.05$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$.

t_f	1	5	10	15	20	30
dgeidradba	1.4227e−1	1.9697e−3	1.2253e−5	7.2541e−8	4.2948e−10	1.5093e−14
dgeidradmo	1.4941e−2	2.5665e−4	1.8157e−6	1.2221e−8	8.2260e−11	3.6758e−15

- Fixed parameters
  - Problem size. For this case study $n = 1$.
  - Maximum number of iterations of Newton's method: $maxiter = 100$.
  - Relative tolerance: $tol = 10^{-6}$.
- Variable parameters
  - Final time: $t_f = [1, 5, 10, 15, 20, 30]$.
  - Step size: $\Delta t = [0.1, 0.05, 0.01, 0.005, 0.001]$.
  - Order of Adams–Moulton or Adams–Bashforth methods: $r = [1, 2, 3]$.

Figs. 2 and 3 and Tables 1 and 2 show execution times and accuracy (relative errors) when $\Delta t = 0.05$, $r = 1$ and varying final time $t_f = 1, 5, 10, 15, 20, 30$, and when $t_f = 5$, $r = 2$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Table 2
Table reporting relative error for the solution of DMRE for Case Study 1 considering $t_f = 15$, $r = 3$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Δt	0.1	0.05	0.01	0.005	0.001
`dgeidradba`	−1.0000e0	2.1411e−1	3.0584e−9	8.4365e−10	3.6262e−11
`dgeidradmo`	8.0152e−10	1.1855e−11	2.2768e−11	1.5735e−11	4.0735e−12

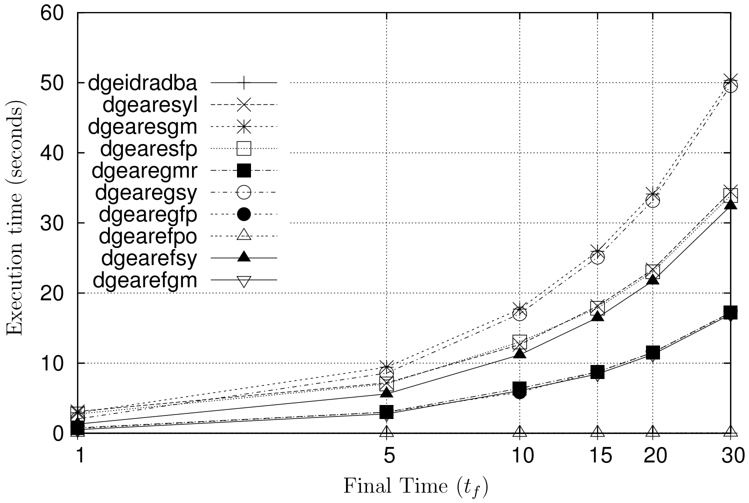

Fig. 4. Execution time (seconds) considering $\alpha = 10$, $\Delta t = 0.01$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$ for Case Study 2.

Table 3
Table reporting relative error for the solution of DMRE for Case Study 2 considering $\alpha = 10$, $\Delta t = 0.01$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$.

t_f	1	5	10	15	20	30
`dgeidradba`	5.5789e−10	0	0	0	0	0
`dgeidradmo`	7.5503e−11	0	0	0	0	0

5.1.2. Case Study 2

The second case study [19,1] consists of the following time-invariant DMRE

$$\dot{X}(t) = A_{21} + A_{22}X(t) - X(t)A_{11} - X(t)A_{12}X(t), \quad 0 \le t \le t_f,$$

where $A_{11} = 0_n$, $A_{12} = A_{21} = \alpha I_n$, $(\alpha > 0)$, $A_{22} = 0_n$, and $X_0 \in \mathbb{R}^{n\times n}$.

The exact solution is given by

$$X(t) = (\alpha(X_0 + I_n)e^{\alpha t} - \alpha(X_0 - I_n)e^{-\alpha t})^{-1}(\alpha(X_0 + I_n)e^{\alpha t} + \alpha(X_0 - I_n)e^{-\alpha t}),$$

which allows the approaches presented in this document to be compared in terms of accuracy.

For this case study the characteristic parameters are the following:

- Fixed parameters
  - · Problem size: $n = 20$.
  - · Maximum number of iterations of Newton's method: $maxiter = 100$.
  - · Relative tolerance: $tol = 10^{-12}$.
- Variable parameters
  - · Stiffness: $\alpha = [1, 10, 100]$.
  - · Final time: $t_f = [1, 5, 10, 15, 20, 30]$.
  - · Step size: $\Delta t = [0.1, 0.05, 0.01, 0.005, 0.001]$.
  - · Order of Adams–Moulton or Adams–Bashforth method: $r = [1, 2, 3]$.

Figs. 4 and 5 and Tables 3 and 4 show execution times and relative errors when $\alpha = 10$, $\Delta t = 0.1$, $r = 1$ and varying final time $t_f = 1, 5, 10, 15, 20, 30$, and when $\alpha = 1$, $t_f = 10$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

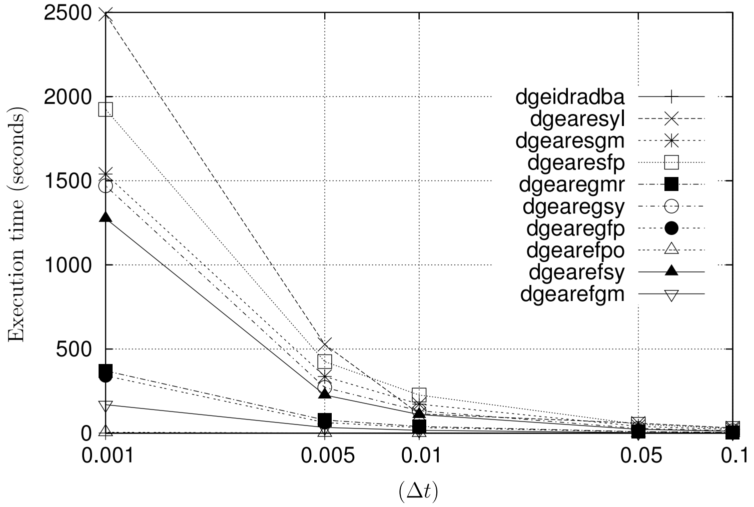

Fig. 5. Execution time (seconds) considering $pr = 1$, $t_f = 10$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$ for Case Study 2.

Table 4
Table reporting relative error for the solution of the DMRE for Case Study 2 considering $\alpha = 1$, $t_f = 10$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Δt	0.1	0.05	0.01	0.005	0.001
`dgeidradba`	3.7603e−9	5.6445e−9	7.3862e−9	7.8760e−9	8.1522e−9
`dgeidradmo`	5.3699e−9	7.4072e−9	7.9191e−9	8.1620e−9	8.2117e−9

5.1.3. *Case Study 3*

Case Study 3 [20] consists of the following time-invariant DMRE

$$\begin{aligned}
&\dot{X}(t) = X(t)T_{2^k} + T_{2^k}X(t) - X(t)T_{2^k}X(t) + \alpha T_{2^k}, \quad t \geq 0,\\
&X(0) = I_{2^k},
\end{aligned}$$

where $k \in \mathbb{N}$, $\alpha \geq 1$ and $X(t), T_{2^k} \in \mathbb{R}^{2^k \times 2^k}$. Matrices T_2^k are generated as follows:

$$\begin{aligned}
T_2 &= \begin{bmatrix} -1 & 1 \\ \alpha & 1 \end{bmatrix},\\
T_{2^k} &= \begin{bmatrix} -T_{2^{k-1}} & T_{2^{k-1}} \\ \alpha T_{2^{k-1}} & T_{2^{k-1}} \end{bmatrix}, \quad k \geq 1,
\end{aligned}$$

where α controls the stiffness of the problem. The solution of this DMRE is given by

$$X(t) = I_{2^k} + \frac{\alpha + 1}{\omega} \tanh \omega t T_{2^k},$$

where $\omega = (\alpha + 1)^{\frac{k+1}{2}}$.

For this case study the characteristic parameters are the following:

- Fixed parameters
  - Problem size: $n = m = 16$ ($k = 4$).
  - Maximum number of iterations of Newton's method: $maxiter = 100$.
  - Relative tolerance: $tol = 10^{-12}$.
- Variable parameters
  - Stiffness: $\alpha = [1, 10]$.
  - Final time: $t_f = [1, 5, 10, 15, 20, 30]$.
  - Step size $\Delta t = [0.1, 0.05, 0.01, 0.005, 0.001]$.
  - Order of Adams–Moulton or Adams–Bashforth method: $r = [1, 2, 3]$.

As in Case Study 2, the α parameter has a special role due to the fact that it determines the stiffness of the problem.

Figs. 6 and 7 and Tables 5 and 6 show execution times and relative errors when $\alpha = 1$, $\Delta t = 0.1$, $r = 1$ and final time is varied, $t_f = 1, 5, 10, 15, 20, 30$, and when $\alpha = 1$, $t_f = 1$, $r = 1$ and $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

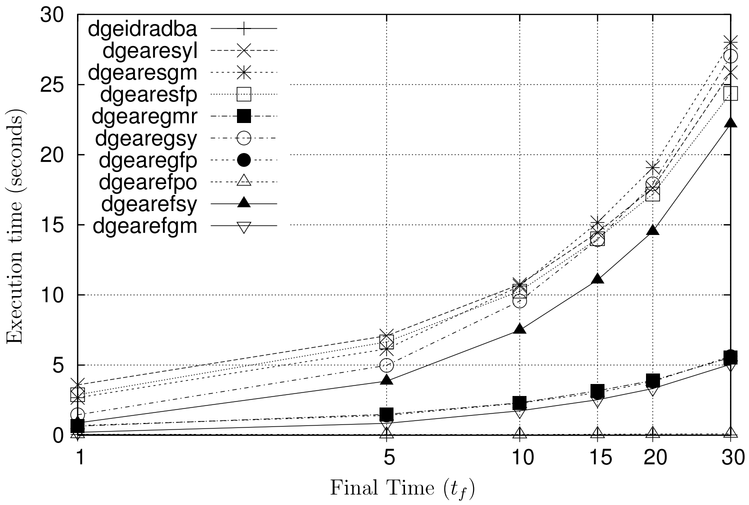

Fig. 6. Execution time (seconds) considering $\alpha = 1$, $\Delta t = 0.1$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$ for Case Study 3.

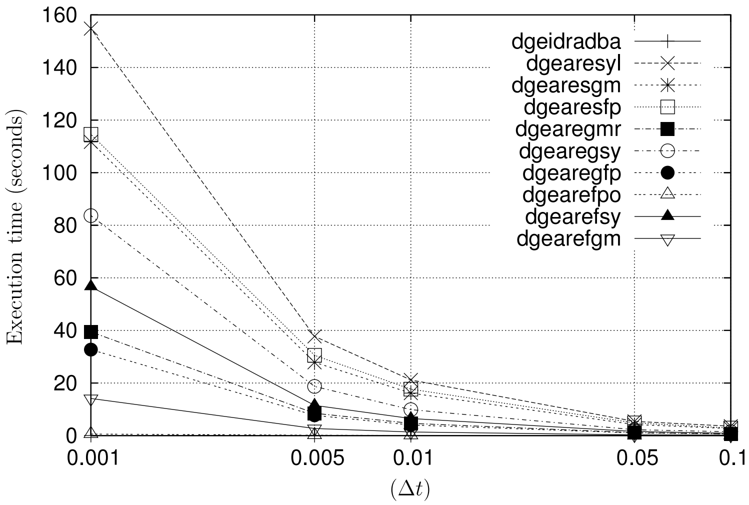

Fig. 7. Execution time (seconds) considering $\alpha = 1$, $t_f = 1$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$ for Case Study 3.

Table 5
Table reporting relative error for the solution of DMRE for Case Study 3 considering $\alpha = 1$, $\Delta t = 0.1$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$.

t_f	1	5	10	15	20	30
dgeidradba	4.6128e−6	2.3402e−16	2.2828e−16	2.3256e−16	2.0997e−16	2.2834e−16
dgeidradmo	1.1211e−6	3.3514e−16	3.0993e−16	3.0724e−16	3.1141e−16	3.1413e−16

Table 6
Table reporting relative error for the solution of DMRE for Case Study 3 considering $\alpha = 1$, $t_f = 1$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Δt	0.1	0.05	0.01	0.005	0.001
dgeidradba	4.6128e−6	1.4156e−5	6.4976e−6	3.6143e−6	7.8637e−7
dgeidradmo	1.1211e−6	1.7861e−6	1.1447e−7	3.0140e−8	1.2554e−9

5.2. Case Study 4

This time-varying DMRE [21,1] comes from a stiff two-point boundary value problem. This DMRE is defined as

$$A_{11}(t) = \begin{bmatrix} -t/2\varepsilon & 0 \\ 0 & 0 \end{bmatrix}, \qquad A_{12}(t) = \begin{bmatrix} 1/\varepsilon & 0 \\ 0 & 1/\varepsilon \end{bmatrix},$$

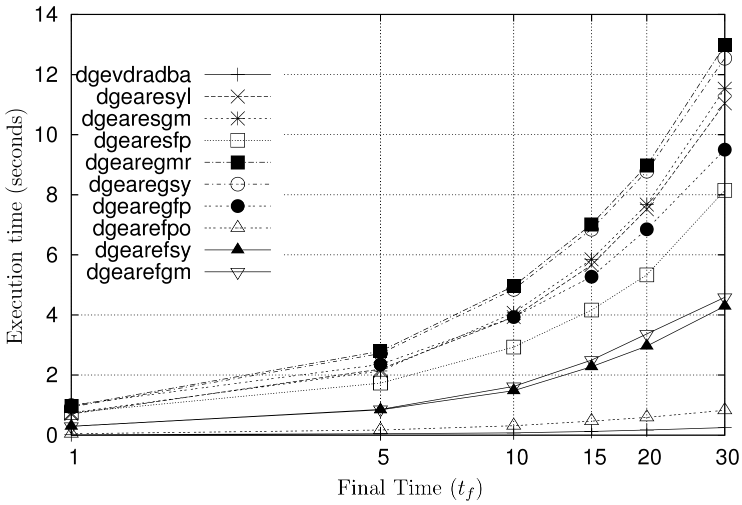

Fig. 8. Execution time (seconds) considering $\epsilon = 0.1$, $\Delta t = 0.01$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$ for Case Study 4.

Table 7
Table reporting relative error for the solution of DMRE for Case Study 4 considering $\epsilon = 0.1$, $\Delta t = 0.01$, $r = 1$ and varying $t_f = 1, 5, 10, 15, 20, 30$.

t_f	1	5	10	15	20	30
dgevdradba	1.6235e−2	2.5966e−16	9.5219e−17	1.2780e−16	1.8079e−16	1.2221e−16
dgevdradmo	1.5911e−2	4.0728e−16	2.5350e−16	1.6930e−16	1.9180e−16	1.2792e−16

$$A_{21}(t) = \begin{bmatrix} 1/2 & 1 \\ 0 & 1 \end{bmatrix}, \qquad A_{22}(t) = \begin{bmatrix} 0 & t/2\varepsilon \\ 0 & 0 \end{bmatrix},$$

where $t \geq -1$, $0 < \varepsilon \ll 1$. The initial condition is

$$X(-1) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The solution has an initial layer and then it approaches

$$X(t) = \begin{bmatrix} -\varepsilon/t & 2(\sqrt{\varepsilon}+1)/2(\sqrt{\varepsilon}-1) \\ 0 & \sqrt{\varepsilon} \end{bmatrix}.$$

For t away from 0, there is a smooth transition around the origin and then

$$X(t) \cong \begin{bmatrix} t/2 & \sqrt{\varepsilon} \\ 0 & \sqrt{\varepsilon} \end{bmatrix}.$$

For this case study the characteristic parameters are the following:

- Fixed parameters
  - · Problem size $n = 2$.
  - · Maximum number of iterations of Newton's method: $maxiter = 100$.
  - · Relative tolerance: $tol = 10^{-12}$.
- Variable parameters
  - · Stiffness: $\epsilon = [0.00001, 0.001, 0.1]$.
  - · Final time: $t_f = [1, 5, 10, 15, 20, 30]$.
  - · Step size: $\Delta t = [0.1, 0.05, 0.01, 0.005, 0.001]$.
  - · Order of Adams–Moulton or Adams–Bashforth method: $r = [1, 2, 3]$.

Again, the ϵ parameter has special relevance on the test determining the stiffness of the problem.

Figs. 8 and 9 and Tables 7 and 8 show execution times and relative errors when $\epsilon = 0.1$, $\Delta t = 0.01$, $r = 1$ and final time is varied, $t_f = 1, 5, 10, 15, 20, 30$, and when $\epsilon = 0.1$, $t_f = 5$, $r = 1$ and $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

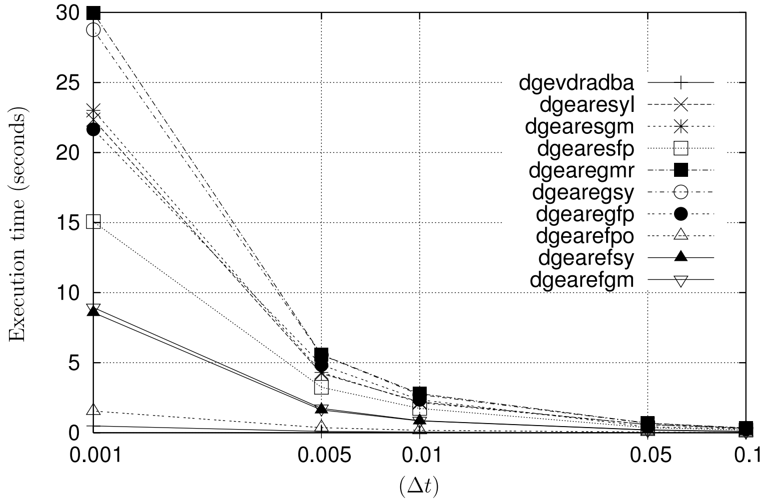

Fig. 9. Execution time (seconds) considering $\epsilon = 0.1$, $t_f = 5$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$ for Case Study 4.

Table 8
Table reporting relative error for the solution of DMRE for Case Study 5 considering $\epsilon = 0.1$, $t_f = 5$, $r = 1$ and varying $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Δt	0.1	0.05	0.01	0.005	0.001
dgevdradba	3.8317e−14	2.1812e−17	2.5966e−16	5.7635e−16	2.5447e−15
dgevdradmo	2.9970e−17	1.7417e−16	4.0728e−16	1.0256e−15	4.7658e−15

5.3. Case Study 5

This time-varying DMRE [22] is defined as

$$\dot{X}(t) = -X(t)T_{2^k}(t) + T_{2^k}(t)X(t) - b(t)X^2(t) - b(t)I_{2^k}, \qquad X(0) = I_{2^k},$$

where $X(t) \in \mathbb{R}^{2^k \times 2^k}$, and $T_{2^k} \in \mathbb{R}^{2^k \times 2^k}$ are generated recursively as follows

$$T_2 = \begin{bmatrix} a(t) & b(t) \\ -b(t) & a(t) \end{bmatrix},$$
$$T_{2^k} = T_2 \otimes I_{2^{k-1}} + I_2 \otimes T_{2^{k-1}}, \quad k \geq 2,$$

where $a(t) = \cos t$ and $b(t) = \sin t$. The analytic solution is

$$X(t) = \frac{1 + \tan(\cos t - 1)}{1 - \tan(\cos t - 1)} I_{2^k}.$$

For this case study the characteristic parameters are the following:

- Fixed parameters
  - · Problem size: $n = m = 16$ ($k = 4$).
  - · Maximum number of iterations of Newton's method: $maxiter = 100$.
  - · Relative tolerance: $tol = 10^{-12}$.
- Variable parameters
  - · Final time: $t_f = [1, 5, 10, 15, 20, 30]$.
  - · Step size: $\Delta t = [0.1, 0.05, 0.01, 0.005, 0.001]$.
  - · Order of Adams–Moulton or Adams–Bashforth methods: $r = [1, 2, 3]$.

Figs. 10 and 11 and Tables 9 and 10 show execution times and relative errors, considering $r = 2$ for Adams–Bashforth algorithm and $r = 3$ for Adams–Moulton algorithms, when $\Delta t = 0.01$ and varying $t_f = 1, 5, 10, 15, 20, 30$, and when $t_f = 1$ and varying step size $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

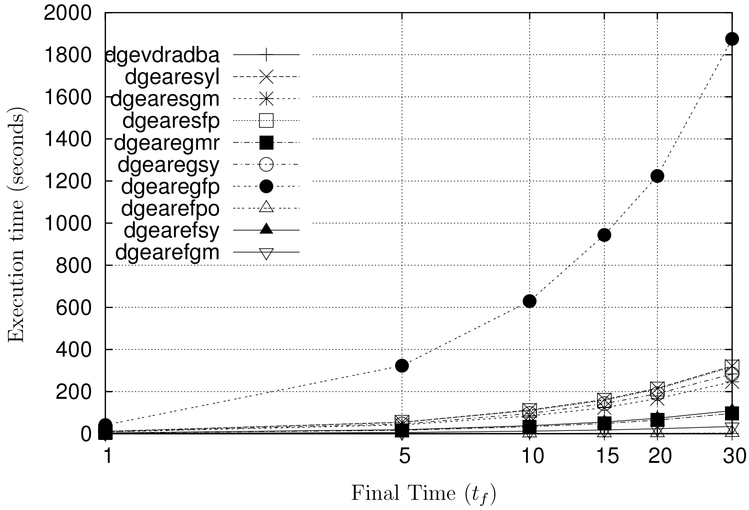

Fig. 10. Execution time (seconds) considering stiffness equal to 1, $\Delta t = 0.01$ and $r = 2$ for Adams–Bashforth and $r = 3$ for Adams–Moulton and varying $t_f = 1, 5, 10, 15, 20, 30$ for Case Study 5.

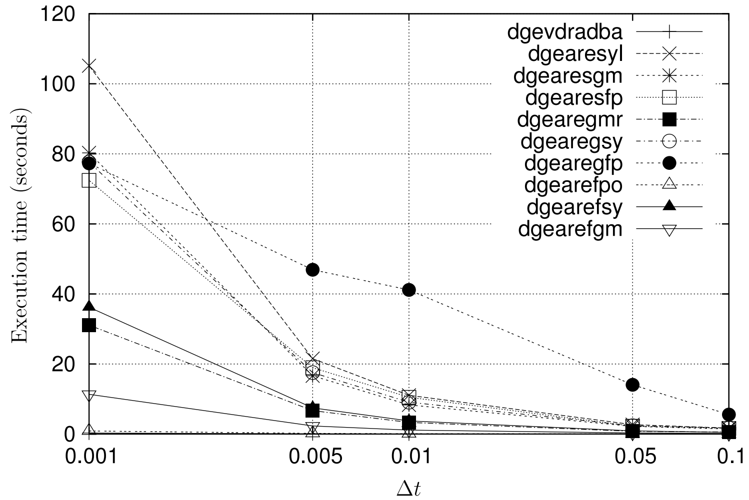

Fig. 11. Execution time (seconds) considering $t = 1$, $r = 2$ for Adams–Bashforth and $r = 3$ for Adams–Moulton algorithms and varying the step size $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Table 9
Table reporting relative error for the solution of DMRE for Case Study 5 considering a stiffness value equal to 0.1, $\Delta t = 0.01$, and $r = 2$ for Adams–Bashforth and $r = 3$ for Adams–Moulton, and varying $t_f = 1, 5, 10, 15, 20, 30$.

t_f	1	5	10	15	20	30
dgevdradba	1.8917e−5	5.6949e−5	1.0120e−4	1.0958e−4	9.9030e−6	1.8441e−4
dgevdradmo	1.1158e−8	4.2116e−8	2.0980e−8	1.6965e−8	1.8950e−8	7.0512e−8

Table 10
Table reporting relative error for the solution of the DMRE for Case Study 5 considering $t = 1$, $r = 2$ for Adams–Bashforth and $r = 3$ for Adams–Moulton algorithms and varying the step size $\Delta t = 0.1, 0.05, 0.01, 0.005, 0.001$.

Δt	0.1	0.05	0.01	0.005	0.001
dgevdradba	6.8760e−4	1.8917e−5	9.6683e−7	3.0113e−7	1.3984e−8
dgevdradmo	2.2853e−4	7.5868e−6	1.1158e−8	6.9029e−10	1.0933e−12

6. Conclusions and future work

In this paper several algorithms to solve differential Riccati equations based on Adams–Bashforth and Adams–Moulton methods are described and implemented in MATLAB. Several conclusions can be extracted:

- If the DMRE is not stiff, the best algorithm in terms of execution time is the Adams–Bashforth implementation. If the DMRE is stiff, the best algorithms are the Adams–Moulton implementations.
- In terms of accuracy of the approach, all of them provide good results if the stiffness is low. However, as the stiffness increases it is necessary to decrease the step size to keep the good behaviour of the implementations.
- The best results have been obtained considering that the parameter r has a value of 1.
- According to the Adams–Moulton algorithms, the implementation based on the fixed-point method gives the lowest execution time. Furthermore, the combinations of fixed-point with another algorithm (Sylvester or GMRES algorithms) also provides good execution time.
- Continuing with the Adams–Moulton implementation, in general, the worst execution time is provided by the Sylvester algorithm and combinations.

There are two items for future work:

- First, adaptive algorithms will be developed.
- Second, the algorithms presented in this work will be implemented in parallel in a distributed memory platform, using the message passing paradigm; MPI [23] and BLACS [24] for communications, and PBLAS [25] and ScaLAPACK [26] for computations.

References

[1] Luca Dieci, Numerical integration of the differential Riccati equation and some related issues, SIAM J. Numer. Anal. 29 (3) (1992) 781–815.
[2] D.R. Vaughan, A negative exponential solution for the matrix Riccati equation, IEEE Trans. Automat. Control 14 (1) (1969) 72–75.
[3] Eduard J. Davison, M.C. Maki, The numerical solution of the matrix Riccati differential equation, IEEE Trans. Automat. Control 18 (1) (1973) 71–73.
[4] C.S. Kenney, R.B. Leipnik, Numerical integration of the differential matrix Riccati equation, IEEE Trans. Automat. Control 30 (1985) 962–970.
[5] H. Chandrasekhar, Generalized Chandrasekhar algorithms: time-varying models, IEEE Trans. Automat. Control 21 (1976) 728–732.
[6] D.W. Rand, P. Winternitz, Nonlinear superposition principles: a new numerical method for solving matrix Riccati equations, Comput. Phys. Comm. 33 (1984) 305–328.
[7] M. Sorine, P. Winternitz, Superposition laws for the solution of differential Riccati equations, IEEE Trans. Automat. Control 30 (1985) 266–272.
[8] Chang Ho Choi, Efficient algorithms for solving stiff matrix-valued Riccati differential equations, Ph.D. Thesis, University of California, 1988.
[9] Peter Benner, Hermann Mena, BDF methods for large-scale differential Riccati equations, in: Sixteenth International Symposium on Mathematical Theory of Network and Systems, MTNS2004, Belgium, Katholieke Universiteit Leuven, 2004.
[10] E. Arias, V. Hernández, J. Ibáñez, J. Peinado, A fixed point-based BDF method for solving Riccati equations, Appl. Math. Comput. 188 (2) (2007) 1319–1333.
[11] V. Hernández, J. Ibáñez, E. Arias, J. Peinado, A GMRES-based BDF method for solving differential Riccati equations, Appl. Math. Comput. 196 (2) (2008) 613–626.
[12] J.M. Sanz-Serna, Symplectic integrators for Hamiltonian problems: an overview, Acta Numer. 1 (1992) 243–286.
[13] Li Ren-Cang, Unconventional reflexive numerical methods for matrix differential Riccati, Technical Report 2000-36, Department of Mathematics, University of Kentucky, 2000.
[14] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations, SIAM, Philadelphia, 1998.
[15] D.H. Griffel, Applied Functional Analysis, Mathematics and its Applications, Ellis Hordwood, New York, 1981.
[16] R.H. Bartels, G.W. Stewart, Solution of the matrix equation $AX + XB = C$: algorithm 432, Commun. ACM 15 (9) (1972) 820–826.
[17] Roger A. Horn, Charles R. Johnson, Topics in Matrix Analysis, Cambridge University Press, London, 1991.
[18] S. Pruess, Interpolation schemes for collocation solution of TPBVPs, SIAM J. Sci. Stat. Comput. 7 (1986) 322–333.
[19] G.H. Meyer, Initial Value Methods for Boundary Value Problems, Academic Press, New York, 1973.
[20] Chang Ho Choi, Alan J. Laub, Efficient matrix-valued algorithms for solving stiff Riccati differential equations, IEEE Trans. Automat. Control 35 (7) (1990) 770–776.
[21] David L. Brown, Jens Lorenz, A high-order method for stiff boundary value problems with turning points, SIAM J. Sci. Stat. Comput. 8 (1987) 790–805.
[22] Chang Ho Choi, Time-varying Riccati differential equations with known analytic solutions, IEEE Trans. Automat. Control 37 (1992) 642–645.
[23] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, MIT Press, 1994.
[24] J.J. Dongarra, R.A. Van De Geijn, Two dimensional basic linear algebra communications subprograms, Technical Report UT-CS-91-138, Department of Computer Science, University of Tenesee, 1991.
[25] J. Choi, J.J. Dongarra, S. Ostrouchov, A. Petitet, D. Walker, A proposal for a set of parallel basic linear algebra subprograms, Technical Report UT-CS-95-292, Department of Computer Science, University of Tenesee, 1995.
[26] L.S. Blackford, et al., ScaLAPACK Users' Guide, SIAM, 1997.