Noname manuscript No.
(will be inserted by the editor)

High order structure preserving explicit methods for solving linear-quadratic optimal control problems

Sergio Blanes

Received: date / Accepted: date

Abstract We consider the numerical integration of linear-quadratic optimal control problems. This problem requires the solution of a boundary value problem: a non-autonomous matrix Riccati differential equation (RDE) with final conditions coupled with the state vector equation with initial conditions. The RDE has positive definite matrix solution and to numerically preserve this qualitative property we propose first to integrate this equation backward in time with a sufficiently accurate scheme. Then, this problem turns into an initial value problem, and we analyse splitting and Magnus integrators for the forward time integration which preserve the positive definite matrix solutions for the RDE. Duplicating the time as two new coordinates and using appropriate splitting methods, high order methods preserving the desired property can be obtained. The schemes make sequential computations and do not require the storrage of intermediate results, so the storage requirements are minimal. The proposed methods are also adapted for solving linear-quadratic N-player differential games. The performance of the splitting methods can be considerably improved if the system is a perturbation of an exactly solvable problem and the system is properly split. Some numerical examples illustrate the performance of the proposed methods.

Keywords Geometric Numerical Integration · Splitting methods · matrix Riccati differential equations · LQ optimal control problems · Differential games

Sergio Blanes
Instituto de Matemática Multidisciplinar, Universitat Politècnica de València.
46022 Valencia, Spain
E-mail: serblaza@imm.upv.es

1 Introduction

Linear-quadratic (LQ) optimal control problems appear in many different fields in engineering [1,3,33,36] as well as in quantum mechanics [31,32,38] (see also [14] and references therein). In general, LQ optimal control problems are described by coupled systems of nonlinear differential equations with boundary conditions. Matrix Riccati differential equations (RDEs) with positive definite solutions, coupled with the state vector equation, have to be solved in some mechanical systems. In quantum mechanics, coupled nonlinear systems of Schrödinger equations for the wave function and a Lagrange multiplier have to be solved [38]. The particular algebraic structure of the equations in each problem makes that, in general, the solutions have some qualitative properties which are relevant for the theoretical study (positive definiteness in the solution of the RDE or the evolution through a unitary transformation in quantum mechanics). It seems natural to look for numerical schemes that preserve these relevant qualitative properties in order to get reliable and accurate results.

Numerical methods for solving nonlinear BVPs are usually more complicated (and computationally more costly) than initial value problems IVPs [4, 26,30]. We consider the numerical integration of LQ optimal control problems with high order explicit and structure preserving methods and we analyse the preservation of the positive definiteness in the numerical solution of the RDE.

It is important to remark that one step and multistep schemes of order greater than one used as direct algorithms cannot guarantee that the solution stays positive definite [16]. On the other hand, the RDE has an associated Hamiltonian system which can be solved using appropriate symplectic integrators [18,34]. These methods can be seen as indirect algorithms, allowing for positive definite higher order approximations. We analyse splitting methods and Magnus integrators as symplectic integrators to solve the RDE coupled with the state vector equation both for the autonomous ans non-autonomous case. We show that some constraints on the coefficients of the methods are required to guarantee the preservation of the qualitative properties. The proposed methods are explicit and the algorithms can be used with variable step as well as variable order. We use different methods for the forward and backward integration of the RDE and this provides an estimation of the accuracy of the solution.

The problem can be reformulated as an IVP where the vector field can be split into solvable parts and high order splitting methods can be used. If the problem is explicitly time dependent or it can be considered as a small perturbation of an exactly solvable problem, the correct split and the splitting methods to be used require a more careful analysis. These methods can also be considered as exponential methods which have shown a high performance for linear problems [9,10,21,22]. Implicit Runge–Kutta methods for linear problems can be considered as rational approximations to exponential integrators. Exponential methods are explicit methods suitable for the numerical integration of stiff and oscillatory problems (see [19] and references therein). Obviously, this requires the computation of exponential of matrices or their action on vectors [2,29,35].

The proposed methods are also adjusted for solving linear-quadratic N-player differential games (extensively studied from the theoretical point of view [3,6,15,17,37]) since they can be considered as optimal linear control problems.

The paper is organized as follows. In Section 2 we introduce the equations to be solved in a LQ optimal control problem and present the nonlinear matrix RDE as an equivalent linear system of equations so, indirect methods can be ued. In Section 3 we formulate the problem as an IVP with a previous backward time integration of the Riccati equation, we present a brief introduction to splitting methods and we show how these methods can be used on autonomous and non-autonomous problems. The case of perturbed systems and the preservation of the positive definiteness of the solution of the RDE by splitting methods is also analysed. Section 4 considers the generalization to the numerical integration of differential games, and Section 5 is devoted to numerical experiments for illustrating the performance of the methods. Finally, Section 6 gives the conclusions of the work.

2 LQ optimal control problems

Let us consider the LQ optimal control problem

$$x' = A(t)x + B(t)u(t), \qquad x(0) = x_0\,, \qquad 0 = t_0 \le t \le t_f = T\,, \tag{1}$$

where the unknown $x(t) \in \mathbb{R}^n$ is the dynamic state. Here $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times r}$, and $u(t) \in \mathbb{R}^r$ is the control.

We consider a quadratic cost function given by

$$J = x^T(T)\, Q_T\, x(T) + \int_0^T \left\{ x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) \right\} dt\,,$$

where $Q(t), Q_T \in \mathbb{R}^{n\times n}$, are symmetric non-negative matrices, $R(t) \in \mathbb{R}^{r\times r}$ is symmetric and positive definite (i.e. $Q(t), Q_T \ge 0,\ R > 0$) and z^T denotes the transpose of z. It is well known that the optimal control is reached when u is written as [1,3,36]

$$u(t) \;=\; -\,R^{-1}(t)\, B^T(t)\, P(t)\, x(t)\,, \tag{2}$$

with $P(t) \in \mathbb{R}^{n\times n}$ satisfying the matrix RDE with final conditions

$$P' = -Q(t) - A^T(t)P - PA(t) + PS(t)P, \qquad P(T) = Q_T, \tag{3}$$

wherein $S(t) \;=\; B(t)\, R^{-1}(t)\, B^T(t)$ is a symmetric $n\times n$ matrix and $S(t) \ge 0$. It is known that $P(t)$ is also a symmetric and non-negative matrix. Substituting (2) into (1) we have that

$$x' = \Big(A(t) - S(t)P(t)\Big)x, \qquad x(0) = x_0\,. \tag{4}$$

Since $P(t)$ is symmetric and non-negative, we can decompose it as $P(t) = G(t)G(t)$ where $G(t)$ is also symmetric and non-negative. Then, $G(t)S(t)G(t)$ is symmetric and non-negative with real and non-negative eigenvalues. Since $S(t)P(t)$ and $G(t)S(t)G(t)$ share the same eigenvalues, we have that the product $S(t)P(t)$ has real and non-negative eigenvalues [20]. This result is of relevance for the stability of (4). It is then natural to look for numerical schemes preserving the positive definiteness of the solution of the RDE.

The goal is to numerically solve the coupled system of nonlinear differential equations with boundary conditions (3) and (4). The numerical solution of nonlinear BVPs is usually computationally costly with significant storage requirements (one has to numerically integrate backward in time some of the equations, to store intermediate results and to make a forward integration using the stored results, or to use shooting methods [4,30]).

If we first integrate backward in time the RDE, the coupled system turns into an IVP to be integrated forward in time. We show that the qualitative properties can be preserved using appropriate splitting and Magnus integrators.

2.1 The matrix Riccati differential equation

It is well known that given the autonomous matrix RDE (3), if the time-dependent matrices $Q(t), A(t), S(t)$ are continuous, $Q(t), S(t)$ are symmetric and non negative for $t \in [0, T]$ and Q_T is a symmetric non negative matrix, the solution $P(t)$ is also a symmetric matrix and $P(t) \geq 0, \ t \in [0, T]$. This is a very important property because the matrix $P(t)$ is coupled with the equation for the state vector and plays an important role. However, from [16] we have the following result:

Theorem 1 *Any one-step method or strictly stable multistep method that preserves positive definiteness in the numerical solution of the RDE (3) has order at most one.*

Alternatively, we can use indirect methods for solving the RDE (3). If we consider the following decomposition $PU = V$, with $U(t), V(t) \in \mathbb{R}^{n\times n}$ and denote

$$y(t) = \begin{bmatrix} U \\ V \end{bmatrix}; \quad K(t) = \begin{bmatrix} A(t) & -S(t) \\ -Q(t) & -A^T(t) \end{bmatrix},$$

where $y(t) \in \mathbb{R}^{2n\times n}, \ K(t) \in \mathbb{R}^{2n\times 2n}$, then, it is easy to check that $y(t)$ is the solution of the ODE

$$y'(t) = K(t)\, y(t); \qquad y(T) = \begin{bmatrix} U(T) \\ V(T) \end{bmatrix} = \begin{bmatrix} I \\ Q_T \end{bmatrix}, \tag{5}$$

with conditions at the end of the interval, to be integrated backward in time. By [17,24], if (5) has an appropriate solution with $U(t)$ non singular, the solution of (3) can be calculated by solving the linear system of equations:

$P(t)\,U(t) = V(t)$. Conditions under which U^{-1} exists are known (see [1,23] and references therein). In this work we assume $U(t)$ is non-singular, otherwise $P(t)$ would be unbounded and the equation for the state vector would not be well defined.

Since eq. (3) has an associated linear Hamiltonian system, the fundamental solution is a symplectic transformation. From [16] we have also the following result:

Theorem 2 *Application of a symplectic Runge-Kutta method given by the coefficients, $a_{i,j}, b_i,\ i,j = 1,\dots,k$, with non-negative b_i's to eq. (5) produces (when defined) symmetric non-negative matrices P_n.*

(This Theorem needs also that $0 \le c_i = \sum_{j=1}^{i} a_{i,j} \le 1$ otherwise the matrix $K(t)$ is evaluated outside the interval $[0,T]$ where its existence or conditions $Q, S \ge 0$ are not guaranteed). In this work we consider high order splitting methods and Magnus integrators. These methods have negative coefficients and we analyse the conditions under which the schemes produce positive definite solutions. To this purpose, it is useful to consider backward error analysis. Suppose that a numerical scheme for a time step h is equivalent to the 1-flow solution of the autonomous equation

$$\frac{d}{dt}\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} A_h & -S_h \\ -Q_h & -A_h^T \end{bmatrix}\begin{bmatrix} U \\ V \end{bmatrix}. \qquad (6)$$

If S_h, Q_h are symmetric and non-negative then the numerical solution P_n is symmetric and non-negative.

3 Geometric integrators for solving LQ optimal control problems

The coupled system of equations to be solved is given by

$$\frac{d}{dt}\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} A(t) & -S(t) \\ -Q(t) & -A^T(t) \end{bmatrix}\begin{bmatrix} U \\ V \end{bmatrix}, \qquad \begin{bmatrix} U(T) \\ V(T) \end{bmatrix} = \begin{bmatrix} I \\ Q_T \end{bmatrix} \qquad (7)$$

$$\frac{dx}{dt} = \Big(A(t) - S(t)\,V(t)\,U(t)^{-1}\Big)x, \qquad x(0) = x_0\,. \qquad (8)$$

The numerical integratio of the eq. (8) in the interval $[t_n, t_n + h]$ typically requires the evaluation of the matrices $A(t), S(t), V(t), U(t)$ at a number of interior points, say $t_n + c_i h,\ i = 1,\dots,k$. However, U and V are not know at these points and have to be numerically computed with sufficient accuracy. This usually requires to solve eq. (7) for the time intervals $[t_n + c_{i-1}h, t_n + c_i h],\ i = 1,\dots,k$, making this procedure poorly efficient.

Since the equation (7) is independent of (8), in order to preserve positivity we propose to integrate backward the RDE to get $U(t_0), V(t_0)$ with sufficiently high accuracy. Intermediate solutions are not needed in this preliminary integration so, we have much freedom on the choice of the numerical method for the backward integration. This allows a fast way to get the initial conditions, $U(t_0), V(t_0)$, for the RDE.

Next, we have to integrate forward in time the system (7)-(8) with the final conditions replaced by $\left[U(t_0)^T \; V(t_0)^T\right]^T = \left[U_0^T \; V_0^T\right]^T$.

The linearized RDE and the equation for the state vector can be written in short as follows

$$\begin{aligned} v' &= M(t)v \\ x' &= N(t,\hat{v})x \end{aligned} \tag{9}$$

with $v = [U^T \; V^T]^T, \;\hat{v} = VU^{-1} = P$ and M, N are matrices of appropriate dimensions. This system of equations is separable into solvable parts and then splitting methods can be used in a simple way. Splitting methods allow to solve sequentially autonomous RDEs and linear autonomous equations for the state vector. These equations can be solved separately using methods which preserve the desired qualitative property. Let us briefly introduce the idea of splitting methods as well as the choice of the methods to be used.

3.1 Splitting Methods

Let us consider the initial value problem

$$x' = f(x) = f^{[a]}(x) + f^{[b]}(x), \qquad x_0 = x(0) \in \mathbb{R}^D \tag{10}$$

with $f : \mathbb{R}^D \longrightarrow \mathbb{R}^D$ and solution $x(t) = \varphi_t(x_0)$, and such that

$$x' = f^{[a]}(x), \qquad x(0) = x_0^a, \tag{11}$$

$$x' = f^{[b]}(x), \qquad x(0) = x_0^b, \tag{12}$$

can be integrated exactly, with solutions $x(h) = \varphi_h^{[a]}(x_0^a)$ and $x(h) = \varphi_h^{[b]}(x_0^b)$, respectively, at $t = h$, the time step. It is well known that the composition, $\chi_h = \varphi_h^{[b]} \circ \varphi_h^{[a]}$, is a first order method and

$$\psi_h = \varphi_{h/2}^{[a]} \circ \varphi_h^{[b]} \circ \varphi_{h/2}^{[a]},$$

is a symmetric second order method. Higher order methods can be obtained by a composition like

$$\psi_h = \varphi_{b_m h}^{[b]} \circ \varphi_{a_m h}^{[a]} \circ \cdots \circ \varphi_{a_2 h}^{[a]} \circ \varphi_{b_1 h}^{[b]} \circ \varphi_{a_1 h}^{[a]},$$

for appropriate choices of coefficients a_i, b_i. For simplicity, we denote the composition as : $b_m \, a_m \, \ldots \, b_1 \, a_1$. For example, an efficient fourth-order method is given by the following symmetric sequence [12] for $m = 7$ with $a_1 = 0$ and $a_{8-i} = a_{i+1}, \; b_{8-i} = b_i, \; i = 1, 2, \ldots$ so, it can be written as follows

$$b_1 \, a_2 \, b_2 \, a_3 \, b_3 \, a_4 \, b_4 \, a_4 \, b_3 \, a_3 \, b_2 \, a_2 \, b_1 \tag{13}$$

which is a 6-stage method. It has 7 coefficients b_i and 6 coefficients a_i. If $\varphi_h^{[b]}$ is the exact solution of the autonomous problem (12) then, we can concatenate

Table 1 Coefficients for different splitting methods

Coefficients of the 6-stage 4th-order method (13)	
$b_1 = 0.0792036964311957$	$a_2 = 0.209515106613362$
$b_2 = 0.353172906049774$	$a_3 = -0.143851773179818$
$b_3 = -0.0420650803577195$	$a_4 = 1/2 - (a_2 + a_3)$
$b_4 = 1 - 2(b_1 + b_2 + b_3)$	
Coefficients of the 10-stage 6th-order method (14)	
$a_1 = 0.0502627644003922$	$b_1 = 0.148816447901042$
$a_2 = 0.413514300428344$	$b_2 = -0.132385865767784$
$a_3 = 0.0450798897943977$	$b_3 = 0.067307604692185$
$a_4 = -0.188054853819569$	$b_4 = 0.432666402578175$
$a_5 = 0.541960678450780$	$b_5 = 1/2 - (b_1 + \ldots + b_4)$
$a_6 = 1 - 2(a_1 + \ldots + a_5)$	
Coefficients of the 2-stage (4,2) method (21)	
$a_1 = \dfrac{3 - \sqrt{3}}{6}$	$b_1 = 1/2$
$a_2 = 1 - 2a_1$	
Coefficients of the 5-stage (8,4) method (22)	
$a_1 = 0.07534696026989288842$	$b_1 = 0.19022593937367661925$
$a_2 = 0.51791685468825678 23$	$b_2 = 0.84652407044352625706$
$a_3 = 1/2 - (a_1 + a_2)$	$b_3 = 1 - 2(b_1 + b_2)$

the last map in one step with the first one in the following one, i.e. $\varphi^{[b]}_{b_1 h} \circ \varphi^{[b]}_{b_m h} = \varphi^{[b]}_{(b_1+b_m)h}$ (it is called the First Same As Last (FSAL) property) and we can save one evaluation. However, if we use a numerical approximation to solve eq. (12) this property is, in general, not satisfied.

A sixth-order method is given by the following 10-stage symmetric sequence

$$a_1\, b_1\, a_2\, b_2\, a_3\, b_3\, a_4\, b_4\, a_5\, b_5\, a_6\, b_5\, a_4\, b_4\, a_4\, b_3\, a_3\, b_2\, a_2\, b_1\, a_1 . \tag{14}$$

The coefficients of methods (13)–(14) (taken from [12]) are collected in Table 1 for convenience of the reader.

The splitting methods we have presented are valid for autonomous equations because the exact solution of the equation (10) can be written formally as the exponential a Lie operator, $\varphi_t = \exp\big(t\, L_f\big)$, with $L_f \equiv f \cdot \nabla$. The map φ_t can be approximated by a composition of exponentials of Lie operators associated to the vector fields $f^{[a]}$ and $f^{[b]}$. If the problem is non-autonomous, one can take the time as a new coordinate and transform the original non-autonomous equation into an autonomous one in an extended phase space. However, in this case it is advantageous to consider the time not as one but as two independent coordinates, and this has to be combined with an appropriate splitting method. For this reason, we consider separately the autonomous from the non-autonomous LQ optimal control problem.

3.2 The autonomous case

If the system is autonomous, we can compute the solution of the RDE (7) at t_0 as follows.

$$\begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = \exp\left((t_0 - T) \begin{bmatrix} A & -S \\ -Q & -A^T \end{bmatrix} \right) \begin{bmatrix} U_T \\ V_T \end{bmatrix} .$$

This map can be accurately computed, for example, by a scaling and squaring method [2,29] i.e. if we take K_i as an accurate approximation to the scaled exponential

$$K_i \simeq \exp\left(\frac{(t_0 - T)}{2^i} \begin{bmatrix} A & -S \\ -Q & -A^T \end{bmatrix} \right) \tag{15}$$

(here K_i can be, for example, a diagonal Padé approximation or a Taylor approximation) then

$$\begin{bmatrix} U_0 \\ V_0 \end{bmatrix} \simeq \underbrace{\left(\cdots (K_i)^2 \cdots \right)^2}_{i-times} \begin{bmatrix} U_T \\ V_T \end{bmatrix} .$$

This method provides an accurate approximation to U_0, V_0 from U_T, V_T in just one step. For stiff problems, the approximation in (15) must be sufficiently accurate. Next, one has to integrate forward in time the system (9) which takes the form:

$$\frac{d}{dt} \begin{bmatrix} v \\ x \end{bmatrix} = \begin{bmatrix} Mv \\ N(\hat{v})x \end{bmatrix} \tag{16}$$

that can be split into solvable parts

$$\frac{d}{dt} \begin{bmatrix} v \\ x \end{bmatrix} = \begin{bmatrix} Mv \\ 0 \end{bmatrix}, \qquad \frac{d}{dt} \begin{bmatrix} v \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ N(\hat{v})x \end{bmatrix} .$$

This is equivalent to the exact solution for the state equation where the continuous symmetric positive definite matrix, $P(t)$, is replaced by a piecewise constant symmetric positive definite matrix. The following algorithm can be used to advance one time step, h, from t_n to $t_{n+1} = t_n + h$, by using a splitting method with coefficients $\{a_i, b_i\}_{i=1}^m$

```
$(v^0, x^0) = (v_n, x_n)$
do  $i = 1, \dots, m$
    $x^i = \exp\left(a_i h N(\hat{v}^{i-1})\right) x^{i-1}$
    $v^i = E_i v^{i-1}$
enddo
$(v_{n+1}, x_{n+1}) = (v^m, x^m)$
$u_{n+1} = - R^{-1} B^T V_{n+1} U_{n+1}^{-1} x_{n+1} .$
```

where $E_i = \exp(b_i h M)$, $i = 1, 2, \dots, m$. To take advantage of the FSAL property to save the computation of a map, one has to slightly adjust the previous algorithm. Obviously, at $t = T$ we must reach the exact solution $P(T) = Q_T$ with up to round off accuracy.

We can choose the flows associated to the coefficients a_i or b_i to advance the RDE. In general, it is convenient to advance the RDE with maps associated to the coefficients having the smaller negative value. For example, we have interchanged the coefficients a_i and b_i in the 6-stage fourth-order method (13) as given in [12] so $\min\{b_i\} > \min\{a_i\}$ because the RDE is advanced with the coefficients b_i in the algorithm.

3.3 The non-autonomous case

If the system is non-autonomous, the solution of the RDE has no analytic solution in a closed form. In this case we have to solve numerically the RDE backward in time using, for example, high order Magnus integrators (e.g. a method of order six or eight [10]) preserving the symplectic structure and positivity is guaranteed for sufficiently small time steps [13]. Other methods like high order extrapolation methods can also be used. Once U_0, V_0 are computed, we need to integrate forward in time the system of non-autonomous equations. If we split the system as in the autonomous case, we end up with two non-autonomous problems, both with no solution in a closed form. We are then looking for an alternative split which provides a more efficient scheme. This can be achieved by frozen the time following a proper sequence. To this purpose, we take the time as two new coordinates as follows

$$\begin{cases} v' = M(t_1)v \\ x' = N(t_2,\hat{v})x \\ t_1' = 1 \\ t_2' = 1 \end{cases}$$

where $' \equiv \frac{d}{dt}$, and we split the system as follows

$$\begin{cases} v' = M(t_1)v \\ x' = 0 \\ t_1' = 0 \\ t_2' = 1 \end{cases} \qquad \begin{cases} v' = 0 \\ x' = N(t_2,\hat{v})x \\ t_1' = 1 \\ t_2' = 0 \end{cases} \tag{17}$$

This corresponds to two linear autonomous equations in the extended phase space, and each part is now exactly solvable. Then, the same splitting methods as in the autonomous case can be used. The algorithm for one time step, h, is given by:

```
$(v^0, x^0, t_1^0, t_2^0) = (v_n, x_n, t_n, t_n)$
do  $i = 1, m$
   $x^i = \exp\big(a_i h N(t_1^{i-1}, \hat{v}^{i-1})\big) x^{i-1}$
   $t_2^i = t_2^{i-1} + a_i h$
   $v^i = \exp\big(b_i h M(t_2^i)\big) v^{i-1}$                    (18)
   $t_1^i = t_1^{i-1} + b_i h$
enddo
$(v_{n+1}, x_{n+1}, t_{n+1}, t_{n+1}) = (v^m, x^m, t_1^m, t_2^m)$
$u_{n+1} = -R^{-1}(t_{n+1})\, B^T(t_{n+1})\, V_{n+1}\, U_{n+1}^{-1}\, x_{n+1}\,.$
```

Since the matrix $M(t)$ changes at each step, the exponentials,

$$\exp\big(b_i h M(t_2^i)\big), \qquad t_2^i = t_n + c_i h$$

with $c_i = \sum_{j=1}^{i} a_j$, need to be computed at each step. Splitting methods of order greater than two necessarily have negative coefficients a_i, b_i (see [7] and references therein) and the preservation of positivity could not be guaranteed. We present, however, the following result:

Theorem 3 *Splitting methods with coefficients a_i, b_i, such that $0 \le c_i = \sum_{j=1}^{i} a_j \le 1$ applied to the system (17) in the sequence given by the algorithm (18) have positive definite solutions for the RDE for sufficiently small time steps.*

Proof Negative coefficients b_i correspond to backward integration of an autonomous RDE so the property is not changed. However, since the numerical solution associated to a splitting method can be considered as the exact solution of a piecewise constant RDE, positivity is only guaranteed for sufficiently small time steps. If the coefficients a_i are such that $0 \le c_i \le 1$ then the variable t_2^i only takes values inside the interval $[t_0, T]$ where the matrix $M(t_2^i)$ is well defined.

If the computation of the exponential is the most time consuming part of the algorithm, it is possible to look for another algorithm where the exponentials can be replaced by a more economical approximation, as for example a second order diagonal Padé approximation

$$\Phi_{b_i h}^{[b]}(t_2^i) = \frac{I + \frac{1}{2} b_i h M(t_2^i)}{I - \frac{1}{2} b_i h M(t_2^i)} = \exp(b_i h M(t_2^i)) + \mathcal{O}(h^3)$$

where Theorem 2 guarantees the solution stays positive definite. In general, M belongs to the Lie algebra of symplectic matrices and the exponential belongs to the associated symplectic Lie group. Diagonal Padé approximations preserve this group property for the symplectic algebra [21].

Now it is important to keep in mind that the coefficients from Table 1 are obtained taking into account the equations are exactly solved (or are numerically solved to high accuracy) and this is not the case if one uses a low order approximation to solve one of the equations. Alternatively, we can look for composition methods.

We can proceed as follows. Let $S_h^{[2]}$ be the following symmetric second order method

$$S_h^{[2]} = \varphi_{h/2}^{[a]} \circ \Phi_h^{[b]} \circ \varphi_{h/2}^{[a]} : \left\{ \begin{array}{l} x^{1/2} = \exp\left(\frac{h}{2} N(t_1^0, \hat{v}^0)\right) x^0 \\ t_2^{1/2} = t_2^0 + \frac{h}{2} \\ v^1 = \Phi_h^{[b]}(t_2^{1/2}) v^{i-1} \\ t_1^1 = t_1^0 + h \\ x^1 = \exp\left(\frac{h}{2} N(t_1^1, \hat{v}^1)\right) x^{1/2} \\ t_2^1 = t_2^{1/2} + \frac{h}{2} \end{array} \right. \qquad (19)$$

Taking $S_h^{[2]}$ as the basic method, we can build methods of order, p, with $p > 2$, as a composition of this basic method

$$S_h^{[p]} = \prod_{i=1}^{m} S_{\alpha_i h}^{[2]}. \tag{20}$$

The simplest fourth-order composition, $S_{\alpha_1 h}^{[2]} \circ S_{\alpha_2 h}^{[2]} \circ S_{\alpha_1 h}^{[2]}$, with $\alpha_1 = 1/(2 - 2^{1/3}) \simeq 1.35, \alpha_2 = 1 - 2\alpha_1 \simeq -1.7$ (usually referred as Yoshida's composition) satisfies $\alpha_1 > 1$ and $\alpha_1 + \alpha_2 < 0$ so, by the same reasons as in Theorem 3, positive definite solutions for the RDE is not guaranteed. However, the following fourth-order method satisfies this property on the coefficients and shows a good performance

$$S_h^{[4]} = S_{\alpha_1 h}^{[2]} \circ S_{\alpha_1 h}^{[2]} \circ S_{\alpha_2 h}^{[2]} \circ S_{\alpha_1 h}^{[2]} \circ S_{\alpha_1 h}^{[2]}$$

with $\alpha_1 = 1/(4 - 4^{1/3}),\ \alpha_2 = -4^{1/3}/(4 - 4^{1/3})$ (where $0 \leq \alpha_1,\ 2\alpha_1,\ 2\alpha_1 + \alpha_2,\ 3\alpha_1 + \alpha_2,\ 4\alpha_1 + \alpha_2 \leq 1$). Several sets of coefficients for methods of different orders and number of stages are collected in [18,28].

Notice that at $t = T$ we must reach a numerical approximation close to the exact solution $P(T) = Q_T$. The methods presented in this work will approximate this value with accuracy up to the order of the method, i.e. $P_{ap}(T) = Q_T + \mathcal{O}(h^p)$, and this can be used as a measure of the accuracy of the algorithm.

3.4 Methods for near-integrable problems

In some cases, LQ optimal control problems can be formulated as a small perturbation of an exactly solvable problem. In this case it can be convenient to split into the dominant part and the perturbation. If we have

$$x' = f^{[a]}(x) + \varepsilon f^{[b]}(x)$$

where $|\varepsilon| \ll 1$ and both equations are exactly solvable (or can be efficiently solved by a numerical method) then splitting methods tailored for this problem usually show a high performance. For example, two highly efficient methods, whose coefficients from [27] are collected in Table 1, are given by the following compositions:

$$a_1\, b_1\, a_2\, b_1\, a_1 \tag{21}$$

referred as a (4,2) method (a second order methods which, in the limit $\varepsilon \to 0$, behaves as a fourth order method) and

$$a_1\, b_1\, a_2\, b_2\, a_3\, b_3\, a_3\, b_2\, a_2\, b_1\, a_1 \tag{22}$$

referred as a (8,4) method. More elaborated methods with more stages are given in [8].

If the problem is non-autonomous

$$x' = f^{[a]}(x,t) + \varepsilon f^{[b]}(x,t)$$

it is still possible to take the time as a new coordinate (only one new coordinate instead of two, contrarily to the previous and more general case). This split allows to preserve the structure of a near integrable problem, in the extended phase space[1] [11]

$$\frac{d}{dt}\begin{bmatrix} x \\ t_1 \end{bmatrix} = \begin{bmatrix} f^{[a]}(x,t_1) \\ 1 \end{bmatrix} + \varepsilon \begin{bmatrix} f^{[b]}(x,t_1) \\ 0 \end{bmatrix}.$$

This requires to integrate exactly (or up to high accuracy) the non-autonomous equation associated to the dominant part

$$x' = f^{[a]}(x,t),$$

and to solve the perturbed part with the time frozen.

For example, LQ optimal control problems when the matrix A is constant and such that $\|A\| \gg \|S\|$ and $\|A\| \gg \|Q\|$ have this structure. We can split the problem as follows

$$x' = f^{[a]}(x,t) \ : \ \begin{cases} \begin{bmatrix} U \\ V \end{bmatrix}' = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}, \\ \qquad x' = \Big(A - S(t)\,V(t)\,U(t)^{-1}\Big)x \end{cases} \tag{23}$$

with solution for the matrix RDE: $U = e^{(t-t^*)A}U^*$, $V = e^{-(t-t^*)A^T}V^*$, where U^*, V^* are the initial conditions at $t = t^*$. The equation for the state vector is then

$$x' = \Big(A - S(t)\, e^{-(t-t^*)A^T} V^*\, U^{*-1} e^{-(t-t^*)A}\Big)x.$$

This is a linear equation with no analytical solution in a closed form, but can be numerically solved to high accuracy by using, for example, a Magnus integrator [9].

Standard Magnus integrators involve commutators of the matrix $M(t)$ evaluated a different time points. Backward error analysis for Magnus integrators applied to the RDE shows that the associated equation (6) only guarantees $Q_h, S_h \geq 0$ for second order methods or, if high order methods are used, for a sufficiently small time step h.

Next, the coordinate t_1 is advanced and for the perturbation we have to solve the autonomous problem ($x' = \varepsilon f^{[b]}(x,t_1)$)

$$\frac{d}{dt}\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} 0 & -S(t_1) \\ -Q(t_1) & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}, \tag{24}$$

where the value of t_1 is frozen. Since $\|Q\|$ and $\|S\|$ are small, it suffices, for most practical purposes, to approximate the exact solution by a low order Taylor method.

[1] If the time is considered as two different coordinates as in the general separable case, this structure of a perturbed integrable problem is lost.

4 Differential games with N players

Let us now consider the problem of differential games with N players given by the equations

$$x'(t) = A(t)x(t) + \sum_{i=1}^{N} B_i(t)u_i(t), \qquad x(0) = x_0\,,$$

and the quadratic cost function

$$\begin{aligned} J_i &= x^T(T)\,Q_{iT}\,x(T) \\ &+ \int_0^T \left\{ x^T(t)Q_i(t)x(t) + u_i^T(t)R_{ii}(t)u_i(t) + \sum_{j=1,j\neq i}^{N} u_j^T(t)R_{ij}(t)u_j(t) \right\} dt \end{aligned}$$

$i = 1, \ldots, N$. From [37], for a zero sum game it is necessary that $R_{ij} \neq 0$, $i \neq j$, but in a non-zero sum game it is natural to choose $R_{ij} = 0$, $i \neq j$, because in the most frequent applications the cost function of each player does not contain the other player control. In this way, the quadratic cost function J_i depends only on the control u_i.

Let us first consider a non-cooperative non-zero sum game where each player, in order to minimize its cost function, determines his action in an independent way knowing only the initial state of the game and the model structure. Under these conditions the optimal controls are given by

$$u_i(t) \;=\; -\,R_{ii}^{-1}(t)\,B_i^T(t)\,P_i(t)\,x(t)\,, \qquad i = 1, \ldots, N,$$

where the matrices P_i, satisfy the coupled matrix RDEs

$$P_i' = -Q_i(t) - A^T(t)P_i - P_iA(t) + \textstyle\sum_{j=1}^{N} P_iS_j(t)P_j \tag{25}$$

with $S_i(t) = B_i(t)\,R_{ii}^{-1}(t)\,B_i^T(t)$, $i = 1, \ldots, N$. If we denote by

$$W(t) = \begin{bmatrix} P_1(t) \\ \vdots \\ P_N(t) \end{bmatrix}, \;\; C(t) = \begin{bmatrix} -Q_1(t) \\ \vdots \\ -Q_N(t) \end{bmatrix}, \;\; D(t) \;=\; \begin{bmatrix} A^T(t) & 0 & \cdots & 0 \\ 0 & A^T(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A^T(t) \end{bmatrix},$$

$$B(t) = [\,-S_1(t) \quad \cdots \quad -S_N(t)\,]\,,$$

then, the coupled system (25) can be written as

$$W'(t) = C(t) - D(t)W(t) - W(t)A(t) - W(t)B(t)W(t)\,.$$

From [23,33], if we consider $y(t) \in \mathbb{R}^{(N+1)n\times n}$ the solution of the linear equation

$$y'(t) \;=\; \begin{bmatrix} A(t) & B(t) \\ C(t) & -D(t) \end{bmatrix} y(t)\,; \qquad y(T) = \begin{bmatrix} U(T) \\ V_1(T) \\ \vdots \\ V_N(T) \end{bmatrix} = \begin{bmatrix} I \\ Q_{1T} \\ \vdots \\ Q_{NT} \end{bmatrix}, \tag{26}$$

an indirect method to compute $P_i(t) = V_i(t)\, U(t)^{-1}$ is obtained.

As in the previous case, we can integrate backward in time the coupled RDEs (26) with a highly accurate method. Once the initial conditions are obtained, we can integrate forward in time the whole system.

This problem has formally the same structure as the LQ optimal control problem (but it has not an associated Hamiltonian), and the numerical integration can be carried out using the same methods as previously. The main difference is on the qualitative properties, the size of the matrices, and then the computational cost.

4.1 Integrators for the zero sum game

From [37], for a zero sum game it is necessary that $R_{ij} \neq 0$, $i \neq j$. In this way, the quadratic cost functions J_i depend on all controls u_i, $i = 1, \ldots, N$. To simplify the presentation, we consider the case of two players. The coupled Riccati differential equations to be solved are

$$\begin{aligned} P_1' &= -Q_1(t) - A^T(t)P_1 - P_1A(t) + P_1S_1(t)P_1 + P_1S_2(t)P_2 + P_2S_{22}(t)P_2\,, \\ P_2' &= -Q_2(t) - A^T(t)P_2 - P_2A(t) + P_2S_2(t)P_2 + P_2S_1(t)P_1 + P_1S_{11}(t)P_1\,, \end{aligned}$$

with the final conditions, $P_1(T) = Q_{1T}\,, \quad P_2(T) = Q_{2T}$, wherein

$$S_{ij}(t) \;=\; B_i(t)\, R_{ij}^{-1}(t)\, B_i^T(t)\,; \quad i = 1,\, 2\,.$$

This problem can not be reformulated as a linear problem. This means we can not use exponential methods as previously. To integrate backward the coupled RDEs we can use a highly accurate method to reduce the error propagation. A high order extrapolation method based on a symmetric second order integrator, say $\Phi_h^{[b]}$, would be a good choice. Then, this symmetric second order method can also be used as the basic map in the composition (19) for the forward time integration, and we can use the composition methods given in (20).

5 Numerical examples

In order to test the performance of the numerical methods presented in this work we consider the problem of air pollutant emissions studied in [25] and we have generalized this problem to the case of N regions (N players) and the constant parameters are replaced by time dependent functions

$$x' = -a(t)x + b(t)\sum_{i=1}^{N} u_i(t); \qquad x(0) = x_0.$$

Here $x(t) \in \mathbb{R}$ is the excess of the pollutant in the atmosphere, $u_i(t) \in \mathbb{R}$, $i = 1, \ldots, N$, denote the emissions of each region and $a(t)$, $b(t)$ are positive functions related with the intervention of the nature on environment. The cost functions to minimize are given by

$$J_i = \int_0^T e^{-\rho t} \left\{ c_i(t) u_i^2(t) + d_i(t) x^2(t) \right\} \, dt \, , \quad i = 1, \ldots, N \, ,$$

where $c_i(t)$, $d_i(t)$, are positive functions related to the costs of emission and pollution withstand respectively, and ρ is a refresh rate.

In our notation, $R_{ij} = 0$, $i \neq j$, $Q_{iT} = 0$, $Q_i(t) = d_i(t)e^{-\rho t}$, $R_{ii} = c_i(t)e^{-\rho t}$, and $S_i = b(t)^2 e^{\rho t}/c_i(t)$, $i = 1, \ldots, N$. Equation (26) becomes

$$y'(t) = K(t) y(t) \, , \qquad y(T) = [1, \, 0, \, \ldots \, , 0]^T \; ;$$

$$y(t) = \begin{bmatrix} u(t) \\ v_1(t) \\ \vdots \\ v_N(t) \end{bmatrix} \; ; \; K(t) = \begin{bmatrix} -a(t) & -\dfrac{b(t)^2}{c_1(t)} e^{\rho t} & \cdots & -\dfrac{b(t)^2}{c_N(t)} e^{\rho t} \\ -d_1(t) e^{-\rho t} & a(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -d_N(t) e^{-\rho t} & 0 & \cdots & a(t) \end{bmatrix} \tag{27}$$

and need be solved from $t = T$ to $t = 0$ to obtain an accurate approximation to $y(0)$. Then, the IVP

$$\begin{aligned} y'(t) &= K(t) y(t) \, , \qquad y(0) = y_0 \, , \\ x' &= \left(-a(t) - b(t)^2 e^{\rho t} \sum_{i=1}^{N} \frac{1}{c_i(t)} \frac{v_i(t)}{u(t)} \right) x \, , \quad x(0) = x_0 \, , \\ u_i(t) &= -\frac{1}{c_i(t)} \, e^{\rho t} \, b(t) \, \frac{v_i(t)}{u(t)} \, x(t) \, , \qquad i = 1, \ldots, N. \end{aligned}$$

need be solved from $t = 0$ to $t = T$.

The new methods, denoted by SPn, where n is the order of the method for the schemes with coefficients given in Table 1, will be tested versus the following standard numerical methods:

- RK4: The well known 4-stage fourth-order Runge-Kutta method.
- ODE45: The variable step and variable order algorithm `ode45` implemented in `Matlab`.
- ODE113: The variable step and variable order algorithm `ode113` implemented in `Matlab`.

We consider the autonomous and non-autonomous cases separately. In all cases we take $T = 1$, initial condition, $x_0 = 10$, and consider the case of $N = 10$ players.

The autonomous problem. This corresponds to the case $\rho = 0$ and the functions a, b, c_i, d_i take constant values, so K in (27) is a constant matrix and the initial condition $y(0)$ is given by

$$y(0) = e^{-TK} y(T).$$

The initial conditions for the matrix RDE are computed up to round off accuracy and the performance of the methods is measured by taking into account the forward time integration. The solution at $t = T$ satisfies that $v_i(T) = 0, \ i = 1, \ldots, N$, and this is exactly satisfied, up to round off error, by the splitting methods so we measure the error as $|x(T) - x_{ap}(T)|$, where x_{ap} is the approximated solution obtained by the numerical integrators and the exact solution $x(T)$ is obtained numerically to a sufficiently high accuracy.

1. We first take the following values for the parameters:

$$a = b = 1, \qquad c_i = \frac{1}{d_i} = \frac{10+i}{2}.$$

   We measure the accuracy versus the number of evaluations of the function $y(t)$, or equivalently of $u(t)$ or $v_i(t)$, for each method when the numerical integration is carried out using different values of the time step (or different values for the absolute and relative tolerances for the methods ODE45 and ODE113). In the numerical experiments we consider `AbsTol`$=10^{-i}$, `RelTol`$=10^{1-i}$, $i = 0, 1, 2, \ldots$ For the splitting methods the cost corresponds to the number of stages. Obviously, the splitting methods require the evaluation of several exponentials of matrices and the computational cost will depend on the problem. However, for the autonomous case most exponentials can be computed and be stored at the beginning, and used repeatedly along the integration.
   Figure 1 shows the obtained results. We observe that the splitting methods show more accurate results at the same number of evaluations. The performance of the new splitting methods could be further improved if the algorithms were implemented with variable order and variable step, as it is the case of ODE45 and ODE113, but this is very much problem dependent.
2. We repeat the same experiment taking

$$a = 2, \qquad b = 1, \qquad c_i = \frac{1}{d_i} = \frac{100+i}{2}.$$

   This problem can be considered as a perturbed problem and we study the performance of the splitting methods (4,2) and (8,4) for the splitting given by (23) and (24), which are tailored for this class of problems.
   We solve separately the dominant part of the system, given by the equations

$$\begin{bmatrix} u \\ v_1 \\ \vdots \\ v_N \end{bmatrix}' = \begin{bmatrix} -2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \end{bmatrix} \begin{bmatrix} u \\ v_1 \\ \vdots \\ v_N \end{bmatrix}, \qquad \begin{bmatrix} u(t_n) \\ v_1(t_n) \\ \vdots \\ v_N(t_n) \end{bmatrix} = \begin{bmatrix} u_n \\ v_{1,n} \\ \vdots \\ v_{N,n} \end{bmatrix}$$

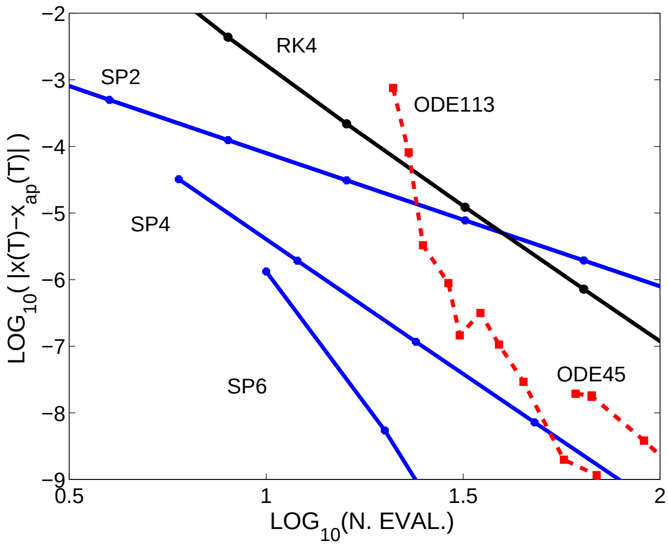

Fig. 1 Error $|x(T) - x_{ap}(T)|$ versus the number of evaluations for the autonomous case with 10 players and $a = b = 1,\ c_i = 1/d_i = (10+i)/2$.

$$x' = \left(-2 - e^{\rho t} \sum_{i=1}^{N} \frac{1}{c_i} \frac{v_i(t)}{u(t)} \right) x \,, \qquad x(0) = x_0 \,, \tag{28}$$

where the RDE has trivial solution, $u(t) = e^{-2(t-t_n)} u_n,\ v_i(t) = e^{2(t-t_n)} v_{i,n},\ i = 1, \ldots, n$, which we plug into the equation of the state vector

$$x' = \left(-2 - e^{\rho t} e^{4(t-t_n)} \sum_{i=1}^{N} \frac{1}{c_i} \frac{v_{i,n}}{u_n} \right) x \,. \tag{29}$$

The equation for the state vector has exact solution since it is a scalar equation. Figure 2 shows the results obtained. The schemes SP4 and SP6 lead to slightly worst results and are not showed.

The non autonomous problem. We consider the one player case, $N = 1$, with the following choice for the functions and parameters

$$a(t) = 2 + \tanh\left(5(t - \frac{1}{2}) \right), \quad \rho = \frac{1}{10}, \quad b = 1, \quad c_1 = \frac{1}{d_1} = \frac{11}{2}.$$

where $a(t) \in [1, 3]$, and we repeat the same experiments replacing the values of c_1, d_1 by $c_1 = 1/d_1 = 101/2$, which makes the system closer to a near integrable systems, in order to study the performance of the splitting methods in this case. We consider the splitting (17) and the algorithm (18). The case of one player corresponds to a LQ optimal control problem and, as already mentioned, the solution of the RDE has to be a positive function. To show

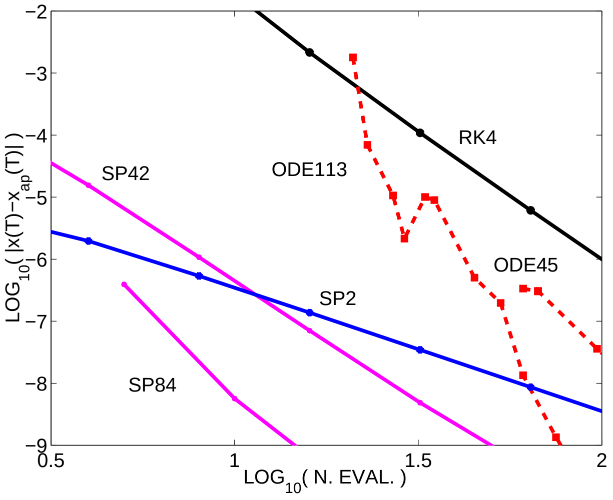

Fig. 2 Error $|x(T) - x_{ap}(T)|$ versus the number of evaluations for the autonomous case with 10 players and $a = 2, b = 1, \ c_i = 1/d_i = (100 + i)/2$.

the superiority of the splitting methods when this property is important to be preserved, we measure the value of $P(T) = \frac{v(T)}{u(T)}$ for each method and choice of the time step. If this value is negative and smaller than a tolerance value, which we take -10^{-8} (i.e. if $P_{ap} < -10^{-8}$) we marked this result with a circle.

Figure 3 illustrates the results obtained. The second order splitting methods preserves positivity for all time steps and the fourth- and sixth-order splitting methods fail to preserve this property only for the largest time step, contrarily to the standard RK method or the methods implemented in `Matlab`. We observe that when the off diagonal coefficients of the RDE are small, the superiority of the splitting methods is even higher. We have repeated the numerical experiments with a higher number of players, and similar results are obtained.

6 Conclusions

We have considered the numerical integration of linear-quadratic optimal control problems and N-player differential games. These problems require the solution of non-autonomous matrix Riccati differential equations coupled with linear differential equations for the dynamic state. This is a boundary value problem where the RDE has positive definite solution matrices.

Direct algorithms for the numerical integration of the RDE of order greater than one (one step schemes or strictly stable multistep schemes, explicit or implicit) cannot guarantee that the solution stays positive definite. However,

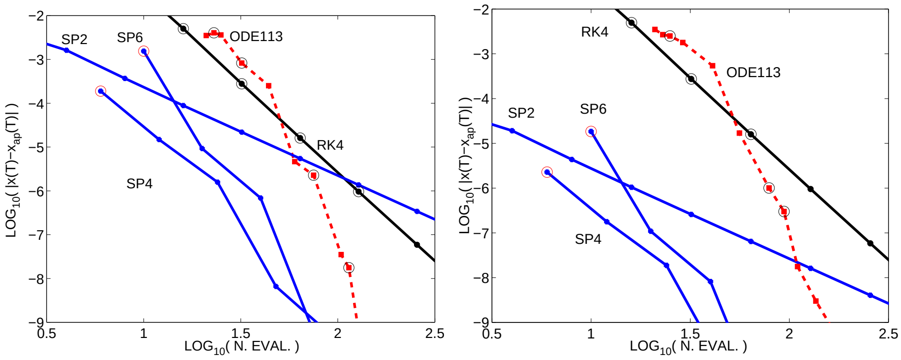

Fig. 3 Error $|x(T)-x_{ap}(T)|$ versus the number of evaluations for the non-autonomous case with 1 player and $a(t)=2+\tanh(5(t-1/2))$, $b=1$, $\rho=1/10$ and the following values for c_1, d_1: Left panel $c_1=1/d_1=11/2$; right panel $c_1=1/d_1=101/2$. Solutions where $P_{ap}(T)<-10^{-8}$ are marked with a circle.

high order symplectic methods used as indirect methods to solve the associated Hamiltonian system can guarantee the solution stays positive definite.

We present high order splitting and Magnus integrators that for this problem can be considered as exponential explicit methods (once the RDE has previously integrated backward in time). In particular, we analyze the positivity of the solution for the associated matrix RDE.

The methods proposed are high-order explicit geometric integrators which consider the time as two new coordinates. This allows us to integrate the whole system forward in time while the solution of the RDE stays positive definite. If the equations correspond to a near-integrable system, tailored splitting methods for perturbed systems provide a further improvement. The numerical examples considered showed the performance of the proposed methods as well as the good preservation of some of the qualitative properties. Similar ideas could be used for solving non-linear optimal control problems (e.g. some linearization methods numerically solve iterative sequences of linear systems [5]) being this problem under consideration at this moment.

Acknowledgement

The author wishes to thank the University of California San Diego for its hospitality where part of this work was done. He also acknowledges the support of the Ministerio de Ciencia e Innovación (Spain) under the coordinated project MTM2010-18246-C03. The author also acknowledges the suggestions by the referees to improve the presentation of this work.

References

1. H. Abou-Kandil, G. Freiling, V. Ionescy and G. Jank, *Matrix Riccati equations in control and systems theory.* Burkhäuser Verlag, Basel (2003)
2. A.H. Al-Mohy and N.J. Higham, Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators. SIAM J. Sci. Comp. **33**, 488–511 (2011)
3. B.D.O. Anderson and J.B. Moore, *Optimal control: linear quadratic methods.* Dover, New York (1990)
4. U.M. Ascher, R.M. Mattheij, and R.D. Russell, Numerical solutions of boundary value problems for ordinary differential equations. Prentice-Hall. Englewood Cliffs, NJ (1988)
5. P. Bader, S. Blanes, and E. Ponsoda, Structure preserving integrators for solving linear quadratic optimal control problems with applications to describe the flight of a quadrotor. J. Comput. Appl. Math. **262**, 223–233 (2014)
6. T. Basar and G.J. Olsder, Dynamic non cooperative game theory, 2nd. Ed.. SIAM, Philadelphhia (1999)
7. S. Blanes and F. Casas. On the necessity of negative coefficients for operator splitting schemes of order higher than two. *Appl. Num. Math.* **54**, 23–37 (2005)
8. S. Blanes, F. Casas, A. Farrés, J. Laskar, J. Makazaga, and A. Murua, New families of symplectic splitting methods for numerical integration in dynamical astronomy. Appl. Numer. Math. **68**, 58–72 (2013)
9. S. Blanes, F. Casas, J.A. Oteo, and J. Ros, The Magnus expansion and some of its applications. Physics Reports **470**, 151–238 (2009)
10. S. Blanes, F. Casas and J. Ros, High order optimized geometric integrators for linear differential equations. BIT **42**, 262–284 (2002)
11. S. Blanes, F. Diele, C. Marangi, and S. Ragni, Splitting and composition methods for explicit time dependence in separable dynamical systems. J. Comput. Appl. Math. **235**, 646–659 (2010)
12. S. Blanes and P.C. Moan, Practical symplectic partitioned Runge-Kutta and Runge-Kutta-Nystrm methods, J. Comput. Appl. Math. **142**, 313–330 (2002)
13. S. Blanes and E. Ponsoda, Magnus integrators for solving linear-quadratic differential games. J. Comput. Appl. Math. **236**, 3394–3408 (2012)
14. C. Brif, R. Chakrabarti, and H. Rabitz, Control of quantum phenomena: past, present and future. New J. Phys. **12**, 075008 (68pp) (2010)
15. J.B. Cruz and C.I. Chen, Series Nash solution of two person non zero sum linear quadratic games. J. Optim. Theory Appl. **7**, 240–257 (1971)
16. L. Dieci & T. Eirola, Positive definitness in the numerical solution of Riccati differential quations. Numer. Math. **67**, 303–313 (1994)
17. J. Engwerda, LQ dynamic optimization and differential games. John Wiley and soons (2005)
18. E. Hairer, C. Lubich and G. Wanner, Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations (2nd edition). Springer Series in Computational Mathematics, 31, Springer-Verlag (2006)
19. M. Hochbruck and A. Ostermann, Exponential integrators. Acta Numerica, **19**, 209–286. 2010.
20. R.A. Horn and C.R. Johnson, Matrix Analysis, Cambridge University Press, New York (1985)
21. A. Iserles, H. Z. Munthe-Kaas, S.P. Nørsett and A. Zanna, Lie group methods. Acta Numerica **9**, 215–365 (2000)
22. A. Iserles and S.P. Nørsett, On the solution of linear differential equations in Lie groups. Phil. Trans. R. Soc. Lond. A **357**, 983–1019 (1999)
23. L. Jódar and E. Ponsoda, Non-autonomous Riccati-type matrix differential equations: existence interval, construction of continuous numerical solutions and error bounds. IMA J. Num. Anal. **15**, 61–74 (1995)
24. L. Jódar, E. Ponsoda and R. Company, Solutions of coupled Riccati equations arising in differential games. Control and Cybernetics **24**, 117–128 (1995)
25. V. Kaitala and M. Pohjola, Sustainable international agreement on greenhouse warming. A game theory study. Control and Game Theoretic Models of the Environment. Carraro and Filar Ed., Birkhauser, Boston, pp. 67–87 (1995)

26. H.B. Keller, *Numerical solution of two point boundary value problems.* CBMS-NSF Regional Conference Series in Applied Mathematics, **24**, SIAM, Philadelphia (1976)
27. R.I. McLachlan, Composition methods in the presence of small parameters, BIT **35**, 258–268 (1995)
28. R. I. McLachlan and, R. Quispel, Splitting Methods, Acta Numer., **11**, 341–434 (2002)
29. C. B. Moler and C. F. Van Loan, Nineteen Dubious Ways to Compute the Exponential of a Matrix, twenty-five years later. SIAM Review **45**, 3–49 (2003)
30. Na, T.Y. *Computational methods in engineering boundary value problems.* Mathematics in science and engineering, **145**, Accademic Press, New York (1979)
31. J.P. Palao and R. Kosloff, Quantum computing by an optimal control algorithm for unitry transformations. Phys. Rev. Lett. **28**, 188304 (2002)
32. A.P. Peirce, M.A. Dahleh, and H. Rabitz, Optimal control of quantum-mechanical systems: existence, numerical approximation, and applications. Phys. Rev. A **37**, 4950–4967 (1988)
33. W.T. Reid, *Riccati Differential Equations.* New York. Academic (1972)
34. J.M. Sanz-Serna and M.P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, London (1994)
35. R.B. Sidje, Expokit: a software package for computing matrix exponentials. ACM Trans. Math. Software **24**, 130–156 (1998)
36. J.L. Speyer and D.H. Jacobson, Primer on optimal control theory. SIAM, Philadelphia (2010)
37. A.W. Starr and Y.C. Ho, Non-zero sum differential games. J. Optim. Theory and Appl. **3**, 179–197 (1969)
38. W. Zhu and H. Rabitz, A rapid monotonically convergent iteration algorithm for quantum optimal control ever the expectation value of a positive definite operator, J. Chem. Phys. **109**, 385–391 (1998)