Ibáñez González, JJ.; Hernández García, V. (2011). Solving differential matrix Riccati equations by a piecewise-linearized method based on diagonal Padé approximants. Computer Physics Communications. 182(3):669-678. doi:10.1016/j.cpc.2010.11.024.

Solving Differential Matrix Riccati Equations by a Piecewise-linearized Method Based on Diagonal Padé Approximants

Javier Ibáñez [a,*] Vicente Hernández [a]

[a] *Instituto de Instrumentación para Imagen Molecular (I3M), Camino de Vera s/n, 46022-Valencia (Spain)*

Abstract

Differential Matrix Riccati Equations (DMREs) appear in several branches of science such as applied Physics and Engineering. For example, these equations play a fundamental role in Control Theory, optimal control, filtering and estimation, decoupling and order reduction, etc. In this paper a new method based on a theorem proved in this paper is described for solving DMREs by a piecewise-linearized approach. This method is applied for developing two block-oriented algorithms based on Diagonal Padé Approximants. MATLAB versions of the above algorithms are developed, comparing, under equal conditions, accuracy and computational costs with other piecewise-linearized algorithms implemented by the authors. Experimental results show the advantages of solving stiff or non-stiff DMREs by the implemented algorithms.

Key words: Differential Matrix Riccati Equation (DMRE), Piecewise-linearized Method, Ordinary Differential Equation (ODE), Initial Value Problem (IVP), Linear Differential Equation (LDE), Conmutant Equation, Algebraic Matrix Sylvester Equation (AMSE), Padé Approximants.
PACS: 87.64.Aa

1 Introduction

This paper presents a methodology for solving Differential Matrix Riccati Equations (DMREs) based on a piecewise-linearized method which uses Padé

* Corresponding author Javier Ibáñez. Telf: +34-96-3877356. Fax: +34-96-3877359
Email address: jjibanez@dsic.upv.es (Javier Ibáñez).

approximants to compute the exponentials of two block-defined matrices. These equations have the form

$$\dot{X} = A_{21}(t) + A_{22}(t)X - XA_{11}(t) - XA_{12}(t)X, \quad t_0 \leq t \leq t_f, \qquad (1)$$
$$X = X(t) \in \mathbb{R}^{m\times n}, \quad X(t_0) = X_0 \in \mathbb{R}^{m\times n},$$

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 play an important role in the electrodynamic theory of stratified media, in the theory of multimode electric power lines, in the hydraulics of pipe lines, etc. They also appear in Control Theory, for example in the time-invariant linear quadratic optimal control problem, in the estimation of the system parameters and in the system state, etc.

Since the mid seventies, many different methods have been proposed: linearization methods [1–3], Chandrasekhar method [4], superposition methods [5,6], BDF methods [7–11], Hamiltonian methods [12], unconventional reflexive numerical methods [13], Piecewise-linearized methods [14], etc.

In [14] we developed a piecewise-linearized method based on the Conmutant and we implemented efficient block-oriented algorithms for solving DMREs. In this paper a new piecewise-linearized method for solving DMREs is presented based on Theorem 3 in Section 4, and two block-oriented algorithms based on this method have been developed.

This paper is structured as follows. Section 2 describes a numerical integration of ODEs based on a piecewise-linearized method [15], which has served as the basis for the methods that are described in the following sections. Section 3 describes a piecewise-linearized method developed by the authors [14] which solves DMREs by the Con mutant Equation. Section 4 presents another piecewise-linearized method based on a theorem proved in this paper (Theorem 3) and two block-oriented algorithms. A theoretical study in terms of flops requirements is included. The experimental results of the MATLAB implementations are shown in Section 5. Finally, some conclusions and future work are outlined in Section 6.

2 Solving ODEs by a Piecewise-linearized Method

In this section we show a piecewise-linearized method to solve ODEs [15] which is used in Sections 3 and 4. A family of one step methods for solving ODEs are the piecewise-linearized methods [16,17,15]. These methods solve an IVP by approximating the right hand-side of the ODE by a degree one Taylor polynomial. The resulting approximation can be integrated analytically to

obtain the solution in each subinterval and yields the exact solution for linear problems.

Let

$$\dot{x}(t) = f(t, x(t)), \quad t \in [t_0, t_f], \tag{2}$$

be an ODE with initial value

$$x(t_0) = x_0 \in \mathbb{R}^n,$$

so that the first order partial derivatives of $f(t, x)$ are continuous on $[t_0, t_f] \times \mathbb{R}^n$. Given a mesh $t_0 < t_1 < \cdots < t_{l-1} < t_l = t_f$, the ODE (2) can be approximated by a set of Linear Differential Equations (LDEs) obtained as a result of a linear approximation of $f(t, x(t))$ at each subinterval,

$$\begin{gathered} \dot{y}(t) = f_i + J_i(y(t) - y_i) + g_i(t - t_i), \quad t \in [t_i, t_{i+1}], \\ y(t_i) = y_i, \quad i = 0, 1, \ldots, l-1, \end{gathered} \tag{3}$$

where

$$\begin{gathered} f_i = f(t_i, y_i) \in \mathbb{R}^n, \\ J_i = \frac{\partial f}{\partial x}(t_i, y_i) \in \mathbb{R}^{n \times n} \text{ (Jacobian matrix)}, \\ g_i = \frac{\partial f}{\partial t}(t_i, y_i) \in \mathbb{R}^n \text{ (gradient vector)}. \end{gathered}$$

The LDE associated to the first subinterval,

$$\dot{y}(t) = f_0 + J_0(y(t) - y_0) + g_0(t - t_0), \quad t \in [t_0, t_1],$$

is solved considering the initial value $y(t_0) = y_0 = x_0$. Its solution is given by

$$y(t) = y_0 + \int_{t_0}^{t} e^{J_0(t-\tau)}[f_0 + g_0(\tau - t_0)]d\tau, \quad t \in [t_0, t_1].$$

Thus, it is possible to compute $y_1 = y(t_1)$.

By proceeding in the same way, the solution of the LDE associated to the subinterval i, $i = 1, \ldots, l-1$, is

$$y(t) = y_i + \int_{t_i}^{t} e^{J_i(t-\tau)}[f_i + g_i(\tau - t_i)]d\tau, \quad t \in [t_i, t_{i+1}].$$

If $f(t, x)$ is a Lipschitz function on $[t_0, t_f] \times \mathbb{R}^n$ and its second order partial derivatives are bounded on that region, the above piecewise-linearized method converges [16, pp. 281]. If the (1,1) Padé approximation is used to compute $e^{J_i(t-t_i)}$, the above method is consistent of order 2 for autonomous ODEs and 1 for non autonomous ODEs, and linearly stable [18, pp. 26].

Theorem 1 ([15]) *The solution of the LDE*

$$\begin{gathered}\dot{y}(t) = f_i + J_i(y(t) - y_i) + g_i(t - t_i), \quad t \in [t_i, t_{i+1}], \\ y(t_i) = y_i, \\ f_i \in \mathbb{R}^n, \quad J_i \in \mathbb{R}^{n\times n}, \quad g_i \in \mathbb{R}^n,\end{gathered}$$

is

$$y(t) = y_i + E_{12}^{(i)}(t - t_i)f_i + E_{13}^{(i)}(t - t_i)g_i, \tag{4}$$

where $E_{12}^{(i)}(t - t_i)$ *and* $E_{13}^{(i)}(t - t_i)$ *are blocks* $(1, 2)$ *and* $(1, 3)$ *of* $E = e^{C_i(t-t_i)}$, *where*

$$C_i = \begin{bmatrix} J_i & I_n & 0_n \\ 0_n & 0_n & I_n \\ 0_n & 0_n & 0_n \end{bmatrix}. \qquad \square$$

If t is replaced by t_{i+1} in (4), the approximate solution of the ODE (2) at t_{i+1} is given by

$$y_{i+1} = y_i + E_{12}^{(i)}(\Delta t_i)f_i + E_{13}^{(i)}(\Delta t_i)g_i.$$

Therefore, the solutions of the ODE (2) at $t_1, t_2, \ldots, t_l = t_f$ can be computed by using the above expression.

3 Solving DMREs by a Piecewise-linearized Method based on the Conmutant Equation

This section describes a piecewise-linearized method developed by the authors [14], which is compared with the piecewise-linearized method developed in this paper. Let us suppose that the right-hand side of (1) ,

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

is a Lipschitz function on $[t_0, t_f]\times\mathbb{R}^{m\times n}$ and its second order partial derivatives are bounded on that region. If vec operator [19, p. 244] is applied to DMRE (1), then

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = \mathrm{vec}(X_0),$$

where

$$x(t) = \mathrm{vec}(X(t)) \in \mathbb{R}^{mn},$$

and

$$f(t, x(t)) = \mathrm{vec}(A_{21}(t) + A_{22}(t)X(t) - X(t)A_{11}(t) - X(t)A_{12}(t)X(t)),$$

which can be expressed as

$$f(t,x(t)) = \text{vec}(A_{21}(t)) + [I_n \otimes A_{22}(t) - A_{11}^T(t) \otimes I_m]x(t) - [I_n \otimes (X(t)A_{12}(t))]x(t)$$

or

$$f(t,x(t)) = \text{vec}(A_{21}(t)) + [I_n \otimes A_{22}(t) - A_{11}^T(t) \otimes I_m]x(t) - [(A_{12}(t)X(t))^T \otimes I_m]x(t).$$

If we consider the mesh $t_0 < t_1 < \cdots < t_{l-1} < t_l = t_f$, and we apply the piecewise-linearized process explained in Section 2, the following LDEs are obtained

$$\begin{gathered} \dot{y}(t) = \text{vec}(F_i) + J_i(y(t) - y_i) + \text{vec}(G_i)(t - t_i), \quad t \in [t_i, t_{i+1}], \\ y(t_i) = y_i, \quad i = 0, 1, \ldots, l-1, \end{gathered} \tag{5}$$

where

$$\begin{aligned} F_i &= A_{21}(t_i) + A_{22}(t_i)Y_i - Y_iA_{11}(t_i) - Y_iA_{12}(t_i)Y_i, \\ G_i &= \dot{A}_{21}(t_i) + \dot{A}_{22}(t_i)Y_i - Y_i\dot{A}_{11}(t_i) - Y_i\dot{A}_{12}(t_i)Y_i. \end{aligned}$$

Since $f(t,x)$ is a Lipschitz function on $[t_0, t_f] \times \mathbb{R}^{mn}$ and its second order partial derivatives are bounded on that region, the above piecewise-linearized process converges [16]. If we apply Theorem 1, the solution of (5) at t_{i+1} is

$$y_{i+1} = y_i + E_{12}^{(i)}(\Delta t_i)f_i + E_{13}^{(i)}(\Delta t_i)g_i, \tag{6}$$

where $E_{12}^{(i)}(\Delta t_i)$ and $E_{13}^{(i)}(\Delta t_i)$ are the (1,2) and (1,3) blocks of matrix $E_i = e^{C_i \Delta t_i}$, where

$$C_i = \begin{bmatrix} J_i & I_{mn} & 0_{mn} \\ 0_{mn} & 0_{mn} & I_{mn} \\ 0_{mn} & 0_{mn} & 0_{mn} \end{bmatrix}, \tag{7}$$

and

$$\begin{aligned} J_i = \frac{\partial f}{\partial x}(t_i, y_i) &= I_n \otimes A_i - B_i^T \otimes I_m, \\ A_i &= A_{22}(t_i) - Y_iA_{12}(t_i), \\ B_i &= A_{11}(t_i) + A_{12}(t_i)Y_i. \end{aligned} \tag{8}$$

If $\text{mat}_{m\times n}$ operator [20, p. 2104] is applied to (6), the approximate solution of (1) at t_{i+1} can be obtained from expression

$$Y_{i+1} = Y_i + \text{mat}_{m\times n}(E_{12}^{(i)}(\Delta t_i)\text{vec}(F_i) + E_{13}^{(i)}(\Delta t_i)\text{vec}(G_i)). \tag{9}$$

Based on the Conmutant Equation, the authors proved in [14] the following theorem and corollary which allow to compute matrix Y_{i+1} in (9).

Theorem 2 *If matrices A_i and B_i of* (8) *do not have eigenvalues in common, then the matrix Y_{i+1} in* (9) *can be computed from*

$$Y_{i+1} = Y_i + W_f + W_g, \tag{10}$$

where W_f, W_g are the solutions of the Algebraic Matrix Sylvester Equations (AMSEs)

$$A_i W_f - W_f B_i = e^{A_i \Delta t_i} F_i e^{-B_i \Delta t_i} - F_i, \tag{11}$$

$$A_i W_g - W_g B_i = W_i - G_i \Delta t_i, \tag{12}$$

and W_i satisfies the AMSE

$$A_i W_i - W_i B_i = e^{A_i \Delta t_i} G_i e^{-B_i \Delta t_i} - G_i. \qquad \square \tag{13}$$

As (11), (12) and (13) have the same coefficient matrices A_i, B_i, the computational cost of computing Y_{i+1} can be reduced.

Corollary 1 *If the DMRE* (1) *is time-invariant and matrices A_i and B_i of* (8) *do not have eigenvalues in common, then matrix Y_{i+1} in* (9) *can be computed as*

$$Y_{i+1} = Y_i + W_f, \tag{14}$$

where W_f is the solution of the AMSE

$$A_i W_f - W_f B_i = e^{A_i \Delta t_i} F_i e^{-B_i \Delta t_i} - F_i. \qquad \square \tag{15}$$

4 A Piecewise-linearized Method for Solving DMREs based on Padé Approximants

As the solution of LDE (5) associated to subinterval $[t_i, t_{i+1}]$ is

$$y(t) = y_i + \int_{t_i}^{t} e^{J_i(t-\tau)}[\mathrm{vec}(F_i) + \mathrm{vec}(G_i)(\tau - t_i)]d\tau, \quad t \in [t_i, t_{i+1}], \tag{16}$$

then the approximate solution of DMRE (1) at t_{i+1} can be obtained as

$$Y_{i+1} = \mathrm{mat}_{m\times n}(y_{i+1}), \tag{17}$$

where

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} e^{J_i(t-\tau)}[\mathrm{vec}(F_i) + \mathrm{vec}(G_i)(\tau - t_i)]d\tau. \tag{18}$$

Theorem 3 *The matrix Y_{i+1}, which appears in* (17)*, can be computed as follows*

$$Y_{i+1} = Y_i + [F_{12}^{(i)}(\Delta t_i) + H_{13}^{(i)}(\Delta t_i)][F_{22}^{(i)}(\Delta t_i)]^{-1}, \tag{19}$$

where $Y_i = \mathrm{mat}_{m\times n}(y_i)$, $F_{12}^{(i)}(\Delta t_i)$ *and* $F_{22}^{(i)}(\Delta t_i)$ *are (1,2) and (2,2) blocks of matrix* $e^{C_i\Delta t_i}$,

$$C_i = \begin{bmatrix} A_i & F_i \\ 0_{n\times m} & B_i \end{bmatrix}, \tag{20}$$

and $H_{13}^{(i)}(\Delta t_i)$ *is the (1,3) block of matrix* $e^{D_i\Delta t_i}$,

$$D_i = \begin{bmatrix} A_i & G_i & 0_{m\times n} \\ 0_{n\times m} & B_i & I_n \\ 0_{n\times m} & 0_{n\times n} & B_i \end{bmatrix}, \tag{21}$$

$$\begin{gathered} F_i = A_{21}(t_i) + A_{22}(t_i)Y_i - Y_iA_{11}(t_i) - Y_iA_{12}(t_i)Y_i, \\ G_i = \dot{A}_{21}(t_i) + \dot{A}_{22}(t_i)Y_i - Y_i\dot{A}_{11}(t_i) - Y_i\dot{A}_{12}(t_i)Y_i, \\ A_i = A_{22}(t_i) - Y_iA_{12}(t_i), \\ B_i = A_{11}(t_i) + A_{12}(t_i)Y_i. \end{gathered}$$

Proof. If we define $s = \tau - t_i$ and $\theta = t - t_i$, vector $y(t)$ in (16) can be expressed as

$$\begin{gathered} y(t) = y_i + \int_0^\theta e^{J_i(\theta-s)}\mathrm{vec}(F_i)ds + \int_0^\theta e^{J_i(\theta-s)}g_i\mathrm{vec}(G_i)sds = \\ = y_i + \int_0^\theta e^{(I_n\otimes A_i - B_i^T\otimes I_m)(\theta-s)}\mathrm{vec}(F_i)ds + \int_0^\theta e^{(I_n\otimes A_i - B_i^T\otimes I_m)(\theta-s)}\mathrm{vec}(G_i)sds = \\ = y_i + \int_0^\theta (e^{-B_i^T(\theta-s)}\otimes e^{A_i(\theta-s)})\mathrm{vec}(F_i)ds + \int_0^\theta (e^{-B_i^T(\theta-s)}\otimes e^{A_i(\theta-s)})\mathrm{vec}(F_i)sds. \end{gathered}$$

If property (4) of Section 1 is applied, $Y(t) = \mathrm{mat}_{m\times n}(y(t))$ is obtained as

$$Y(t) = Y_i + \int_0^\theta e^{A_i(\theta-s)}F_ie^{-B_i(\theta-s)}ds + \int_0^\theta e^{A_i(\theta-s)}G_ie^{-B_i(\theta-s)}sds, \tag{22}$$

where

$$F_i = \mathrm{mat}_{m\times n}(f_i), \quad G_i = \mathrm{mat}_{m\times n}(g_i).$$

In order to compute the first integral in (22), let us consider

$$C_i = \begin{bmatrix} A_i & F_i \\ 0_{n\times m} & B_i \end{bmatrix},$$

and the exponential of matrix $C_i\theta$,

$$e^{C_i\theta} = \begin{bmatrix} F_{11}^{(i)}(\theta) & F_{12}^{(i)}(\theta) \\ 0_{n\times m} & F_{22}^{(i)}(\theta) \end{bmatrix}.$$

Since

$$\frac{de^{C_i\theta}}{d\theta} = C_i e^{C_i\theta},$$

then

$$\begin{bmatrix} \frac{dF_{11}^{(i)}(\theta)}{d\theta} & \frac{dF_{12}^{(i)}(\theta)}{d\theta} \\ 0_{n\times m} & \frac{dF_{22}^{(i)}(\theta)}{d\theta} \end{bmatrix} = \begin{bmatrix} A_i & F_i \\ 0_{n\times m} & B_i \end{bmatrix} \begin{bmatrix} F_{11}^{(i)}(\theta) & F_{12}^{(i)}(\theta) \\ 0_{n\times m} & F_{22}^{(i)}(\theta) \end{bmatrix}$$
$$= \begin{bmatrix} A_i F_{11}^{(i)}(\theta) & A_i F_{12}^{(i)}(\theta) + F_i F_{22}^{(i)}(\theta) \\ 0_{n\times m} & B_i F_{22}^{(i)}(\theta) \end{bmatrix}.$$

Equating blocks (1,1), (1,2) and (2,2) of both members of the previous equation and considering that

$$e^{C_i\theta}\Big|_{\theta=0} = I_{m+n},$$

the following LDEs are obtained

$$\frac{dF_{11}^{(i)}(\theta)}{d\theta} = A_i F_{11}^{(i)}(\theta), \quad F_{11}^{(i)}(0) = I_m, \tag{23}$$

$$\frac{dF_{22}^{(i)}(\theta)}{d\theta} = B_i F_{22}^{(i)}(\theta), \quad F_{22}^{(i)}(0) = I_n, \tag{24}$$

$$\frac{dF_{12}^{(i)}(\theta)}{d\theta} = A_i F_{12}^{(i)}(\theta) + F_i F_{22}^{(i)}(\theta), \quad F_{12}^{(i)}(0) = 0_{m\times n}. \tag{25}$$

Solving (23) and (24), then

$$F_{11}^{(i)}(\theta) = e^{A_i\theta},$$
$$F_{22}^{(i)}(\theta) = e^{B_i\theta}.$$

If we replace $F_{22}^{(i)}(\theta) = e^{B_i\theta}$ in (25), the following LDE is obtained

$$\frac{dF_{12}^{(i)}(\theta)}{d\theta} = A_i F_{12}^{(i)}(\theta) + F_i e^{B_i\theta}, \quad F_{12}^{(i)}(0) = 0_{m\times n},$$

therefore

$$F_{12}^{(i)}(\theta) = \int_0^\theta e^{A_i(\theta-s)} F_i e^{B_i s} ds.$$

In order to compute the second integral that appears in expression (22), let

$$D_i = \begin{bmatrix} A_i & G_i & 0_{m\times n} \\ 0_{n\times m} & B_i & I_n \\ 0_{n\times m} & 0_{n\times n} & B_i \end{bmatrix},$$

and exponential of matrix $D_i\theta$,

$$e^{D_i\theta} = \begin{bmatrix} H_{11}^{(i)}(\theta) & H_{12}^{(i)}(\theta) & H_{13}^{(i)}(\theta) \\ 0_{n\times m} & H_{22}^{(i)}(\theta) & H_{23}^{(i)}(\theta) \\ 0_{n\times m} & 0_{n\times n} & H_{33}^{(i)}(\theta) \end{bmatrix}. \tag{26}$$

Since

$$\frac{de^{D_i\theta}}{d\theta} = D_i e^{D_i\theta},$$

then

$$\begin{bmatrix} \frac{dH_{11}^{(i)}(\theta)}{d\theta} & \frac{dH_{12}^{(i)}(\theta)}{d\theta} & \frac{dH_{13}^{(i)}(\theta)}{d\theta} \\ 0_{n\times m} & \frac{dH_{22}^{(i)}(\theta)}{d\theta} & \frac{dH_{23}^{(i)}(\theta)}{d\theta} \\ 0_{n\times m} & 0_{n\times n} & \frac{dH_{33}^{(i)}(\theta)}{d\theta} \end{bmatrix}$$

$$= \begin{bmatrix} A_i & G_i & 0_{m\times n} \\ 0_{n\times m} & B_i & I_n \\ 0_{n\times m} & 0_{n\times n} & B_i \end{bmatrix} \begin{bmatrix} H_{11}^{(i)}(\theta) & H_{12}^{(i)}(\theta) & H_{13}^{(i)}(\theta) \\ 0_{n\times m} & H_{22}^{(i)}(\theta) & H_{23}^{(i)}(\theta) \\ 0_{n\times m} & 0_{n\times n} & H_{33}^{(i)}(\theta) \end{bmatrix}$$

$$= \begin{bmatrix} A_i H_{11}^{(i)}(\theta) & A_i H_{12}^{(i)}(\theta) + G_i H_{22}^{(i)}(\theta) & A_i H_{13}^{(i)}(\theta) + G_i H_{23}^{(i)}(\theta) \\ 0_{n\times m} & B_i H_{22}^{(i)}(\theta) & B_i H_{23}^{(i)}(\theta) + H_{33}^{(i)}(\theta) \\ 0_{n\times m} & 0_{n\times n} & B_i H_{33}^{(i)}(\theta) \end{bmatrix}.$$

Equating the corresponding blocks of both members of the previous equation and considering that

$$e^{D_i\theta}\Big|_{\theta=0} = I_{m+2n},$$

the following LDEs can be obtained

$$\frac{dH_{11}^{(i)}(\theta)}{d\theta} = A_i H_{11}^{(i)}(\theta), \quad H_{11}^{(i)}(0) = I_m, \tag{27}$$

$$\frac{dH_{22}^{(i)}(\theta)}{d\theta} = B_i H_{22}^{(i)}(\theta), \quad H_{22}^{(i)}(0) = I_n, \tag{28}$$

$$\frac{dH_{33}^{(i)}(\theta)}{d\theta} = B_i H_{33}^{(i)}(\theta), \quad H_{33}^{(i)}(0) = I_n, \tag{29}$$

$$\frac{dH_{12}^{(i)}(\theta)}{d\theta} = A_i H_{12}^{(i)}(\theta) + G_i H_{22}^{(i)}(\theta), \quad H_{12}^{(i)}(0) = 0_{m\times n}, \tag{30}$$

$$\frac{dH_{23}^{(i)}(\theta)}{d\theta} = B_i H_{23}^{(i)}(\theta) + H_{33}^{(i)}(\theta), \quad H_{23}^{(i)}(0) = 0_{n\times n}, \tag{31}$$

$$\frac{dH_{13}^{(i)}(\theta)}{d\theta} = A_i H_{13}^{(i)}(\theta) + G_i H_{23}^{(i)}(\theta), \quad H_{13}^{(i)}(0) = 0_{m\times n}. \tag{32}$$

The solutions of (27), (28) and (29) are

$$H_{11}^{(i)}(\theta) = e^{A_i\theta},$$
$$H_{22}^{(i)}(\theta) = e^{B_i\theta},$$
$$H_{33}^{(i)}(\theta) = e^{B_i\theta}.$$

Since $H_{22}^{(i)}(\theta) = e^{B_i\theta}$ and $H_{33}^{(i)}(\theta) = e^{B_i\theta}$, the solutions of (30) and (31) are

$$H_{12}^{(i)}(\theta) = \int_0^\theta e^{A_i(\theta-s)} G_i e^{B_i s} ds,$$

$$H_{23}^{(i)}(\theta) = e^{B_i\theta}\theta.$$

Finally, replacing $H_{23}^{(i)}(\theta) = e^{B_i\theta}\theta$ in (32), the following equation can be obtained

$$\frac{dH_{13}^{(i)}(\theta)}{d\theta} = A_i H_{13}^{(i)}(\theta) + G_i e^{B_i\theta}\theta, \quad H_{13}^{(i)}(0) = 0_{m\times n},$$

whose solution is

$$H_{13}^{(i)}(\theta) = \int_0^\theta e^{A_i(\theta-s)} G_i e^{B_i s} s ds. \tag{33}$$

Considering the previous expressions and $H_{22}^{(i)}(\theta) = F_{22}^{(i)}(\theta) = e^{B_i\theta}$,

$$Y(t) = Y_i + [F_{12}^{(i)}(\theta) + H_{13}^{(i)}(\theta)][F_{22}^{(i)}(\theta)]^{-1}.$$

If t is replaced by t_{i+1} in the previous expression, we obtain

$$Y_{i+1} = Y_i + [F_{12}^{(i)}(\Delta t_i) + H_{13}^{(i)}(\Delta t_i)][F_{22}^{(i)}(\Delta t_i)]^{-1}, \tag{34}$$

where $\Delta t_i = t_{i+1} - t_i$. $\square$

4.1 Algorithm based on Padé approximants with scale-squaring

The (s,t) Padé approximation to e^A is defined by

$$R_{st} = [D_{st}(A)]^{-1} N_{st}(A),$$

where

$$N_{st}(A) = \sum_{k=0}^{s} p_k A^k, \quad p_k = \frac{(s+t-k)!s!}{(s+t)!k!(s-k)!} \tag{35}$$

and

$$D_{st}(A) = \sum_{k=0}^{s} q_k A^k, \quad q_k = \frac{(-1)^k (s+t-k)!s!}{(s+t)!k!(s-k)!}. \tag{36}$$

Non-singularity of $D_{st}(A)$ is assured if s and t are large enough or if the eigenvalues of A are negative. The problem with this method is that it only provides good approaches near the origin [21, p.573].

Scaling-squaring method is one of the most widely used methods for computing the matrix exponential [22,23] and avoids that problem by exploiting the equality

$$e^A = \left(e^{A/2^j}\right)^{2^j}.$$

The idea is to choose j so that $e^{A/2^j}$ can be reliably and efficiently computed, and then to form the matrix $\left(e^{A/2^j}\right)^{2^j}$ by repeated squaring. One commonly used criterion for choosing j is to select the smallest natural number such that $||A||/2^j \leq 1/2$ [21, p. 573]. There are reasons for favoring the diagonal Padé approximants ($s = t$), such as for example, their lower computational costs in comparison to the non diagonal Padé approximants.

Algorithm 1 (`exmdpa`) computes the exponential of a matrix by means of a scaling-squaring diagonal Padé approximation method.

Since diagonal blocks (1,1) and (2,2) of matrices $e^{C_i \Delta t}$ and $e^{D_i \Delta t}$ are equal, it is possible to obtain a block-oriented algorithm based on Algorithm 1 which allows to compute simultaneously $F_{12}^{(i)}(\Delta t_i)$ and $H_{13}^{(i)}(\Delta t_i)$ without explicitly forming the exponential of matrices C_i and D_i in expressions (20) and (21). This is done in Algorithm 3. Algorithm 3 (`dauvdreplpa`) is a double precision auxiliary algorithm that computes the approximate solution at t_{i+1} of time-varying DMRE (1) using a piecewise-linearized method based on Padé approximants. Lines 4-7 of this algorithm avoid overflow problems by controlling that the norms of $\Delta t_i C_i$ or $\Delta t_i D_i$ (see expressions (20) and (21)) are lower than a prefixed constant M.The approximate computational cost of this

Algorithm 1 computes a matrix exponential by a scaling-squaring method based on diagonal Padé approximants.

Function $F = \texttt{exmdpa}(A, s)$
Inputs: Matrix $A \in \mathbb{R}^{n\times n}$; order $s \in \mathbb{N}$ of the diagonal Padé approximation of the exponential function
Output: Matrix $F = e^A \in \mathbb{R}^{n\times n}$

Compute the vectors of coefficients p and q (expressions (35) and (36)) of the diagonal Padé approximants of the exponential function ($p_0 = 1, q_0 = 1$)
$nor = ||A||_\infty$
$j = \max(0, 1 + \lceil \log_2(nor) \rceil)$
$A = A/2^j$
$X = A$
$N = I_n + p_1 X$
$D = I_n + q_1 X$
for $k = 2 : s$ **do**
    $X = XA$
    $N = N + p_k X$
    $D = D + q_k X$
end for
Solve $DF = N$ for F using Gaussian elimination
for $k = 1 : j$ **do**
    $F = F^2$
end for

algorithm is $(2m^3 + 6m^2n + 6mn^2 + 4n^3)s + (2m^3 + 6m^2n + 8mn^2 + 6n^3)j + \frac{2}{3}m^3 + 2m^2n + \frac{10}{3}n^3$ flops.

Algorithm 2 (`dgevdreplpa`) solves, for double precision general matrices, time-varying DMREs by a piecewise-linearized method based on Padé approximants. The approximate cost by iteration of this algorithm is $8m^2n + 6mn^2 +$ cost(`data`) + `cost`(`datad`) + cost(`Algorithm` 3) flops.

Algorithms 2 and 3 can easily be adapted for time-invariant DMREs : it is sufficient to evaluate the coefficient matrices A_{ij} once, consider $\dot{A}_{22} = 0$, and therefore eliminate matrix G from these algorithms. The adapted algorithms for time-invariant will be denoted as `dauidreplpa` and `dgeidreplpa` respectively.

5 Numerical experiments

In this section the algorithms shown in the previous sections are compared with the algorithms presented in [14]. The implementations were tested on

Algorithm 2 solves time-varying DMREs by means of a piecewise-linearized method based on diagonal Padé approximants of the exponential function.

Function $\{Y_i\} = \texttt{dgevdreplpa}(\text{data}, \text{datad}, t_0, X_0, t_f, \Delta t, s)$

Inputs: Function $\texttt{data}(\tau)$ that computes the matrices $A_{11}(\tau) \in \mathbb{R}^{n\times n}$, $A_{12}(\tau) \in \mathbb{R}^{n\times m}$, $A_{21}(\tau) \in \mathbb{R}^{m\times n}$ and $A_{22}(\tau) \in \mathbb{R}^{m\times m}$, $(\tau \in \mathbb{R})$, of time-varying DMRE (1); function $\texttt{datad}(\tau)$ that computes the derivatives $\dot{A}_{11}(\tau) \in \mathbb{R}^{n\times n}$, $\dot{A}_{12}(\tau) \in \mathbb{R}^{n\times n}$, $\dot{A}_{21}(\tau) \in \mathbb{R}^{n\times n}$ and $\dot{A}_{22}(\tau) \in \mathbb{R}^{m\times m}$ $(\tau \in \mathbb{R})$ of the above matrices; initial condition (t_0, X_0), $t_0 \in \mathbb{R}$, $X_0 \in \mathbb{R}^{m\times n}$; final time $t_f \in \mathbb{R}$; step size $\Delta t \in \mathbb{R}$; order $s \in \mathbb{N}$ of the diagonal Padé approximation of the exponential function

Outputs: Solution matrices $\{Y_i\}$ $(Y_i \in \mathbb{R}^{m\times n})$ at $t_0, t_0 + \Delta t, t_0 + 2\Delta t, \ldots$

1: Compute coefficient vectors p and q of (35) and (36) ($p_0 = q_0 = 1$ are not computed)
2: $l = \lceil (t_f - t_0)/\Delta t \rceil$
3: $Y_0 = X_0$
4: **for** $i = 0$ to $l - 1$ **do**
5: $\quad [\dot{A}_{11}, \dot{A}_{12}, \dot{A}_{21}, \dot{A}_{22}] = \texttt{datad}(t_i)$
6: $\quad \dot{A}_{22} = \dot{A}_{22} - Y_i \dot{A}_{12}$
7: $\quad G = \dot{A}_{21} + \dot{A}_{22} Y_i$
8: $\quad G = G - Y_i \dot{A}_{11}$
9: $\quad [A_{11}, A_{12}, A_{21}, A_{22}] = \texttt{data}(t_i)$
10: $\quad A_{22} = A_{22} - Y_i A_{12}$
11: $\quad F = A_{21} + A_{22} Y_i$
12: $\quad F = F - Y_i A_{11}$
13: $\quad A_{11} = A_{11} + A_{12} Y_i$
14: $\quad Y_{i+1} = \texttt{dauvdreplpa}(A_{22}, A_{11}, F, G, Y_i, \Delta t, p, q)$ ▷ Algorithm 3
15: $\quad t_{i+1} = t_i + \Delta t$
16: **end for**

an Intel Core 2 Duo T9400 at 2.52 GHz with 4 GB main memory, using 7.7 (R2008b) MATLAB version. The implemented algorithms are available online at [24].

To test the algorithms a set of six case studies were considered, all with well-known solutions. For each case study, the values of parameters which offer better accuracy and lower computational cost were determined. Three kinds of tests were carried out varying Δt, t_f and the dimension of the problem.

In all case studies, the following results are shown:

- Tables which contain the relative error

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

where X^* is the computed solution and X is the analytic solution.

Algorithm 3 computes the approximate solution of DMRE (1) at t_{i+1}.

Function $Y_{i+1} = \texttt{dauvdreplpa}(A_i, B_i, F_i, G_i, Y_i, \Delta t_i, p, q)$

Inputs: Matrices $A_i \in \mathbb{R}^{m\times m}$, $B_i \in \mathbb{R}^{n\times n}$, $F_i \in \mathbb{R}^{m\times n}$, $G_i \in \mathbb{R}^{m\times n}$ and $Y_i \in \mathbb{R}^{m\times n}$; step size $\Delta t_i \in \mathbb{R}$; vectors $p, q \in \mathbb{R}^s$ which contain the coefficients $p_1, p_2, ..., p_s, q_1, q_2, ..., q_s$ of the (s, s) Padé approximation of the exponential function

Output: Matrix $Y_{i+1} \in \mathbb{R}^{m\times n}$ of (34)

$M = \log(realmax)/10$ ▷ $realmax$ is the largest positive float point number
$norA = ||A_i||_\infty$; $norB = ||B_i||_\infty$; $norF = ||F_i||_\infty$
$nor = \Delta t_i \max(norA + \max(norF, norG), norB + 1)$
if $nor > M$ **then**
    $Y_{i+1} = \texttt{dauvdreplce}(A_i, B_i, F_i, G_i, Y_i, \Delta t_i, p, q)$ ▷ Algorithm 5 of [14]
    return
end if
$j = \max(0, 1 + \lceil \log_2(nor) \rceil)$; $t = \frac{\Delta t_i}{2^j}$
$A_i = tA_i$;$B_i = tB_i$;$F_i = tF_i$; $G_i = tG_i$
$X_{11} = A_i$; $X_{12} = G_i$; $Y_{12} = F_i$
$X_{22} = B_i$; $X_{13} = 0_{m\times n}$; $X_{23} = tI_n$
$N_{11} = I_m + p_1 X_{11}$; $N_{12} = p_1 X_{12}$;$M_{12} = p_1 Y_{12}$
$N_{13} = p_1 X_{13}$; $N_{22} = I_n + p_1 X_{22}$; $N_{23} = p_1 X_{23}$
$D_{11} = I_m + q_1 X_{11}$; $D_{12} = q_1 X_{12}$;$P_{12} = q_1 Y_{12}$
$D_{13} = q_1 X_{13}$; $D_{22} = I_n + q_1 X_{22}$; $D_{23} = q_1 X_{23}$
for $k = 2$ to s **do**
    $X_{11} = A_i X_{11}$; $X_{12} = A_i X_{12} + G_i X_{22}$; $Y_{12} = A_i Y_{12} + F_i X_{22}$
    $X_{13} = A_i X_{13} + G_i X_{23}$; $X_{23} = B_i X_{23} + sX_{22}$; $X_{22} = B_i X_{22}$
    $N_{11} = N_{11} + p_k X_{11}$; $N_{12} = N_{12} + p_k X_{12}$; $M_{12} = M_{12} + p_k Y_{12}$
    $N_{13} = N_{13} + p_k X_{13}$; $N_{22} = N_{22} + p_k X_{22}$; $N_{23} = N_{23} + p_k X_{23}$
    $D_{11} = D_{11} + q_k X_{11}$; $D_{12} = D_{12} + q_k X_{12}$; $P_{12} = P_{12} + q_k Y_{12}$
    $D_{13} = D_{13} + q_k X_{13}$; $D_{22} = D_{22} + q_k X_{22}$; $D_{23} = D_{23} + q_k X_{23}$
end for
Solve $D_{11}F_{11} = N_{11}$ for F_{11} using the LU decomposition
Solve $D_{22}F_{22} = N_{22}$ for F_{22} using the LU decomposition
Solve $D_{11}F_{12} = N_{12} - D_{12}F_{22}$ for F_{12} LU decomposition
Solve $D_{11}G_{12} = M_{12} - P_{12}F_{22}$ for G_{12} using the LU decomposition
Solve $D_{22}F_{23} = N_{23} - D_{23}F_{22}$ for F_{23} using the LU decomposition
Solve $D_{11}F_{13} = N_{13} - D_{12}F_{23} - D_{13}F_{22}$ for F_{13} using the LU decomposition
for $k = 1$ to j **do**
    $F_{13} = F_{11}F_{13} + F_{12}F_{23} + F_{13}F_{22}$
    $F_{12} = F_{11}F_{12} + F_{12}F_{22}$
    $G_{12} = F_{11}G_{12} + G_{12}F_{22}$; $F_{23} = F_{22}F_{23} + F_{23}F_{22}$
    $F_{11} = F_{11}^2$; $F_{22} = F_{22}^2$
end for
Solve $Y_{i+1}F_{22} = G_{12} + F_{13}$ for Y_{i+1} using the LU decomposition

- Tables/graphics with execution times (Te) in seconds.

Below is a short description of compared algorithms:

- `dgevdreplpa` (Algorithm 2) and `dgeidreplpa`: They solve time-varying and time-invariant DMREs by the piecewise-linearized method based on the diagonal Padé approximants presented in this paper.
- `dgevdreplce` (Algorithm 7 of [14]) and `dgeidreplce`: They solve time-varying and time-invariant DMREs by the piecewise-linearized method based on the Conmutant Equation explained in Section 3.

5.1 Case Study 1

The first time-invariant DMRE is taken from a two-point boundary value problem [25]. This DMRE is defined for $t \geq 0$ by the coefficient matrices

$$A_{11} = \begin{bmatrix} 0 & 0 \\ -100 & -1 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} 0 & 1 \\ 100 & 0 \end{bmatrix},$$
$$A_{21} = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} 0 & 0 \\ -10 & -1 \end{bmatrix},$$

and the initial condition

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

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

$$X = \begin{bmatrix} 1 & 0.11 \\ 0 & -0.1 \end{bmatrix}.$$

In this case study s=1 was used for `dgeidreplpa` and `dgeidreplce`. In the only test done, final time $t_f = 30$ was considered and Δt was varied between 0.1, 0.05 and 0.01. Tables 1 and 2 show the results.

Considering the same step size, the conclusions for this case study are:

- Both functions have similar accuracy.

Table 1
Case study 1: Relative error considering $t_f = 30$ and Δt variable.

Er	Δt=0.1	Δt=0.05	Δt=0.01
`dgeidreplpa`	3.243e-14	7.760e-15	8.588e-16
`dgeidreplce`	3.180e-14	6.671e-15	8.618e-16

Table 2
Case study 1: Execution time considering $t_f = 30$ and Δt variable.

Te	Δt=0.1	Δt=0.05	Δt=0.01
`dgeidreplpa`	0.034	0.058	0.293
`dgeidreplce`	0.197	0.364	1.817

- Relative errors decreased as Δt decreased.
- `dgeidreplpa` has the shorter execution time.

5.2 Case Study 2

The second case study [26,7] consists of the following time-invariant DMRE

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

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

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

For this case study we considered s=1 both for `dgeidreplpa` and `dgeidreplce`. Table 3 and Figure 1 show execution times for $\alpha = 100$ (stiff problem), $\Delta t = 0.1$, $t_f = 1$ and the dimension of problem equal to 50, 100, 150 and 200. For two implementations, the relative errors were 0.

Considering the same step size, the conclusions for this case study are:

- Both functions achieved very high accuracy.
- `dgeidreplce` execution times are longer than `dgeidreplpa` execution times.

5.3 Case Study 3

The third case study [27] consists of the following time-invariant DMRE

Table 3
Case study 2: Execution time considering $\Delta t = 0.1$, $t_f = 1$ and n variable.

Te	n=50	n=100	n=150	n=200
`dgeidreplpa`	0.984	3.989	8.990	16.394
`dgeidreplce`	2.941	11.709	26.293	47.282

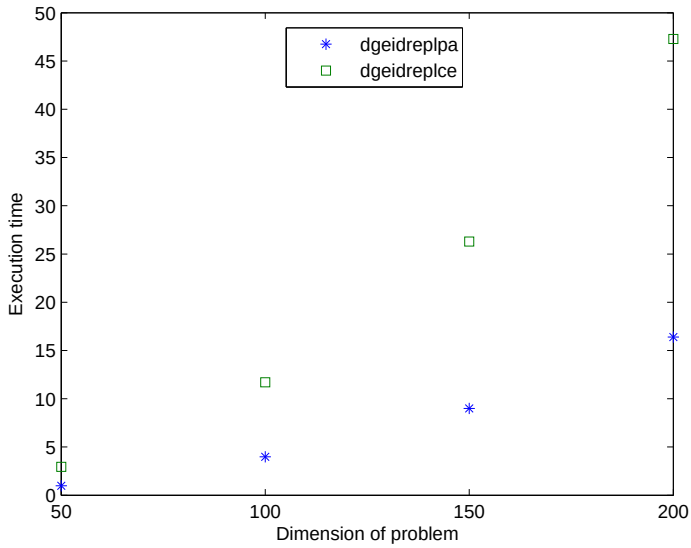

Fig. 1. Case study 2: Execution time considering $\Delta t = 0.1$, $t_f = 2$ and variable dimension of problem .

$$\dot{X}(t) = \alpha T_{2^k} + T_{2^k} X(t) + X(t) T_{2^k} - X(t) T_{2^k} X(t), \quad t \geq 0 \ ,$$
$$X(t_0) = X_0,$$

where $X(t), T_{2^k} \in \mathbb{R}^{2^k \times 2^k}$ and $\alpha \in \mathbb{R}^+$.

The matrices T_{2^k} are generated recursively as follows:

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

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

The solution is given by

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

Table 4
Case study 3: Relative error (Er) considering $\Delta t = 0.1$, $t_f = 5$ and $m = n$ variable.

Er	n=32	n=64	n=128	n=256
`dgeidreplpa`	1.185e-16	1.999e-16	3.357e-18	7.297e-16
`dgeidreplce`	1.185e-16	1.999e-16	3.357e-18	7.297e-16

Table 5
Case study 3: Execution time (Te) considering $\Delta t = 0.1$, $t_f = 5$ and $m = n$ variable.

Te	n=32	n=64	n=128	n=256
`dgeidreplpa`	4.894	18.221	67.976	247.246
`dgeidreplce`	4.925	18.023	67.663	248.415

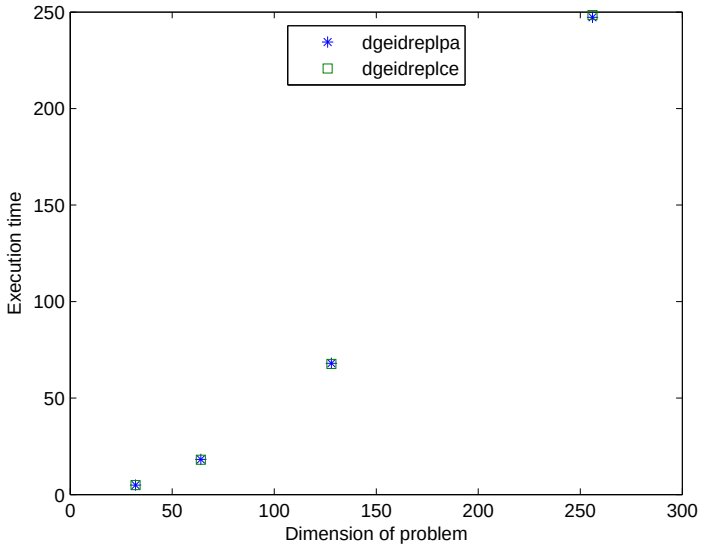

Fig. 2. Case study 3: Execution time considering $t_f = 5$, $\Delta t = 0.1$ and n variable.

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

The parameters of problem were $\alpha = 100$ (stiff problem) and $X(0) = I$. The order of Padé approximants for the two implementations was $s = 2$. In tests $t_f = 5$ was considered, varying the dimension of the problem between 32, 64, 128 and 256, and step sizes between 0.1, 0.05, 0.01, 0.005 and 0.001. Both functions achieved smaller relative error for $\Delta t = 0.1$. Tables 4 and 5 and Figure 2 show the results for $\Delta t = 0.1$. In this case study `dgeidreplpa` and `dgeidreplce` achieved very high accuracy for $\Delta t = 0.1$ with similar execution times.

Table 6
Case study 4: Relative error (Er) with $n = 1$, $\Delta t = 0.1$ and t_f variable.

Er	t_f=10	t_f=20	t_f=30	t_f=40	t_f=50
`dgevdreplpa`	4.999e-7	2.500e-7	1.667e-7	1.250e-7	1.000e-7
`dgevdreplce`	4.999e-7	2.500e-7	1.667e-7	1.250e-7	1.000e-7

Table 7
Case study 4: Execution time (Te) with $n = 1$, $\Delta t = 0.1$ and t_f variable.

Te	t_f=10	t_f=20	t_f=30	t_f=40	t_f=50
`dgevdreplpa`	0.063	0.121	0.178	0.235	0.293
`dgevdreplce`	0.061	0.116	0.173	0.226	0.283

5.4 Case Study 4

This scalar time-varying DMRE is a widely used for testing stiff problems, known as the "knee problem" ([28,7]), defined as

$$\dot{x} = 1 - \frac{t}{\varepsilon}x + \frac{x^2}{\varepsilon}, -1 \leq t \leq 1, \quad x(-1) = -1, \quad 0 < \varepsilon << 1,$$

associated to the coefficient matrix

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix} = \begin{bmatrix} t/\varepsilon & -1/\varepsilon \\ 0.5 & 0 \end{bmatrix}, \quad n = m = 1.$$

The reduced solution $x \cong t$ is stable before 0 and $x \cong 0$ is stable past it.

In the tests done $\varepsilon = 10^{-5}$ (stiff problem) and the same order of the diagonal Padé approximants ($s = 1$) was considered. Both functions achieved smaller relative error for $\Delta t = 0.1$. Tables 6 and 7 and Figure 3 show relative errors and execution times for $\Delta t = 0.1$ and t_f variable. In this case study, `dgevdreplpa` and `dgevdreplce` achieved smaller relative error for $\Delta t = 0.1$, with similar execution times.

5.5 Case study 5

This stiff time-varying DMRE [29,7] comes from a stiff two-point boundary value problem. This DMRE is defined as

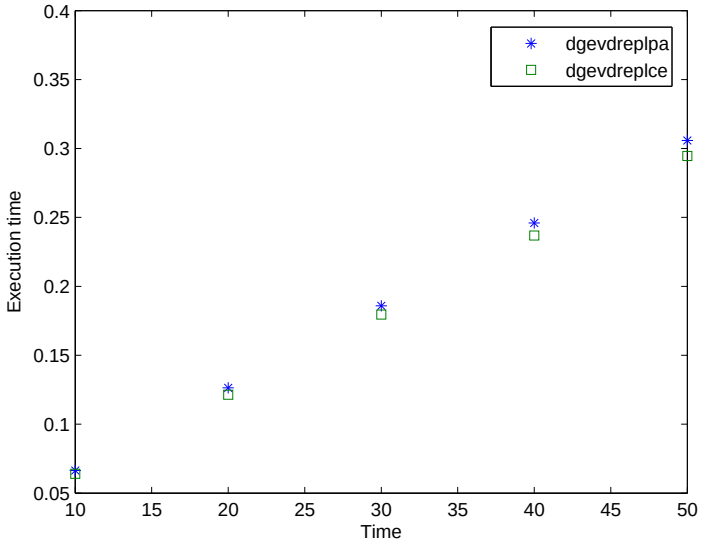

Fig. 3. Case study 4: Execution time considering $\Delta t = 0.001$ and t_f variable.

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

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

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

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

The solution has an initial layer and then it approaches

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

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

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

In the tests $\varepsilon = 10^{-5}$ (stiff problem), and the same order of the diagonal Padé

Table 8
Case study 5: Relative error (Er) with $n = 2$, $\Delta t = 0.1$ and t_f variable.

Er	t_f=10	t_f=20	t_f=30	t_f=40	t_f=50
`dgevdreplpa`	8.668e-20	1.776e-16	2.891e-20	2.168e-20	1.421e-16
`dgevdreplce`	8.668e-20	1.776e-16	2.891e-20	2.168e-20	1.421e-16

Table 9
Case study 5: Execution time (Te) with $n = 2$, $\Delta t = 0.1$ and t_f variable.

Te	t_f=10	t_f=20	t_f=30	t_f=40	t_f=50
`dgevdreplpa`	0.178	0.343	0.508	0.672	0.835
`dgevdreplce`	0.177	0.342	0.502	0.667	0.829

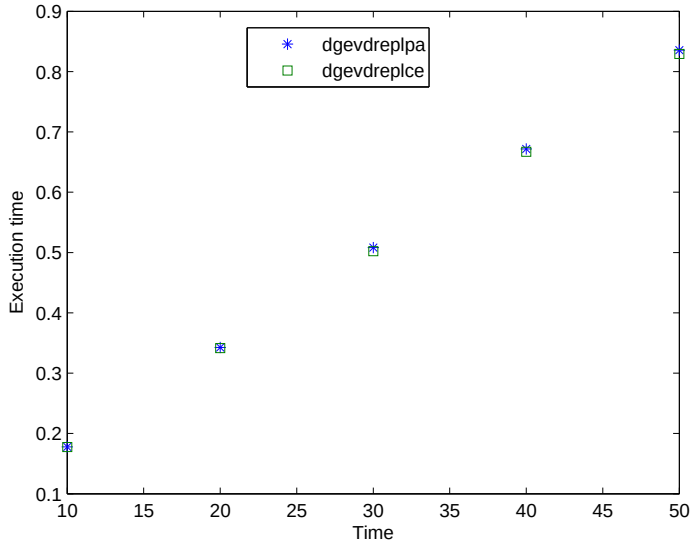

Fig. 4. Case study 5: Execution time considering $\Delta t = 0.1$ and t_f variable.

approximants ($s = 1$) was considered. In the tests t_f was variable with step size $\Delta t = 0.1$. Tables 8 and 9 and Figure 4 show that both implementations achieved high accuracy with a similar execution time.

5.6 Case study 6

This equation corresponds to a time-varying DMRE [27] defined as

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

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

Table 10
Case study 6: Relative error (Er) considering $n = 8$, $t_f = 5$ and Δt variable.

Er	Δt=0.1	Δt=0.05	Δt=0.01	Δt=0.005	Δt=0.001
`dgevdreplpa`	1.209e-02	4.014e-03	1.958e-04	5.000e-05	2.034e-06
`dgevdreplce`	1.620e-02	4.134e-03	1.967e-04	5.006e-05	2.035e-06

Table 11
Case study 6: Execution time (Te) considering $n = 8$, $t_f = 5$ and Δt variable.

Te	Δt=0.1	Δt=0.05	Δt=0.01	Δt=0.005	Δt=0.001
`dgevdreplpa`	0.021	0.043	0.228	0.418	2.092
`dgevdreplce`	0.328	0.668	3.414	6.748	33.389

Table 12
Case Study 6: Relative error (Er) considering $\Delta t = 0.01$, $t_f = 5$ and n variable.

Er	n=8	n=16	n=32	n=64
`dgevdreplpa`	1.958e-04	1.959e-04	1.962e-04	1.970e-04
`dgevdreplce`	1.967e-04	1.983e-04	2.014e-04	2.068e-04

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

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

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

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

In this case study an order of Padé approximants $s = 2$ was selected. Tables 10 and 11 show the results for $n = m = 16$ (k=4), t_f=1 and Δt variable. Tables 12 and 13 and Figure 5 show the results for Δt=0.01, $t_f = 5$ and dimension of problem variable. The following conclusions can be emphasized:

- Considering the same step size, both implementations have similar accuracy, but `dgevdreplpa` has the shorter execution time.
- For both implementations, relative error decreased as Δt decreased.
- As dimension of problem is increased `dgevdreplce` execution time increased quicker than `dgevdreplpa` execution time: For $n = 8$ the execution time ratio is $\frac{3.433}{0.212} \cong 16.193$ and for $n = 64$ is $\frac{285.013}{2.381} \cong 119.703$.

Table 13
Case Study 6: Execution time (Te) considering $\Delta t = 0.01$, $t_f = 5$ and n variable.

Te	n=8	n=16	n=32	n=64
`dgevdreplpa`	0.212	0.311	0.621	2.381
`dgevdreplce`	3.433	19.593	54.235	285.013

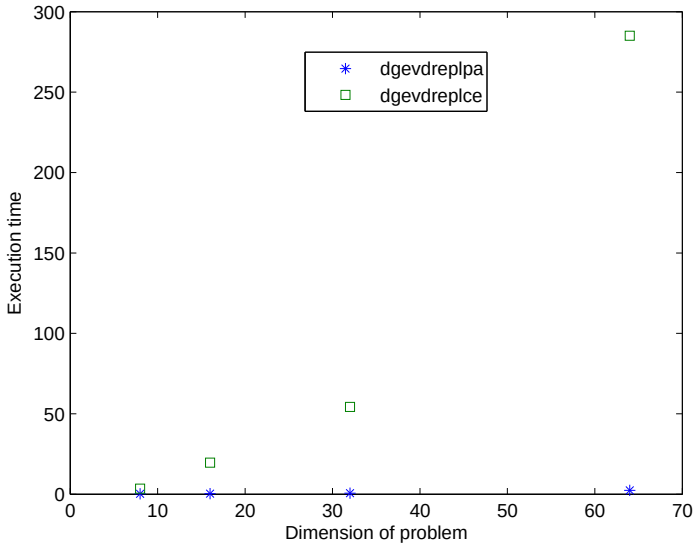

Fig. 5. Case Study 6: Execution time considering $t_f = 5$, Δt=0.01 and variable dimension of problem.

Table 14
Comparison of execution times for the six case studies: the symbols $+$, $\cong$ and $-$ indicate longer, similar and shorter execution time. The symbols S and NS indicate stiff and non-stiff problem.

Case Study	1: NS	2: S	3: S	4: S	5: S	6: NS
`dgeidreplpa-dgevdreplpa`	$+$	$+$	$\cong$	$\cong$	$\cong$	$+$
`dgeidreplce-dgevdreplce`	$-$	$-$	$\cong$	$\cong$	$\cong$	$-$

5.7 Summary of results

Table 14 show a comparison of execution times for the six case studies analyzed, when the implementations have similar accuracy. For each case study, problem stiffness (S= stiff problem, NS=non-stiff problem) is indicated.

- In general, for the same step size, the relative errors of all implementations were similar.
- In three case studies, `dgeidreplpa-dgevdreplpa` execution times were lower than `dgeidreplce-dgevdreplce` execution times.

- All implementations showed a good behavior in stiff problems.

6 Conclusions and future work

In this paper a method for solving DMREs has been developed. This method is based on Theorem 3 in Section 4 which allows an efficient computation of the integral that appears in the piecewise-linearized methods.

Two MATLAB implementations have been developed based on the piecewise-linearized method developed in Section 4. In order to verify the benefits of these implementations, numerous tests were made on six case studies, comparing, under equal conditions all implementations.

Possible future lines of research are:

- To develop other methods to solve DMREs based on the piecewise-linearized approach. A possibility consists in computing the product of a matrix exponential by a vector using Krylov subspaces (this case will be suitable for higher dimension problems).
- To include adaptive selection of the step size in the algorithms developed in this paper.
- To adapt the implementations for special DMREs such as DMREs with sparse coefficient matrices.

References

[1] D. R. Vaughan, A negative exponential solution for the matrix Riccati equation, IEEE Trans. on Automatic Control 14 (1) (1969) 72–75.

[2] E. J. Davison, M. C. Maki, The numerical solution of the matrix Riccati differential equation, IEEE Trans. on Automatic Control 18 (1) (1973) 71–73.

[3] C. S. Kenney, R. B. Leipnik, Numerical integration of the differential matrix Riccati equation, IEEE Trans. on Automatic Control 30 (1985) 962–970.

[4] H. Chandrasekhar, Generalized Chandrasekhar algorithms: Time-varying models, IEEE Trans.on Automat. Control 21 (1976) 728–732.

[5] D. W. Rand, P. Winternitz, Nonlinear superposition principles: A new numerical method for solving matrix Riccati equations, Comp. Phys. Commun. 33 (1984) 305–328.

[6] M. Sorine, P. Winternitz, Superposition laws for the solution of differential Riccati equations, IEEE Trans. Automat. Contr. 30 (1985) 266–272.

[7] L. Dieci, Numerical integration of the differential Riccati equation and some related issues, SIAM Journal on Numerical Analysis. 29 (3) (1992) 781–815.

[8] C. H. Choi, Efficient algorithms for solving stiff matrix-valued Riccati differential equations, Ph.D. thesis, University of California (1988).

[9] P. Benner, H. Mena, BDF methods for large-scale differential Riccati equations, in: Sixteenth International Symposium on Mathematical Theory of Network and Systems (MTNS2004), Katholieke Universiteit Leuven, Belgium, 2004.

[10] E. Arias, V. Hernández, J. Ibáñez, J. Peinado, A fixed point-based BDF method for solving Riccati equations, Applied Mathematics and Computation 188 (2) (2007) 1319–1333.

[11] V. Hernández, J. Ibáñez, E. Arias, J. Peinado, A GMRES-based BDF method for solving differential Riccati equations, Applied Mathematics and Computation 196 (2) (2008) 613–626.

[12] J. M. Sanz-Serna, Symplectic integrators for hamiltonian problems: an overview, Acta Numerica 1 (1992) 243–286.

[13] L. Ren-Cang, Unconventional reflexive numerical methods for matrix differential Riccati, Tech. Rep. 2000-36, Department of Mathematics, University of Kentucky (2000).

[14] J. Ibáñez, V. Hernández, Solving differential matrix Riccati equations by a piecewise-linearized method based on the conmutant equation, Computer Physics Communications 180 (2010) 2103–2114.

[15] J. Ibáñez, V. Hernández, E. Arias, P. Ruiz, Solving initial value problems for ordinary differential equations by two approaches: BDF and piecewise-linearized methods, Computer Physics Communications 180 (5) (2009) 712–723.

[16] J. I. Ramos, C. M. García, Piecewise-linearized methods for initial-value problems, Applied Mathematics and Computation 82 (1997) 273–302.

[17] C. M. García, Piecewise-linearized and linearized θ-methods for ordinary and partial differential equation problems, Computer & Mathematics with Applications 45 (2003) 351–381.

[18] C. M. García, Métodos de linealización para la resolución numérica de ecuaciones diferenciales, Ph.D. thesis, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga (1998).

[19] R. A. Horn, C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, London, 1991.

[20] J. Ibáñez, V. Hernández, Solving differential matrix Riccati equations by a piecewise-linearized method based on diagonal Padé approximants, Computer Physics Communications(To appear).

[21] G. H. Golub, C. V. Loan, Matrix Computations, 3rd Edition, Johns Hopkins Studies in Mathematical Sciences, The Johns Hopkins University Press, 1996.

[22] C. B. Moler, C. V. Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review 45 (2003) 3–49.

[23] N. J. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM J. Matrix Anal. Appl. 26 (4) (2005) 1179–1193.

[24] http://www.grycap.upv.es/dmretoolbox.

[25] S. Pruess, Interpolation schemes for collocation solution of TPBVPs, SIAM Journal on Scientific and Statistical Computing 7 (1986) 322–333.

[26] G. H. Meyer, Initial Value Methods for Boundary Value Problems, Academic Press, New York, 1973.

[27] C. H. Choi, Time-varying Riccati differential equations with known analytic solutions, IEEE Trans. Automat. Contr. 37 (1992) 642–645.

[28] G. Dahlquist, L. Edsberg, G. Sköllermo, G. Söderlind, Are the Numerical Methods and Software Satisfactory for Chemical Kinetics?, in Numerical Integration of DE and Large Linear Systems, Vol. 968/1992 of Lecture Notes in Computer Mathematics, Springer Berlin / Heidelberg, 1982, pp. 149–164.

[29] D. L. Brown, J. Lorenz, A high-order method for stiff boundary value problems with turning points, SIAM Journal on Scientific and Statistical Computing 8 (1987) 790–805.