Informe Técnico / Technical Report

Ref. No: DSIC-II/08/09          Pages: 30
Title: A Family of BDF Methods for Solving Differential Matrix Riccati Equations Using Adaptative Techniques
Author(s): Javier Ibañez, Jesús Peinado, Vicente Hernández, Enrique Arias
Date: 17 – November - 2009
Key Words: Differential Matrix Riccati Equations (DMRE), BDF Methods, Algebraic Matrix Riccati Equations (AMRE), Algebraic Matrix Sylvester Equations (AMSE), Fixed-Point Method

VºBº
Leader of Reasearch Group

Vicente Hernández
A Family of BDF Methods for Solving Differential Matrix Riccati Equations Using Adaptive Techniques

Javier Ibáñez, Jesús Peinado, Vicente Hernández, Enrique Arias

November 16, 2009

Abstract

Differential Matrix Riccati Equations play a fundamental role in control theory, for example, in optimal control, filtering and estimation, decoupling and order reduction, etc. One of the most popular codes to solve stiff Differential Matrix Riccati Equations (DMREs) are based on Backward Differentiation Formula (BDF) such as the DRESOL package by Luca Dieci. In previous papers the authors of this paper developed three algorithms for solving DMREs based on BDF methods: a Sylvester algorithm, an iterative Generalized Minimum RESidual (GMRES) algorithm and a Fixed-Point algorithm.

In this technical report we present two contributions to improve the above algorithms. Firstly six variants of previous algorithms are carried out by using one of above algorithms in the first step and another algorithm to carry out the other steps until reaching convergence. Numerous tests on six case studies have been done comparing both precision and computational costs of MATLAB implementations of the above algorithms. Experimental results show that in some cases these algorithms improve on the speed and convergence of the original algorithms. Secondly, using the previous experimental results and since all algorithms have a similar structure and there is no best algorithm to solve all problems, two general-purpose adaptive MATLAB implementation have been designed for selecting the most appropriate algorithm, which can be chosen using a parameter that indicates the stiffness of the DMRE to be solved.

Actually MATLAB does not have built-in functions for solving DMREs. The developed codes is on-line to be downloaded.
1 Introduction

In this paper we consider DMREs of the form

$$\dot{X}(t) = A_{21}(t) + A_{22}(t)X(t) - X(t)A_{11}(t) - X(t)A_{12}(t)X(t), \quad (1)$$

$$t_0 \leq t \leq t_f,$$

$$X(t_0) = X_0 \in \mathbb{R}^{m \times n},$$

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

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

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 value problem by decoupling this problem in two initial value problems [15]. Since the mid seventies, many different methods have been proposed: linearization approach [28, 14, 21], Chandrasekhar approach [8], superposition methods [24, 27], BDF approach [15, 9, 5, 1, 19], Hamiltonian approach [26], unconventional reflexive numerical methods [25], etc.

The following definitions and properties will be used in this paper:

1. The vec operator [20, p. 244] is a mapping

   $$\text{vec} : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{mn}$$

   defined by

   $$\text{vec}(A) = [a_{11}, a_{21}, \cdots, a_{m1}, a_{12}, a_{22}, \cdots, a_{m2}, \cdots, a_{1n}, a_{2n}, \cdots, a_{mn}]^T,$$

   where $A = [a_{ij}] \in \mathbb{R}^{m \times n}$.

2. The mat$_{m \times n}$ operator ($m, n \in \mathbb{N}$) is a mapping

   $$\text{mat}_{m \times n} : \mathbb{R}^{mn} \longrightarrow \mathbb{R}^{m \times n}$$

   defined by

   $$\text{mat}_{m \times n}(v) = \begin{bmatrix}
   v_1 & v_{m+1} & \cdots & v_{mn-m+1} \\
   v_2 & v_{m+2} & \cdots & v_{mn-m+2} \\
   \vdots & \vdots & \ddots & \vdots \\
   v_m & v_{2m} & \cdots & v_{mn}
   \end{bmatrix},$$

   where $v = [v_i] \in \mathbb{R}^{mn}$.
3. The Kronecker product of $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ [20, p. 243] is denoted by $A \otimes B$ and is defined as the block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}Ba_{m2}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp \times nq}.$$ 

This paper is organized as follows. First, Section 2 describes a family of BDF algorithms for solving DMREs based on three approaches: matrix Sylvester equations [11, 15], GMRES method [19] and Fixed-Point iteration [1]. Section 3 shows all developed algorithms. In Section 4, six case studies and the results obtained are presented. From experimental results, an adaptive algorithm is explained in Section 4.2. Finally, the conclusions and future work are outlined in Section 5.

2 A family of BDF methods

In this section we will describe a family of BDF methods for solving DMREs. In a BDF 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, in the context of stiff DMREs, one of the better choices for solving the associated AMRE is to apply implicit schemes based on Newton’s or quasi-Newton methods. 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 partition $\{t_0,t_1,t_2,\ldots,t_f\}$ of interval $[t_0,t_f]$ and we apply a BDF scheme, then the approximate solution $X_k$ at $t_k$ is obtained by means of solving the following matrix equation

$$-X_k + \sum_{j=1}^{r} \alpha_j X_{k-j} + \Delta t_{k-1} \beta F(t_k,X_k) = 0, \quad (3)$$

where $\Delta t_{k-1} = t_k - t_{k-1}$, and $\alpha_j$ ($j = 1,2,\ldots,r$) and $\beta$ are values that appear in Table 1, being $r$ the order of BDF method. Equation (3) can be expressed as the AMRE

$$\tilde{A}_{21} + \tilde{A}_{22}X_k + X_k \tilde{A}_{11} + X_k \tilde{A}_{12}X_k = 0, \quad (4)$$

where

$$\tilde{A}_{21} = -\beta \Delta t_{k-1} A_{21}(t_k) - \sum_{j=1}^{r} \alpha_j X_{k-j},$$

$$\tilde{A}_{22} = -\beta \Delta t_{k-1} A_{22}(t_k) + I_m,$$

$$\tilde{A}_{11} = \beta \Delta t_{k-1} A_{11}(t_k),$$

$$\tilde{A}_{12} = \beta \Delta t_{k-1} A_{12}(t_k).$$
Table 1: Parameters of BDF method (order $r=1, 2, 3, 4$ and 5)

<table>
<thead>
<tr>
<th>$r$</th>
<th>$\beta$</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>$\alpha_4$</th>
<th>$\alpha_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2/3</td>
<td>4/3</td>
<td>-1/3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6/11</td>
<td>18/11</td>
<td>-9/11</td>
<td>2/11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>60/137</td>
<td>300/137</td>
<td>-300/137</td>
<td>200/137</td>
<td>-75/137</td>
<td>12/137</td>
</tr>
</tbody>
</table>

2.1 A Sylvester method

Equation (4) can be solved by the Newton’s iteration

$$\hat{G}'_{(l)}(X_{k}^l - X_{k}^{l-1}) = -G(X_{k}^{l-1}), l \geq 1,$$

where $\hat{G}'_{(l)}$ is the Fréchet derivative \[17, p. 310\] 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:

$$C_{22}^{l-1}\Delta X_{k}^{l-1} + \Delta X_{k}^{l-1}C_{11}^{l-1} = C_{21}^{l-1},$$

where

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

and $\Delta X_{k}^{l-1} = X_{k}^l - X_{k}^{l-1}$.

Therefore $X_{k}^l$ can be obtained by solving (6) for $\Delta X_{k}^{l-1}$ 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 \[4\].

2.2 A GMRES method

This method was developed in \[19\]. If we apply the vec operator to (6), then

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

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 the GMRES method, and then

$$X_{k}^l = X_{k}^{l-1} + \text{mat}(\Delta x_{k}^{l-1}, m, n).$$
2.3 A Fixed-Point method

This method has been developed in [1]. 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 to define the following Fixed-Point iteration

\[
(\bar{A}_{22} + X_{l-1} \bar{A}_{12})X_{l-1}^k = -(\bar{A}_{21} + X_{l-1} \bar{A}_{11}), l \geq 1, \quad X_0^k = X_{k-1}.
\]  \hspace{1cm} (7)

Similarly, from (4) we obtain the Fixed-Point iteration

\[
X_{l-1}^k (\bar{A}_{11} + \bar{A}_{12}X_{l-1}^{k-1}) = -(\bar{A}_{21} + A_{22}X_{l-1}^{k-1}), l \geq 1, \quad X_0^k = X_{k-1}.
\]  \hspace{1cm} (8)

2.4 Other BDF approaches

This is the one of the contributions of this paper. It is possible modify these methods to improve speed and convergence. The idea consists of combining two BDF methods applying a first step using a 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-Points), 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.

3 A family of BDF algorithms for solving DM-REs

Another contribution of this paper is a general algorithm scheme which allows to choose ”the best algorithm” for each DMRE problem. With this scheme up to
nine different BDF algorithms can be applied. First of all, a general algorithm nomenclature is described. Later, some of the implementations (algorithms) are shown.

There are two kinds of algorithms: driver algorithms and computational algorithms. Driver algorithms solve DMREs by using BDF methods. Computational algorithms are single or combined algorithms to solve AMREs (Subsection 2.4).

3.1 A nomenclature to describe the algorithms

In this subsection two driver algorithms, and nine single and one combined computational algorithms are described.

All the algorithms, driver algorithms and computational algorithms, have names of the form \texttt{XYYZZZVVV} where:

- \texttt{X} indicates data type, i.e. letter \texttt{d} indicates double precision.
- \texttt{YY} indicates the type of matrix, i.e \texttt{ge} indicates general matrix.
- \texttt{ZZZ} indicates the problem solved:
  - \texttt{vdr}: time-varying differential matrix Riccati equation.
  - \texttt{idr}: time-invariant differential matrix Riccati equation.
  - \texttt{are}: algebraic matrix Riccati equation.
- \texttt{VVV} indicates the algorithm used to solve the problem.
  - For driver algorithms, \texttt{VVV} is equal to \texttt{bdf} because a BDF algorithm is used.
  - For computational algorithms, it depends whether the algorithm used is single or combined. If the algorithm is single, the algorithm name is indicated, \texttt{syl} (for Sylvester), \texttt{fpo} (for Fixed-Point) and \texttt{gmr} (for GMRES). If the algorithm is combined, first letter indicates the algorithm used and second and third, the kind of approximation, e.g. \texttt{gfp} (GMRES algorithm with a Fixed-Point approach, see Algorithm 1).

3.2 Algorithms

All algorithms explained before have been implemented, but only some of them are shown (for space reasons). Therefore there is a driver algorithm for the time-invariant case \texttt{dgeidrbdf} (not shown) and another for the time-varying case (Algorithm 1) \texttt{dgevdrbdf}. This algorithm solves a time-varying DMRE by calling other algorithms to solve AMREs. There are nine algorithms to solve AMREs (see Figure 1), but only four of them, (Algorithms 2, 3, 4, and 5) are shown.

The computational algorithms correspond to the three single approaches: \texttt{dgearesyl} for Sylvester (Algorithm 2), \texttt{dgearegmr} for GMRES (Algorithm 3)
Figure 1: Implemented algorithms and their inter-dependencies
Algorithm 1 Solves DMREs by means of a BDF algorithm

Function \( \{X_k\}_{k=1}^p = \text{dgevdrbdf}(\text{alg}, \text{data}, t_0, X_0, \Delta t, tol, \text{maxiter}) \)

Inputs: \( \text{alg} \) is the algorithm used to solve the AMRE associated to DMRE (Figure 1); \( \text{data}(t) \) is the function that computes the coefficient matrices of (1) at instant \( t \); initial time \( t_0 \in \mathbb{R} \); starting guess matrix \( X_0 \in \mathbb{R}^{m \times n} \); final time \( t_f \in \mathbb{R} \); step size \( \Delta t \in \mathbb{R} \); order of BDF method \( r \in \{1, 2, 3, 4, 5\} \); \( tol \in \mathbb{R}^+ \) is the tolerance used in BDF method; \( \text{maxiter} \in \mathbb{N} \) is the maximum number of Newton iterations in BDF method.

Outputs: Matrices \( X_k \in \mathbb{R}^{m \times n}, k = 1, 2, \ldots, p \) \( (p = \lceil t_f - t_0 \rceil / \Delta t) \).

1: Initialize \( \alpha \) and \( \beta \) with the values given in Table 1
2: \( t = t_0 \)
3: \( p = \lceil t_f - t_0 \rceil / \Delta t \)
4: for \( k = 1 \) to \( p \) do
5: \( s = \min(r, k) \)
6: \( [A_{11}, A_{12}, A_{21}, A_{22}] = \text{data}(t) \)
7: \( \tilde{A}_{21} = -\beta_s \Delta t A_{21} - \sum_{j=1}^{s} \alpha_j X_{k-j} \)
8: \( \tilde{A}_{22} = -\beta_s \Delta t A_{22} + I_m \)
9: \( \tilde{A}_{11} = \beta_s \Delta t A_{11} \)
10: \( \tilde{A}_{12} = \beta_s \Delta t A_{12} \)
11: \( [X_k, e] = \text{alg}(\tilde{A}_{11}, \tilde{A}_{12}, \tilde{A}_{21}, \tilde{A}_{22}, X_{k-1}, tol, \text{maxiter}) \)
12: \( t = t + \Delta t \)
13: end for
14: if \( e = -1 \) then
15: error (‘there is no convergence’)
16: end if
dgearefpo for Fixed-Point (Algorithm 4), and a combined algorithm dgearegfp for GMRES with a initial Fixed-Point iteration (Algorithm 5). The other combined approaches are not shown.

Algorithm 2 Solves the AMRE (4) using the Newton-Sylvester algorithm

Function $[X_k, l] = $dgearesyl($\bar{A}_{11}, \bar{A}_{12}, \bar{A}_{21}, \bar{A}_{22}, X_{k-1}, tol, maxiter$)

Inputs: Coefficient matrices of AMRE (4) $\bar{A}_{11}, \bar{A}_{12}, \bar{A}_{21}, \bar{A}_{22}$; 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’s iterations used to reach convergence, or $-1$ if convergence is not reached.

1: $X_k = X_{k-1}$
2: $l = 1$
3: while $l \leq maxiter$ do
4: $C_{22} = A_{22} + X_k A_{12}$
5: $C_{11} = A_{11} + A_{12} X_k$
6: $C_{21} = -A_{21} - A_{22} X_k - X_k C_{11}$
7: Solve the AMSE $C_{22} \Delta X_k + \Delta X_k C_{11} = C_{21}$ for $\Delta X_k$
8: $X_k = X_k + \Delta X_k$
9: if $\|\Delta X_k\|_\infty < tol$ then
10: break
11: end if
12: $l = l + 1$
13: end while
14: if $l > maxiter$ then
15: $l = -1$
16: end if

4 Experimental Results

The main objective of this section is to compare the developed algorithms. The implementations have been tested on an Apple Macintosh iMac 2.16 Ghz Core 2 Duo processor with 2 Gb of RAM, MacOsX(Unix) OS and MATLAB 7.4.

4.1 Comparative between implemented algorithms

As test battery, six case studies have been used: three cases that correspond to time-invariant DMREs (cases 1, 2 and 3), and three cases that correspond to time-varying DMREs (cases 4, 5 and 6).

For each case study, the characteristic parameters which offer better accuracy (less error), and lower computational cost have been determined.

Two kinds of tests have been carried out:

1. To fix the size of problem and the final time ($t_f$), varying the step size
Algorithm 3 Solves the AMRE (4) using the Newton-GMRES algorithm

Function \([X_k, l] = \text{dgearegmr}(\bar{A}_{11}, \bar{A}_{12}, \bar{A}_{21}, \bar{A}_{22}, X_{k-1}, \text{tol}, \text{maxiter})\)

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

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 \(l \leq \text{maxiter}\) do
4: \(C_{22} = \bar{A}_{22} + X_k \bar{A}_{12}\)
5: \(C_{11} = \bar{A}_{11} + \bar{A}_{12}X_k\)
6: \(C_{21} = -\bar{A}_{21} - \bar{A}_{22}X_k - X_k C_{11}\)
7: Solve \(\begin{bmatrix} I_n \otimes C_{22} + (C_{11})^T \otimes I_m \end{bmatrix} \Delta x_k = \text{vec}(C_{21})\) for \(\Delta x_k\) by using a GMRES method
8: \(X_k = X_k + \text{mat}(\Delta x_k, m, n)\)
9: if \(\|\Delta x_k\|_\infty < \text{tol}\) then
10: Leave while loop
11: end if
12: \(l = l + 1\)
13: end while
14: if \(l > \text{maxiter}\) then
15: \(l = -1\)
16: end if
Algorithm 4 Solves the AMRE (4) using the Newton-Fixed-Point algorithm

Function \([X_k, l] = dgearefpo(\bar{A}_{11}, \bar{A}_{12}, \bar{A}_{21}, \bar{A}_{22}, X_{k-1}, \text{tol}, \text{maxiter})\)

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

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 \(l \leq \text{maxiter}\) do
4: \(C_{22} = \bar{A}_{22} + X_k \bar{A}_{12}\)
5: \(C_{21} = -(\bar{A}_{21} + X_k \bar{A}_{11})\)
6: Solve the linear system \(C_{22}X = C_{21}\) for \(X\)
7: \(\text{norm} = \|X - X_k\|_\infty\)
8: \(X_k = X\)
9: if \(\text{norm} < \text{tol}\) then
10: Leave the while loop
11: end if
12: \(l = l + 1\)
13: end while
14: if \(l > \text{maxiter}\) then
15: \(l = -1\)
16: end if
Algorithm 5 Solves the AMRE (4) using the Newton-GMRES algorithm with a Fixed-Point initial iteration

**Function** \( [X_k, l] = \text{dgearegfp}(\bar{A}_{11}, \bar{A}_{12}, \bar{A}_{21}, \bar{A}_{22}, X_{k-1}, \text{tol}, \text{maxiter}) \)

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

**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: \( C_{22} = \bar{A}_{22} + X_k \bar{A}_{12} \)
3: \( C_{21} = -(\bar{A}_{21} + X_k \bar{A}_{11}) \)
4: Solve \( C_{22} X_k = C_{21} \) for \( X_k \)
5: \( l = 1 \)
6: **while** \( l \leq \text{maxiter} \) **do**
   7: \( C_{22} = \bar{A}_{22} + X_k \bar{A}_{12} \)
   8: \( C_{11} = \bar{A}_{11} + \bar{A}_{12} X_k \)
   9: \( C_{21} = -\bar{A}_{21} - \bar{A}_{22} X_k - X_k C_{11} \)
10: Solve \( [I_n \otimes C_{22} - (C_{11})^T \otimes I_m] \Delta x_k = \text{vec}(C_{21}) \) for \( \Delta x_k \)
    by using a GMRES method
11: \( X_k = X_k + \text{mat}(\Delta x_k, m, n) \)
12: **if** \( \|\Delta x_k\|_\infty < \text{tol} \) **then**
13: Leave while loop
14: **end if**
15: \( l = l + 1 \)
16: **end while**
17: **if** \( l > \text{maxiter} \) **then**
18: \( l = -1 \)
19: **end if**
(\Delta t). This kind of test has been done in case studies where changing \((t_f)\) improves the behaviour of the results.

2. To fix the problem size and the step size \((\Delta t)\), varying the final time \((t_f)\). This kind of test has been done in case studies where changing \((\Delta t)\) improves the behaviour of the results.

In problems where it is possible to vary the size of problem, a representative size has been chosen.

All algorithms explained before have been tested, but only the ones with the best results are shown:

1. Sylvester algorithm \((\text{dgearesyl})\).
2. GMRES algorithm \((\text{dgearegmr})\).
3. Fixed-Point algorithm \((\text{dgearefpo})\).
4. Sylvester algorithm with an initial Fixed-Point iteration \((\text{dgearesfp})\).
5. GMRES algorithm with an initial Fixed-Point iteration \((\text{dgearegfp})\).

The driver algorithms developed are:

- \text{dgeidrbdf} (not shown) solves time-invariant DMREs by means of a BDF method.
- \text{dgevdrbdf} (Algorithm 1) solves time-varying DMREs by means of a BDF method.

For both algorithms the following parameters have been used:

- \text{alg}: algorithm chosen for solving the AMRE. We used all algorithms explained in subsection 2.4., but the results of the combined algorithms \text{dgearefsy}, \text{dgearefgm}, \text{dgearefsy} and \text{dgearesgm} are not shown because there is no improvement in execution time or accuracy.
- \text{data}: function to compute the coefficient matrices of (1).
- \text{t0}: initial time.
- \text{X0}: starting guess.
- \text{\Delta t}: step size.
- \text{r}: order of BDF method.
- \text{tol}: tolerance used in Newton’s method.
- \text{maxiter}: maximum number of iterations used in Newton’s method. For all cases \text{maxiter} = 100.
These parameters have been adjusted in each case study to get the best execution time \((T_e)\) and 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 and 6) or an approximate solution (Case studies 1, 4 and 5).

These tests allow us to extract some conclusions for each case study.

**Case study 1.** The first time-invariant DMRE is a non stiff case taken from a two-point boundary value problem in [23]. This DMRE is defined for \(t \geq 0\) by means of the coefficient matrices

\[
\begin{align*}
A_{11} &= \begin{bmatrix} 0 & 0 \\ -100 & -1 \end{bmatrix}, & A_{12} &= \begin{bmatrix} 0 & 1 \\ 100 & 0 \end{bmatrix}, \\
A_{21} &= \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}, & A_{22} &= \begin{bmatrix} 0 & 0 \\ -10 & -1 \end{bmatrix},
\end{align*}
\]

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

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

The following parameters were used for this problem:

- \(r = 2, \ tol = 1e^{-6}.\)
- Size of problem \(n = 2\) (fixed by problem).

As explained before we carried out two kinds of tests:

1. Fixing the final time \((t_f) = 2\) seconds and varying the step size \((\Delta t) = 0.1, \ 0.05, \ 0.01, \ 0.005\) and \(0.001\) seconds. Tables 2 and 3 show these tests.
2. Fixing the step size \((\Delta t) = 0.01\), varying the final time \((t_f) = 2, 5, 10, 15, \ 20, \) and \(30\) seconds. Figure 2 and Table 4 show these tests.

Conclusions for case study 1:

- Fixed-Point algorithm is much faster than other algorithms.
- Relative error is similar for all the algorithms.
Table 2: Case Study 1: Execution time when $t_f=2$, $n=m=2$ and $\Delta t$ variable ($r=2$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th>$T_e$</th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>0.07</td>
<td>0.09</td>
<td>0.41</td>
<td>0.81</td>
<td>2.81</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.07</td>
<td>0.09</td>
<td>0.43</td>
<td>0.87</td>
<td>2.73</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.04</td>
<td>0.20</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.05</td>
<td>0.09</td>
<td>0.26</td>
<td>0.49</td>
<td>2.27</td>
</tr>
</tbody>
</table>

Table 3: Case Study 1: Relative error when $t_f=2$, $n=m=2$ and $\Delta t$ variable ($r=2$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th>$Er$</th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>7.41e-01</td>
<td>2.69e-02</td>
<td>1.42e-03</td>
<td>3.64e-04</td>
<td>1.49e-05</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>7.41e-01</td>
<td>2.69e-02</td>
<td>1.42e-03</td>
<td>3.64e-04</td>
<td>1.49e-05</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>7.41e-01</td>
<td>2.69e-02</td>
<td>1.42e-03</td>
<td>3.64e-04</td>
<td>1.49e-05</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>7.41e-01</td>
<td>2.69e-02</td>
<td>1.42e-03</td>
<td>3.64e-04</td>
<td>4.85e-15</td>
</tr>
</tbody>
</table>

Table 4: Case Study 1: Relative error when $\Delta t=0.01$, $n=m=2$ and $t_f$ variable ($r=2$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th>$Er$</th>
<th>$t_f=2$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.42e-03</td>
<td>7.02e-05</td>
<td>4.73e-07</td>
<td>3.18e-09</td>
<td>2.16e-11</td>
<td>1.51e-15</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>1.42e-03</td>
<td>7.02e-05</td>
<td>4.73e-07</td>
<td>3.18e-09</td>
<td>2.16e-11</td>
<td>1.74e-15</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>1.42e-03</td>
<td>7.02e-05</td>
<td>4.73e-07</td>
<td>3.18e-09</td>
<td>2.16e-11</td>
<td>1.84e-15</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>1.42e-03</td>
<td>7.02e-05</td>
<td>4.73e-07</td>
<td>3.18e-09</td>
<td>2.16e-11</td>
<td>1.48e-15</td>
</tr>
</tbody>
</table>

Case study 2. The second case study [22, 15] consists of the following time-invariant DMRE

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

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

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.

The following parameters were used for this problem:
Figure 2: Case Study 1: Execution time when $\Delta t=0.01$, $n = m=2$ and $t_f$ variable

$(r=2, tol=1e-6, maxiter=100)$

Table 5: Case Study 2: Execution time when $t_f=1$, $n = m=20$ and $\Delta t$ variable

$(r=1, tol=1e-5, maxiter=100)$

<table>
<thead>
<tr>
<th>$Te$</th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.89</td>
<td>2.73</td>
<td>9.88</td>
<td>18.74</td>
<td>88.79</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.18</td>
<td>0.26</td>
<td>0.96</td>
<td>1.83</td>
<td>8.60</td>
</tr>
<tr>
<td>dgearefpo</td>
<td>No Conv.</td>
<td>0.03</td>
<td>0.03</td>
<td>0.05</td>
<td>0.24</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>1.81</td>
<td>2.64</td>
<td>9.62</td>
<td>18.35</td>
<td>86.21</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.18</td>
<td>0.26</td>
<td>0.95</td>
<td>1.86</td>
<td>8.48</td>
</tr>
</tbody>
</table>

- $r = 1$, $tol = 1e - 5$.
- Size of problem $n = 20$.
- $\alpha = 100$ (stiff case).

We carried out two kinds of test:

1. Fixing the final time ($t_f$) = 1 second and varying the step size ($\Delta t$) = 0.1, 0.05, 0.01, 0.005 and 0.001 seconds. Tables 5 and 6 show these tests.

2. Varying the final time ($t_f$) = 1, 5, 10, 15, 20, and 30 seconds, with a step size ($\Delta t$) = 0.01 for algorithm $dgearefpo$ and ($\Delta t$) = 0.1 for the other algorithms. Tables 7 and 8 show these tests.

Conclusions for case study 2:
Table 6: Case Study 2: Relative error when \( t_f = 1 \), \( n = m = 20 \) and \( \Delta t \) variable 
\( (r=1, tol=1e-5, maxiter=100) \)

<table>
<thead>
<tr>
<th>( Er )</th>
<th>( \Delta t = 0.1 )</th>
<th>( \Delta t = 0.05 )</th>
<th>( \Delta t = 0.01 )</th>
<th>( \Delta t = 0.005 )</th>
<th>( \Delta t = 0.001 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>2.22e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.33e-16</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>1.11e-16</td>
<td>0</td>
<td>0</td>
<td>3.33e-16</td>
<td>0</td>
</tr>
<tr>
<td>dgearefpo</td>
<td>No Conv.</td>
<td>3.67e-09</td>
<td>0</td>
<td>0</td>
<td>2.22e-16</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>1.11e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.33e-16</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>1.11e-16</td>
<td>2.22e-16</td>
<td>0</td>
<td>0</td>
<td>8.88e-16</td>
</tr>
</tbody>
</table>

Table 7: Case Study 2: Execution time when \( n = m = 20 \) and \( t_f \) variable 
\( (r=1, tol=1e-5, maxiter=100) \)

<table>
<thead>
<tr>
<th>( Te )</th>
<th>( t_f = 1 )</th>
<th>( t_f = 5 )</th>
<th>( t_f = 10 )</th>
<th>( t_f = 15 )</th>
<th>( t_f = 20 )</th>
<th>( t_f = 30 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl(( \Delta t = 0.1 ))</td>
<td>1.91</td>
<td>5.2</td>
<td>9.32</td>
<td>13.45</td>
<td>17.48</td>
<td>25.77</td>
</tr>
<tr>
<td>dgearegmr(( \Delta t = 0.1 ))</td>
<td>0.19</td>
<td>0.5</td>
<td>0.89</td>
<td>1.29</td>
<td>1.67</td>
<td>2.49</td>
</tr>
<tr>
<td>dgearefpo(( \Delta t = 0.01 ))</td>
<td>0.03</td>
<td>0.12</td>
<td>0.22</td>
<td>0.32</td>
<td>0.43</td>
<td>0.64</td>
</tr>
<tr>
<td>dgearesfp(( \Delta t = 0.1 ))</td>
<td>1.81</td>
<td>5.12</td>
<td>9.25</td>
<td>13.40</td>
<td>17.4</td>
<td>25.74</td>
</tr>
<tr>
<td>dgearegfp(( \Delta t = 0.1 ))</td>
<td>0.18</td>
<td>0.5</td>
<td>0.9</td>
<td>1.3</td>
<td>1.7</td>
<td>2.51</td>
</tr>
</tbody>
</table>

Table 8: Case Study 2: Relative error when \( n = m = 20 \) and \( t_f \) variable 
\( (r=1, tol=1e-5, maxiter=100) \)

<table>
<thead>
<tr>
<th>( Er )</th>
<th>( t_f = 1 )</th>
<th>( t_f = 5 )</th>
<th>( t_f = 10 )</th>
<th>( t_f = 15 )</th>
<th>( t_f = 20 )</th>
<th>( t_f = 30 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl(( \Delta t = 0.1 ))</td>
<td>2.2e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dgearegmr(( \Delta t = 0.1 ))</td>
<td>1.1e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dgearefpo(( \Delta t = 0.01 ))</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dgearesfp(( \Delta t = 0.1 ))</td>
<td>1.1e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dgearegfp(( \Delta t = 0.1 ))</td>
<td>2.2e-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- There is convergence for all the algorithms choosing a good \( \Delta t \), but fastest is \( \text{dgearefpo} \). See Table 5.
- Relative error is similar for all algorithms (a bit worse for Fixed-Point). See Table 6.

Case study 3. This case study [11] consists of the following time-invariant DMRE

\[
\dot{X}(t) = X(t)T_{2k} + T_{2k}X(t) - X(t)T_{2k}X(t) + \alpha^2T_{2k}, \quad t \geq 0,
\]

\[
X(0) = I_{2k},
\]
Table 9: Case Study 3: Execution time when $t_f=1$, $n=m=16$ and $\Delta t$ variable ($r=1$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.93</td>
<td>2.36</td>
<td>6.49</td>
<td>11.81</td>
<td>51.59</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.29</td>
<td>0.39</td>
<td>0.97</td>
<td>1.74</td>
<td>7.99</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>1.83</td>
<td>2.31</td>
<td>6.28</td>
<td>11.55</td>
<td>50.47</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.29</td>
<td>0.36</td>
<td>0.99</td>
<td>1.78</td>
<td>8.16</td>
</tr>
</tbody>
</table>

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

$$T_2 = \begin{bmatrix} -1 & 1 \\ \alpha^2 & 1 \end{bmatrix},$$

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

where $\alpha$ controls the stiffness of the problem. The solution of this DMRE is given by

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

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

The following parameters were used for this problem:

- $r = 1$, $tol = 1e-5$.
- Size of problem $n = m = 16$ ($k = 4$).
- $\alpha = 100$ (stiff problem).

We carried out two kinds of test:

1. Fixing the final time ($t_f$) = 1 second and varying the step size ($\Delta t$) = 0.1, 0.05, 0.01, 0.005 and 0.001 seconds. Tables 9 and 10 show these tests. The Fixed-Point algorithms have not converged.

2. Fixing the step size ($\Delta t$) = 0.1, varying the final time ($t_f$) = 1, 5, 10, 15, 20, and 30 seconds. Figure 3 and Table 11 show these tests.

Conclusions for case study 3:

- There is convergence for four algorithms but, fastest are GMRES ($dgearegmr$) and GMRES with a first iteration of Fixed-Point ($dgearegfp$) algorithms. Fixed-Points algorithms have a bad behaviour.

- Relative error is similar for all algorithms that reach convergence. Anyway $\Delta t$ must be carefully chosen for each algorithm to reach the best convergence.
Table 10: Case Study 3: Relative error when $t_f=1$, $n=m=16$ and $\Delta t$ variable
($r=1$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
<td>2.48e-20</td>
<td>3.72e-20</td>
<td>1.90e-16</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>5.75e-20</td>
<td>3.72e-20</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>2.48e-20</td>
<td>2.68e-20</td>
<td>2.64e-20</td>
<td>2.48e-20</td>
<td>1.64e-20</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>4.42e-20</td>
<td>1.90e-16</td>
<td>2.73e-20</td>
<td>3.08e-20</td>
<td>1.90e-16</td>
</tr>
</tbody>
</table>

Figure 3: Case Study 3. Execution time when $\Delta t=0.1$, $n=m=16$ and $t_f$ variable
($r=1$, $tol=1e-5$, $maxiter=100$)

Table 11: Case Study 3: Relative error when $\Delta t=0.1$, $n=m=16$ and $t_f$ variable
($r=1$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$t_f=1$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
<td>1.90e-16</td>
<td>2.68e-20</td>
<td>2.68e-20</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>5.75e-20</td>
<td>5.75e-20</td>
<td>5.53e-20</td>
<td>5.75e-20</td>
<td>5.58e-20</td>
<td>5.58e-20</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>2.48e-20</td>
<td>2.48e-20</td>
<td>2.48e-20</td>
<td>2.48e-20</td>
<td>2.33e-20</td>
<td>2.32e-20</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>4.42e-20</td>
<td>4.42e-20</td>
<td>5.1e-20</td>
<td>4.42e-20</td>
<td>6.04e-20</td>
<td>6.04e-20</td>
</tr>
</tbody>
</table>

Case study 4. This stiff time-varying DMRE is a widely used test problem, known as the ”knee problem” [13, 15] defined as

$$\varepsilon \dot{x}(t) = \varepsilon - tx(t) + x^2(t), t \geq -1, x(-1) = -1, 0 < \varepsilon << 1,$$
Table 12: Case Study 4: Execution time when $t_f=1$, $n=m=2$ and $\Delta t$ variable ($r=2$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>0.02</td>
<td>0.04</td>
<td>0.14</td>
<td>0.28</td>
<td>1.12</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.02</td>
<td>0.05</td>
<td>0.19</td>
<td>0.36</td>
<td>1.41</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>0.01</td>
<td>0.02</td>
<td>0.08</td>
<td>0.21</td>
<td>0.80</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.02</td>
<td>0.03</td>
<td>0.10</td>
<td>0.27</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 13: Case Study 4: Relative error when $t_f=1$, $n=m=2$ and $\Delta t$ variable ($r=2$, $tol=1e-05$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>4.16e-06</td>
<td>4.76e-06</td>
<td>4.95e-06</td>
<td>4.97e-06</td>
<td>4.95e-06</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>4.16e-06</td>
<td>4.76e-06</td>
<td>4.95e-06</td>
<td>4.97e-06</td>
<td>4.95e-06</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>1.20</td>
<td>1.05</td>
<td>1.01</td>
<td>4.97e-06</td>
<td>4.99e-06</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>1.20</td>
<td>1.05</td>
<td>1.01</td>
<td>4.97e-06</td>
<td>4.99e-06</td>
</tr>
</tbody>
</table>

associated to coefficient matrix

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix} = \begin{bmatrix} t/\epsilon & -1/\epsilon \\ 1/2 & 0 \end{bmatrix}, n = m = 1.$$

The reduced solution $x = t$ is stable before 0 and $x \cong 0$ is stable past it. Parameter $\epsilon$ controls the stiffness of the problem.

The following parameters were used for this problem:

- $r = 2$, $tol = 1e-5$.
- Size of problem $n = 2$.
- $\epsilon = 1e-5$ (stiff problem).

We carried out two kinds of tests:

1. Fixing the final time ($t_f$)= 1 second and varying the step size ($\Delta t$)= 0.1, 0.05, 0.01, 0.005 and 0.001 seconds. Tables 12 and 13 show these tests.
2. Fixing the step size ($\Delta t$)= 0.1, varying the final time ($t_f$)= 1, 5, 10, 15, 20, and 30 seconds. Figure 4 and Table 14 show these tests.

Conclusions for Case Study 4:

- There is convergence for Sylvester and GMRES algorithms, but the fastest is the Sylvester algorithm (dgearesyl). Some algorithms converge to an incorrect solution and must be not considered. Only Sylvester and GMRES algorithms converge always to the correct solution.
Figure 4: Case Study 4: Execution time when $\Delta t=0.01$, $n=m=2$ and $t_f$ variable

\[(r=2, \text{tol}=1e-5, \text{maxiter}=100)\]

Table 14: Case Study 4: Relative error when $\Delta t=0.01$, $n=m=2$ and $t_f$ variable

\[(r=2, \text{tol}=1e-5, \text{maxiter}=100)\]

<table>
<thead>
<tr>
<th>Er</th>
<th>$t_f=1$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>4.16e-06</td>
<td>9.60e-07</td>
<td>4.99e-07</td>
<td>3.2e-07</td>
<td>2.49e-07</td>
<td>1.66e-07</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>4.16e-06</td>
<td>9.60e-07</td>
<td>4.99e-07</td>
<td>3.32e-07</td>
<td>2.49e-07</td>
<td>1.66e-07</td>
</tr>
</tbody>
</table>

- Relative error is similar for all the algorithms that reach convergence to the correct solution.

**Case study 5.** This stiff time-varying DMRE ([7, 15]) comes from a stiff two-point boundary value problem. This DMRE is defined as

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

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

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

\[
X(t) = \begin{bmatrix} -\varepsilon/t & 2(\sqrt{\varepsilon} + t)/2(1 - t\sqrt{\varepsilon}) \\ 0 & \sqrt{\varepsilon} \end{bmatrix}.
\]
Table 15: Case Study 5: Execution time when $t_f=1$, $n=m=2$ and $\Delta t$ variable ($r=1$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th>Method</th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>0.43</td>
<td>0.11</td>
<td>0.41</td>
<td>0.95</td>
<td>3.84</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.17</td>
<td>0.10</td>
<td>0.39</td>
<td>0.88</td>
<td>3.51</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>0.07</td>
<td>0.12</td>
<td>0.47</td>
<td>0.93</td>
<td>3.16</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.07</td>
<td>0.11</td>
<td>0.45</td>
<td>0.86</td>
<td>2.92</td>
</tr>
</tbody>
</table>

Table 16: Case Study 5: Relative error when $t_f=1$, $n=m=2$ and $\Delta t$ variable ($r=1$, $tol=1e-5$, $maxiter=100$)

<table>
<thead>
<tr>
<th>Method</th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>1.41</td>
<td>1.41</td>
<td>1.41</td>
<td>0.00</td>
<td>1.22e-18</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>1.41</td>
<td>1.41</td>
<td>1.41</td>
<td>2.20e-16</td>
<td>2.23e-16</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>7.22e-19</td>
<td>1.65e-18</td>
<td>0.00</td>
<td>0.00</td>
<td>1.22e-18</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>2.98e-18</td>
<td>2.33e-18</td>
<td>1.92e-18</td>
<td>1.22e-18</td>
<td>4.48e-16</td>
</tr>
</tbody>
</table>

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

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

The following parameters were used for this problem:

- $r = 1$, $tol = 1e-5$.
- Size of problem $n = 2$.
- $\epsilon = 1e-5$ (stiff problem), controls the stiffness of the problem.

We carried out two kinds of tests:

1. Fixing the final time ($t_f$)= 1 second and varying the step size ($\Delta t$)= 0.1, 0.05, 0.01, 0.005 and 0.001 seconds. Tables 15 and 16 show these tests.
2. Fixing the step size ($\Delta t$)= 0.1, varying the final time ($t_f$)= 1, 5, 10, 15, 20, and 30 seconds. Tables 17 and 18 show these tests. Only dgearesyl and dgearegmr have been considered.

Conclusions for Case Study 5:

- There is convergence for several algorithms but, fastest are the combined dgearesfp and dgearegfp algorithms. Some algorithms (dgearesyl and dgearegmr) converge sometimes to an incorrect solution.
- Relative error is best for the "combined" methods.
Table 17: Case Study 5: Execution time when $\Delta t=0.1$, $n=m=2$ and $t_f$ variable

<table>
<thead>
<tr>
<th></th>
<th>$t_f=1$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesfp</td>
<td>0.14</td>
<td>0.14</td>
<td>0.24</td>
<td>0.34</td>
<td>0.44</td>
<td>0.62</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.13</td>
<td>0.13</td>
<td>0.23</td>
<td>0.33</td>
<td>0.43</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Table 18: Case Study 5: Relative error when $\Delta t=0.1$, $n=m=2$ and $t_f$ variable

<table>
<thead>
<tr>
<th></th>
<th>$t_f=1$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesfp</td>
<td>1.85e-16</td>
<td>1.70e-16</td>
<td>1.74e-16</td>
<td>1.16e-16</td>
<td>1.76e-16</td>
<td>2.88e-20</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>3.68e-18</td>
<td>1.84e-18</td>
<td>1.74e-16</td>
<td>2.45e-15</td>
<td>9.64e-20</td>
<td>1.18e-16</td>
</tr>
</tbody>
</table>

Case study 6. This time-varying DMRE [10] is defined as

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

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

$$T_2 = \begin{bmatrix} a(t) & b(t) \\ -b(t) & a(t) \end{bmatrix},$$

$$T_{2^k} = T_2 \otimes I_{2^{k-1}} + I_2 \otimes T_{2^{k-1}}, k \geq 2,$$

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

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

The following parameters were used for this problem:

- $r = 3$, $tol = 1e-6$.
- Size of problem $n = m = 16$ ($k = 4$). We chose this size because it is quite representative.

We carried out two kinds of tests:

1. Fixing the final time ($t_f)= 1$ second and varying the step size ($\Delta t)= 0.1$, 0.05, 0.01, 0.005 and 0.001 seconds. Tables 19 and 20 show these tests.
2. Fixing the step size ($\Delta t)= 0.001$, varying the final time ($t_f)= 1$, 5, 10, 15, 20, and 30 seconds. Figure 5 and Table 21 show these tests.

Conclusions for Case Study 6:

- When converging fastest are the Fixed-Point algorithms.
Table 19: Case Study 6: Execution time when $t_f=1$, $n=m=16$ and $\Delta t$ variable
($r=3$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>0.99</td>
<td>1.63</td>
<td>5.51</td>
<td>10.97</td>
<td>54.64</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>0.19</td>
<td>0.32</td>
<td>1.09</td>
<td>2.16</td>
<td>10.83</td>
</tr>
<tr>
<td>dgearefpo</td>
<td>No Conv.</td>
<td>0.03</td>
<td>0.09</td>
<td>0.18</td>
<td>0.74</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>0.98</td>
<td>1.42</td>
<td>5.52</td>
<td>10.99</td>
<td>50.30</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>0.82</td>
<td>0.33</td>
<td>1.83</td>
<td>3.15</td>
<td>16.15</td>
</tr>
</tbody>
</table>

Table 20: Case Study 6: Relative error when $t_f=1$, $n=m=16$ and $\Delta t$ variable
($r=3$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t=0.1$</th>
<th>$\Delta t=0.05$</th>
<th>$\Delta t=0.01$</th>
<th>$\Delta t=0.005$</th>
<th>$\Delta t=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>4.35e-02</td>
<td>5.95e-03</td>
<td>2.15e-04</td>
<td>5.33e-05</td>
<td>2.11e-06</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>4.35e-02</td>
<td>5.95e-03</td>
<td>2.15e-04</td>
<td>5.33e-05</td>
<td>2.11e-06</td>
</tr>
<tr>
<td>dgearefpo</td>
<td>No Conv.</td>
<td>5.95e-03</td>
<td>2.16e-04</td>
<td>5.44e-05</td>
<td>1.52e-06</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>4.35e-02</td>
<td>5.95e-03</td>
<td>2.15e-04</td>
<td>5.33e-05</td>
<td>2.11e-06</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>4.35e-02</td>
<td>5.95e-03</td>
<td>2.15e-04</td>
<td>5.33e-05</td>
<td>2.11e-06</td>
</tr>
</tbody>
</table>

Table 21: Case Study 6: Relative error when $\Delta t=0.001$, $n=m=16$ and $t_f$
variable
($r=3$, $tol=1e-6$, $maxiter=100$)

<table>
<thead>
<tr>
<th></th>
<th>$t_f=1$</th>
<th>$t_f=5$</th>
<th>$t_f=10$</th>
<th>$t_f=15$</th>
<th>$t_f=20$</th>
<th>$t_f=30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgearesyl</td>
<td>2.11e-06</td>
<td>9.14e-06</td>
<td>1.48e-06</td>
<td>1.37e-06</td>
<td>3.39e-16</td>
<td>1.07e-05</td>
</tr>
<tr>
<td>dgearegmr</td>
<td>2.11e-06</td>
<td>9.14e-06</td>
<td>1.48e-06</td>
<td>1.37e-06</td>
<td>3.39e-16</td>
<td>1.07e-05</td>
</tr>
<tr>
<td>dgearefpo</td>
<td>1.52e-06</td>
<td>6.41e-06</td>
<td>1.06e-06</td>
<td>1.03e-06</td>
<td>2.95e-16</td>
<td>8.36e-06</td>
</tr>
<tr>
<td>dgearesfp</td>
<td>2.11e-06</td>
<td>9.14e-06</td>
<td>1.48e-06</td>
<td>1.37e-06</td>
<td>3.39e-16</td>
<td>1.07e-05</td>
</tr>
<tr>
<td>dgearegfp</td>
<td>2.11e-06</td>
<td>9.14e-06</td>
<td>1.48e-06</td>
<td>1.37e-06</td>
<td>3.39e-16</td>
<td>1.07e-05</td>
</tr>
</tbody>
</table>

- Relative error is similar for all the algorithms, but a little better for the Fixed-Point algorithms.

From the above experimental results we can extract several conclusions about algorithms developed in Section 3:

- There is no the best algorithm for solving DMREs. It depends on the problem to be solved. We have determined the best algorithm for each case study by looking for the fastest execution time ($Te$) and the least relative error ($Er$) for a given $\Delta t$. The conclusions on execution time are summarized in Table 22. All algorithms have a similar relative error except
Figure 5: Case Study 6: Execution time when $\Delta t=0.001$, $n=m=16$ and $t_f$ variable

\[(r=3, tol=1e-6, maxiter=100)\]

Table 22: Summary of the best algorithms in terms of execution time ($T_e$).

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Algorithm for Exec. Time</td>
<td>Fixed-Point</td>
<td>Fixed-Point</td>
<td>GMRES</td>
<td>Sylvester</td>
<td>Combined: \texttt{dgearesfp} \texttt{dgearegfp}</td>
</tr>
</tbody>
</table>

for case study 5, where the best are the combined algorithms (\texttt{dgearesfp} and \texttt{dgearegfp}).

- Combined algorithms are a good choosing for some problems.

- Another important problem the stiffness. Fixed-Point algorithms are not good when the problem is very stiff. In general GMRES and Sylvester algorithms (in this order) work better than Fixed-Point algorithms for very stiff problems.

- With the algorithm scheme presented in this paper, that is, solving a DMRE using one (or several) AMREs, other algorithms for solving AMREs can be presented, and added to our scheme.

4.2 A general adaptive Algorithm

It is possible to design an efficient and adaptive algorithm that solves DMREs depending on the stiffness of problem. This algorithm is based on variable-
coefficient strategies [2] by using an interpolating polynomial $P_k$ which interpolates the points $(t_k, X_{k-r}), (t_{k+1}, X_{k-r+1}), \ldots, (t_k, X_k)$ and then the implicit equation

$$\dot{P}_k(t_k) = A_{21}(t) + A_{22}(t_k)X_k - X_kA_{11}(t_k) - X_kA_{12}(t)X_k$$

is solved for $X_k$ by using one of the AMREs algorithms developed in Section 3.

For selecting the step size in the $k$ step we consider the parameters $\varepsilon_1$, $\varepsilon_2$ and $s$ such as $0 < \varepsilon_1 \lesssim E_k \lesssim \varepsilon_2$, where

$$E_k = \frac{\|X_k - X_{k-1}\|_\infty}{\|X_k\|_\infty}.$$

The step size is selected according to the following scheme:

1: if $\|X_k - X_{k-1}\|_\infty < \varepsilon_1 \|X_k\|_\infty$ then
2: $\Delta t_k = \delta \Delta t_{k-1}$
3: else if $\|X_k - X_{k-1}\|_\infty > \varepsilon_2 \|X_k\|_\infty$ then
4: $\Delta t_k = \max(\Delta t_{k-1}/\delta, \varepsilon_3)$
5: end if

where $\varepsilon_3$ ($0 < \varepsilon_3 < \varepsilon_1$) is a necessary parameter to avoid that the step size becomes very small, and $\delta > 1$.

We used the following values for our tests:

- $\varepsilon_1 = \Delta t_0 \cdot 10^{-2}$,
- $\varepsilon_2 = \Delta t_0 \cdot 10^{-1}$,
- $\varepsilon_3 = \Delta t_0 \cdot 10^{-3}$,
- $\delta = 1 + \varepsilon_3$,

where $\Delta t_0$ is a initial step size provided by the user.

The complete adaptive algorithm has two versions: dgevdragbd for time-varying DMREs and dgeidragbd for time-invariant DMREs. The input parameters used are similar to Algorithm 1, but now the user does not tell the algorithm to solve the AMRE. The user indicates the stiffness by a parameter $s$ that indicates if the problem is non stiff or low stiff ($s = 0$) or stiff ($s \neq 0$):

- $s=0$ (non stiff or low stiff): The DMRE is solved by the Fixed-Point algorithm.
- $s=1$ (stiff): The DMRE is solved by the combined BDF-Sylvester/Fixed-Point algorithm.
- $s=2$ (stiff): The DMRE is solved by the combined BDF-GMRES/Fixed-Point algorithm.
- $s=3$ (stiff): The DMRE is solved by the Sylvester algorithm.
### Table 23: Relative errors for adaptive algorithm \( r=2, \; tol=1e-5, \; maxiter=100 \)

<table>
<thead>
<tr>
<th>Er</th>
<th>Case Study 2</th>
<th>Case Study 3</th>
<th>Case Study 4</th>
<th>Case Study 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s = 1 )</td>
<td>0</td>
<td>3.29e-20</td>
<td>4.999e-7</td>
<td>5.92e-15</td>
</tr>
<tr>
<td>( s = 2 )</td>
<td>0</td>
<td>3.81e-16</td>
<td>4.999e-7</td>
<td>6.00e-15</td>
</tr>
<tr>
<td>( s = 3 )</td>
<td>0</td>
<td>1.64e-20</td>
<td>4.999e-7</td>
<td>1.56e-15</td>
</tr>
<tr>
<td>( s = 4 )</td>
<td>0</td>
<td>3.81e-16</td>
<td>4.999e-7</td>
<td>1.56e-15</td>
</tr>
</tbody>
</table>

### Table 24: Execution times for adaptive algorithm \( r=2, \; tol = 10^{-5}, \; maxiter=100 \)

<table>
<thead>
<tr>
<th>Te</th>
<th>Case Study 2</th>
<th>Case Study 3</th>
<th>Case Study 4</th>
<th>Case Study 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s = 1 )</td>
<td>1.30</td>
<td>1.06</td>
<td>2.76</td>
<td>4.71</td>
</tr>
<tr>
<td>( s = 2 )</td>
<td>0.11</td>
<td>0.13</td>
<td>2.67</td>
<td>3.41</td>
</tr>
<tr>
<td>( s = 3 )</td>
<td>1.33</td>
<td>1.11</td>
<td>2.61</td>
<td>6.4</td>
</tr>
<tr>
<td>( s = 4 )</td>
<td>0.11</td>
<td>0.13</td>
<td>2.46</td>
<td>5.0</td>
</tr>
</tbody>
</table>

- \( s=4 \) (stiff): The DMRE is solved by the GMRES algorithm.

Tables 23 and 24 contain relative errors and execution times of Case studies 2, 3, 4 and 5 (stiff problems) when \( s \in \{1,2,3,4\} \). The parameters of problems and algorithms were:

- Case study 2: \( n = 16, \; \alpha = 1000, \; t_f = 3, \; r = 2, \; tol = 10^{-5}, \; \Delta t_0 = 0.1, \; maxiter = 100 \).
- Case study 3: \( n = 16, \; \alpha = 100, \; t_f = 1, \; r = 2, \; tol = 10^{-5}, \; \Delta t_0 = 0.1, \; maxiter = 100 \).
- Case study 4: \( n = 1, \; \epsilon = 10^{-4}, \; t_f = 100, \; r = 2, \; tol = 10^{-5}, \; \Delta t_0 = 0.01, \; maxiter = 100 \).
- Case study 5: \( n = 2, \; \epsilon = 10^{-4}, \; t_f = 50, \; r = 2, \; tol = 10^{-5}, \; \Delta t_0 = 0.01, \; maxiter = 100 \).

Table 23 shows that the combined algorithms (\( s = 1,2 \)) have the same relative errors as the single algorithms (\( s = 3,4 \)) in two of the four case studies and lower relative errors than the single algorithms in the other case studies. Table 24 shows that single algorithms have lower or equal execution times than combined algorithms in three of the four case studies.

### 5 Conclusions and Future Work

We developed a family of algorithms for solving DMREs using adaptive techniques. Because there is no best algorithm to solve all problems, a general
purpose adaptive implementation which selects the most adequate adaptive algorithm depending on the DMRE has been designed. The algorithms can be chosen by using a parameter that indicates the stiffness of the problem. MATLAB implementations of these algorithms have been also done. The source codes can be downloaded from the following http address:

http://www.grycap.upv.es/dmretoolbox/DMRE_BDF.htm

There are three items for future work:

- Use other BDF algorithms for solving AMREs as Adams-Moulton algorithm [3, pp. 127].
- Parallel implementation of the algorithms presented in this work will be carried out in a distributed memory platform, using the message passing paradigm, MPI [18] and BLACS [16] for communications, and PBLAS [12] and ScalAPACK [6] for computations.

References


