A Precise Integration Algorithm for Matrix Riccati Differential Equations

Wan-Xie Zhong[1] and Jianping Zhu[2]

[1] State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116023, China
[2] Department of Mathematics & Statistics, Mississippi State University
Mississippi State, MS 39762, USA
jzhu@math.msstate.edu

Abstract. An efficient precise integration method for solving the matrix Riccati differential equation is described in this paper. The method is based on repeated combination of extremely small time intervals, which leads to solutions with an accuracy within the machine precision.

1 Introduction

The general matrix Riccati differential equation can be written as

$$\dot{\mathbf{S}} = -\mathbf{B} + \mathbf{SA} - \mathbf{CS} + \mathbf{SDS} \tag{1}$$

where $\mathbf{S}(t)$ is an $m \times n$ matrix to be solved, $\dot{\mathbf{S}}$ is the derivative of $\mathbf{S}(t)$ with respect to *t*, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are all given matrices with dimensions $n \times n$, $m \times n$, $m \times m$, $n \times m$, respectively. The solution of the matrix Riccati differential equation is very important in various applications, such as in optimal control theory, wave propagation, structural mechanics, and game theory [1-4]. The integration domain is $0 \le t \le t_{\mathrm{f}}$, where t_{f} is given, and the boundary condition is given by

$$\mathbf{S}(t_{\mathrm{f}}) = \mathbf{S}_{\mathrm{f}}, \qquad \text{for } t = t_{\mathrm{f}}, \tag{2}$$

where $\mathbf{S}_{\mathrm{f}}$ is given. Note that the integration of (2) goes backward from t_{f} to 0.

The respective dual Riccati differential equation can be written as

$$\dot{\mathbf{T}} = -\mathbf{D} - \mathbf{TC} + \mathbf{AT} + \mathbf{TBT}, \tag{3}$$

where $\mathbf{T}(t)$ is an $n \times m$ matrix to be solved with the initial condition

$$\mathbf{T}(0) = \mathbf{G}_0 . \tag{4}$$

Since both equations (1) and (3) are nonlinear, it is very difficult, if not impossible, to find analytical solutions for application problems. The most commonly used solution methods are numerical integration schemes based on finite difference [1,5]. The application of these schemes can be difficult when very high accuracy is desirable, or

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2073, pp. 947-956, 2001.
© Springer-Verlag Berlin Heidelberg 2001

when the solution changes dramatically (caused by large matrix $\mathbf{S}_\mathrm{f}$ at the boundary, for example), or else when the problems being solved are stiff. In this paper, an efficient and accurate scheme for solving Riccati differential equations will be presented. The new scheme is based on the precise time integration method for systems of linear differential equations [6-8]. It can provide accurate numerical solutions to equation (1) with errors in the order of computer round-off errors.

2 Linear Equations and Boundary Conditions

The n -dimensional linear transport process can be described [1,2] by

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{q} + \mathbf{D}\mathbf{p}, \qquad \dot{\mathbf{p}} = \mathbf{B}\mathbf{q} + \mathbf{C}\mathbf{p} \tag{5}$$

where $\mathbf{q}, \mathbf{p}$ are vectors of dimension n and m, respectively. When $n = m$, $\mathbf{C} = -\mathbf{A}^\mathrm{T}$, and $\mathbf{B}, \mathbf{D}$ are non-negative symmetric matrices, equation (5) becomes the dual equation of continuous time optimal control problem. In general, most problems require $m + n$ boundary conditions corresponding to (5) in the form of

$$\mathbf{q}(0) = \mathbf{q}_0, \quad \text{when} \quad t = 0; \quad \mathbf{p}(t_\mathrm{f}) = \mathbf{p}_\mathrm{f} \quad \text{when } t = t_\mathrm{f}, \tag{6}$$

where $\mathbf{q}_0, \mathbf{p}_\mathrm{f}$ are given vectors of dimensions n and m, respectively. The precise time integration method in [6] was for initial value problems, and those in [7,8] were for conservative systems with $m = n$. The present paper will discuss the precise time integration method for two point boundary value problems in the form of (5).

For numerical solution of most boundary value problems, the finite difference method is the most commonly used algorithm, which could be difficult to use for some cases due to the loss of accuracy or important properties of the original equation, for example, the conservation property.

To derive the precise time integration method for the Riccati equation, we first need to establish the equations that connect the state vectors $\mathbf{q}_\mathrm{a}, \mathbf{p}_\mathrm{a}$ at $t = t_\mathrm{a}$, with $\mathbf{q}_\mathrm{b}, \mathbf{p}_\mathrm{b}$ at $t = t_\mathrm{b}$. If the interval $(t_\mathrm{a}, t_\mathrm{b})$ is considered as an interval of the entire integration domain $[0, t_\mathrm{f}]$, the equations can be expressed as

$$\mathbf{q}_\mathrm{b} = \mathbf{F}\mathbf{q}_\mathrm{a} - \mathbf{G}\mathbf{p}_\mathrm{b} \tag{7a}$$

$$\mathbf{p}_\mathrm{a} = \mathbf{Q}\mathbf{q}_\mathrm{a} + \mathbf{E}\mathbf{p}_\mathrm{b} \tag{7b}$$

where $\mathbf{F}, \mathbf{G}, \mathbf{Q}, \mathbf{E}$ are $n \times n$, $n \times m$, $m \times n$, $m \times m$ matrices, respectively, to be determined. For time independent system, the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are independent of t . Hence, the $\mathbf{F}, \mathbf{G}, \mathbf{Q}$ and $\mathbf{E}$ will only depend on the length of the interval

$$\Delta t = t_\mathrm{b} - t_\mathrm{a} \tag{8}$$

Treating $\mathbf{q}_\mathrm{a}, \mathbf{p}_\mathrm{a}$ as the given initial vectors at t_a, and taking the partial derivative of equation (7a) with respect to t_b, we have

$$\frac{\partial \mathbf{q}_\mathrm{b}}{\partial t_\mathrm{b}} = \frac{\partial \mathbf{F}}{\partial t_\mathrm{b}} - \frac{\partial \mathbf{G}}{\partial t_\mathrm{b}} - \mathbf{G}\frac{\partial \mathbf{p}_\mathrm{b}}{\partial t_\mathrm{b}}, \qquad \mathbf{0} = \frac{\partial \mathbf{Q}}{\partial t_\mathrm{b}} + \frac{\partial \mathbf{E}}{\partial t_\mathrm{b}} + \mathbf{E}\frac{\partial \mathbf{p}_\mathrm{b}}{\partial t_\mathrm{b}}, \tag{9a}$$

Since equation (5) can be written as

$$\frac{\partial \mathbf{q}_{\mathrm{b}}}{\partial t_{\mathrm{b}}} = \mathbf{A}\mathbf{q}_{\mathrm{a}} + \mathbf{D}\mathbf{p}_{\mathrm{b}}, \qquad \frac{\partial \mathbf{p}_{\mathrm{b}}}{\partial t_{\mathrm{b}}} = \mathbf{B}\mathbf{q}_{\mathrm{a}} + \mathbf{C}\mathbf{p}_{\mathrm{b}}. \tag{10a}$$

Equations (10) can be substituted into (9) to get

$$\frac{\partial \mathbf{F}}{\partial t_{\mathrm{b}}} - (\mathbf{GB} + \mathbf{A})\mathbf{q}_{\mathrm{b}} - (\mathbf{D} + \mathbf{GC} + \frac{\partial \mathbf{G}}{\partial t_{\mathrm{b}}})\mathbf{p}_{\mathrm{b}} = \mathbf{0}, \tag{11a}$$

$$\frac{\partial \mathbf{Q}}{\partial t_{\mathrm{b}}}\mathbf{q}_{\mathrm{a}} + \mathbf{EB}\mathbf{q}_{\mathrm{b}} + (\mathbf{EC} + \frac{\partial \mathbf{E}}{\partial t_{\mathrm{b}}})\mathbf{p}_{\mathrm{b}} = \mathbf{0} \quad . \tag{11b}$$

Note further that the vectors $\mathbf{q}_{\mathrm{a}}, \mathbf{q}_{\mathrm{b}}$ and $\mathbf{p}_{\mathrm{b}}$ in equation (11) are not linearly independent. Substituting equation (7a) into (11), we obtain

$$\left[\frac{\partial \mathbf{F}}{\partial t_{\mathrm{b}}} - (\mathbf{GB} + \mathbf{A})\mathbf{F}\right]\mathbf{q}_{\mathrm{a}} + \left[\mathbf{AG} + \mathbf{GBG} - \mathbf{D} - \mathbf{GC} - \frac{\partial \mathbf{G}}{\partial t_{\mathrm{b}}}\right]\mathbf{p}_{\mathrm{b}} = \mathbf{0}, \tag{12a}$$

$$\left[\frac{\partial \mathbf{Q}}{\partial t_{\mathrm{b}}} + \mathbf{EBF}\right]\mathbf{q}_{\mathrm{a}} + \left[\frac{\partial \mathbf{E}}{\partial t_{\mathrm{b}}} + \mathbf{E}(\mathbf{C} - \mathbf{BG})\right]\mathbf{p}_{\mathrm{b}} = \mathbf{0}. \tag{12b}$$

Since $\mathbf{q}_{\mathrm{a}}$ and $\mathbf{p}_{\mathrm{b}}$ are linearly independent, equation (12) leads to the following

$$\frac{\partial \mathbf{G}}{\partial t_{\mathrm{b}}} = \mathbf{AG} + \mathbf{GBG} - \mathbf{D} - \mathbf{GC}, \quad \frac{\partial \mathbf{F}}{\partial t_{\mathrm{b}}} = (\mathbf{GB} + \mathbf{A})\mathbf{F}, \tag{13a}$$

$$\frac{\partial \mathbf{E}}{\partial t_{\mathrm{b}}} = \mathbf{E}(\mathbf{BG} - \mathbf{C}), \quad \frac{\partial \mathbf{Q}}{\partial t_{\mathrm{b}}} = -\mathbf{EBF}. \tag{13b}$$

The initial conditions at $t_{\mathrm{b}} = t_{\mathrm{a}}$ are

$$\mathbf{G} = \mathbf{0}, \quad \mathbf{Q} = \mathbf{0}, \quad \mathbf{E} = \mathbf{I}_m, \quad \mathbf{F} = \mathbf{I}_n, \tag{14}$$

where $\mathbf{I}_m$ and $\mathbf{I}_n$ are identity matrices with dimensions m and n, respectively.

Similarly, we can treat t_{a} as a variable while fixing t_{b}, which leads to

$$\frac{\partial \mathbf{G}}{\partial t_{\mathrm{a}}} = \mathbf{FDE}, \qquad \frac{\partial \mathbf{F}}{\partial t_{\mathrm{a}}} = -\mathbf{F}(\mathbf{A} + \mathbf{DQ}), \tag{15a}$$

$$\frac{\partial \mathbf{E}}{\partial t_{\mathrm{a}}} = (\mathbf{C} - \mathbf{QD})\mathbf{E}, \quad \frac{\partial \mathbf{Q}}{\partial t_{\mathrm{a}}} = \mathbf{B} - \mathbf{QA} + \mathbf{CQ} - \mathbf{QDQ}. \tag{15b}$$

The initial conditions are similar to (14) at $t_{\mathrm{a}} = t_{\mathrm{b}}$. For time independent system with matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ independent of time, the matrices $\mathbf{F}, \mathbf{G}, \mathbf{Q}$ and $\mathbf{E}$ depend only on the length of the interval $\Delta t = t_{\mathrm{b}} - t_{\mathrm{a}}$. Therefore the relations

$$\mathbf{Q}(t_{\mathrm{a}}, t_{\mathrm{b}}) = \mathbf{Q}(\Delta t), \quad \frac{\partial \mathbf{Q}}{\partial t_{\mathrm{b}}} = \frac{\partial \mathbf{Q}}{\partial (\Delta t)}, \quad \frac{\partial \mathbf{Q}}{\partial t_{\mathrm{a}}} = -\frac{\partial \mathbf{Q}}{\partial (\Delta t)}, \tag{16}$$

hold for matrix $\mathbf{Q}$, and similarly also hold for matrices $\mathbf{F},\mathbf{G},\mathbf{E}$. With these relations, equations (13) can be written as

$$\dot{\mathbf{G}} = \mathbf{AG} + \mathbf{GBG} - \mathbf{D} - \mathbf{GC}, \tag{17a}$$

$$\dot{\mathbf{F}} = (\mathbf{GB} + \mathbf{A})\mathbf{F}, \tag{17b}$$

$$\dot{\mathbf{E}} = \mathbf{E}(\mathbf{BG} - \mathbf{C}), \tag{17c}$$

$$\dot{\mathbf{Q}} = -\mathbf{EBF}. \tag{17d}$$

The dot above $\mathbf{F},\mathbf{G},\mathbf{Q}$ and $\mathbf{E}$ now represents derivatives with respect to Δt. Similarly, equation (15) can be written as

$$\dot{\mathbf{G}} = -\mathbf{FDE}, \tag{18a}$$

$$\dot{\mathbf{F}} = \mathbf{F}(\mathbf{A} + \mathbf{DQ}), \tag{18b}$$

$$\dot{\mathbf{E}} = -(\mathbf{C} - \mathbf{QD})\mathbf{E}, \tag{18c}$$

$$\dot{\mathbf{Q}} = -\mathbf{B} + \mathbf{QA} - \mathbf{CQ} + \mathbf{QDQ}. \tag{18d}$$

Although equations (17) appear to be quite different from (18), it can be proved that they are consistent with each other. Note that equation (18d) is the same as equation in (1). If an algorithm can be developed to calculate the matrix $\mathbf{Q}$ in (18d), such that $\mathbf{Q}$ also satisfies the boundary condition (2), then $\mathbf{Q}$ is the solution matrix $\mathbf{S}$ of (1).

3 Interval Combination

Given two contiguous intervals (t_a, t_b) and (t_b, t_c), we can eliminate the interior state vectors $\mathbf{q}_b, \mathbf{p}_b$ at t_b to form a larger combined interval (t_a, t_c), and obtain equations similar to those in (7) that connect state vectors defined at the two ends t_a and t_c, respectively. Mathematically, the equations for the interval (t_a, t_b) are

$$\mathbf{q}_b = \mathbf{F}_1\mathbf{q}_a - \mathbf{G}_1\mathbf{p}_b, \tag{19a}$$

$$\mathbf{p}_a = \mathbf{Q}_1\mathbf{q}_a + \mathbf{E}_1\mathbf{p}_b, \tag{19b}$$

and those for the interval (t_b, t_c) are

$$\mathbf{q}_c = \mathbf{F}_2\mathbf{q}_b - \mathbf{G}_2\mathbf{p}_c, \tag{20a}$$

$$\mathbf{p}_b = \mathbf{Q}_2\mathbf{q}_b + \mathbf{E}_2\mathbf{p}_c. \tag{20b}$$

To eliminate the interior vectors $\mathbf{q}_b, \mathbf{p}_b$, we solve from (19a) and (20b)

$$\mathbf{q}_b = (\mathbf{I}_n + \mathbf{G}_1\mathbf{Q}_2)^{-1}\mathbf{F}_1\mathbf{q}_a - (\mathbf{I}_n + \mathbf{G}_1\mathbf{Q}_2)^{-1}\mathbf{G}_1\mathbf{E}_2\mathbf{p}_c, \tag{21a}$$

$$\mathbf{p}_b = (\mathbf{I}_m + \mathbf{Q}_2\mathbf{G}_1)^{-1}\mathbf{Q}_2\mathbf{F}_1\mathbf{q}_a + (\mathbf{I}_m + \mathbf{Q}_2\mathbf{G}_1)^{-1}\mathbf{E}_2\mathbf{p}_c, \tag{21b}$$

and substituting (21) into (20a) and (19b), respectively. This leads to, after eliminating $\mathbf{q}_b, \mathbf{p}_b$ and combining the intervals (t_a, t_b) and (t_b, t_c), the equations

$$\mathbf{q}_c = \mathbf{F}_c\mathbf{q}_a - \mathbf{G}_c\mathbf{p}_c, \qquad \mathbf{p}_a = \mathbf{Q}_c\mathbf{q}_a + \mathbf{E}_c\mathbf{p}_c, \tag{22}$$

where

$$\mathbf{G}_{\mathrm{c}} = \mathbf{G}_2 + \mathbf{F}_2(\mathbf{I}_n + \mathbf{G}_1\mathbf{Q}_2)^{-1}\mathbf{G}_1\mathbf{E}_2\,, \tag{23a}$$

$$\mathbf{Q}_{\mathrm{c}} = \mathbf{Q}_1 + \mathbf{E}_1(\mathbf{I}_m + \mathbf{Q}_2\mathbf{G}_1)^{-1}\mathbf{Q}_2\mathbf{F}_1\,, \tag{23b}$$

$$\mathbf{F}_{\mathrm{c}} = \mathbf{F}_2(\mathbf{I}_n + \mathbf{G}_1\mathbf{Q}_2)^{-1}\mathbf{F}_1, \qquad \mathbf{E}_{\mathrm{c}} = \mathbf{E}_1(\mathbf{I}_m + \mathbf{Q}_2\mathbf{G}_1)^{-1}\mathbf{E}_2. \tag{23c}$$

4 The 2^N Type Algorithm

In structural mechanics, the substructuring technique has been widely used to improve computational efficiency. If there are multiple identical substructures, only one of them needs to be analyzed and the result can be used for all other identical substructures. This technique has been used successfully for the computation of some optimal control problems [9]. In the present paper, we extend this technique to the solution of matrix Riccati differential equations. Note that the equations in (7) describe state vectors at a small interval from t_{a} to t_{b}, which corresponds to a single substructure, while those in (22) connect state vectors defined at t_{a} and t_{c}, which correspond to the combination of two contiguous substructures after elimination of the state vectors at t_{b}. The 2^N type algorithm described in [10] is very efficient for this kind of combination involving a large number of similar substructures.

Let η be a typical time step length of an interval $[t_{\mathrm{a}}, t_{\mathrm{b}}]$ for the integration of the equations. We can further divide it uniformly into 2^N subintervals of length τ. For example, with $N = 20$, the length of a subinterval is

$$\tau = \eta / 2^N = \eta / 1048576\,. \tag{24}$$

For time independent systems, all equations corresponding to different subintervals are the same. After $N = 20$ combination steps, all 1048576 subintervals would have been combined to generate a equation system like (7). Note that the entire domain of integration runs from 0 to t_{f}, in which the integration can also be done using the 2^N type algorithm to combining all intervals of length η.

The main part of the computation of this 2^N type algorithm is the repeated execution of

$$\mathbf{G}_{\mathrm{c}} = \mathbf{G} + \mathbf{F}(\mathbf{I}_n + \mathbf{G}\mathbf{Q})^{-1}\mathbf{G}\mathbf{E}, \quad \mathbf{Q}_{\mathrm{c}} = \mathbf{Q} + \mathbf{E}(\mathbf{I}_m + \mathbf{Q}\mathbf{G})^{-1}\mathbf{Q}\mathbf{F}, \tag{25a}$$

$$\mathbf{F}_{\mathrm{c}} = \mathbf{F}(\mathbf{I}_n + \mathbf{G}\mathbf{Q})^{-1}\mathbf{F}, \qquad \mathbf{E}_{\mathrm{c}} = \mathbf{E}(\mathbf{I}_m + \mathbf{Q}\mathbf{G})^{-1}\mathbf{E}, \tag{25b}$$

for N times. Each time the calculated matrices $\mathbf{G}_{\mathrm{c}}, \mathbf{Q}_{\mathrm{c}}, \mathbf{F}_{\mathrm{c}}$ and $\mathbf{E}_{\mathrm{c}}$ are put into the right-hand side of (25) to calculate new matrices for the larger combined intervals.

To start the recursive computation given by equations in (25), it is necessary to generate $\mathbf{G}, \mathbf{Q}, \mathbf{F}$ and $\mathbf{E}$ corresponding to the smallest subinterval of length τ defined by (24). These matrices are defined by equations in (17) (or its equivalent equations in (18)), with the initial conditions in (14). Although equation

(17) is non-linear, the power series expansion method can be used to solve them approximately.

Let $\Delta t = \tau$ in equations (17) and (18), and expand $\mathbf{G}, \mathbf{Q}, \mathbf{F}$ and $\mathbf{E}$ as

$$\mathbf{G}(\tau) = \mathbf{g}_1\tau + \mathbf{g}_2\tau^2 + \mathbf{g}_3\tau^3 + \mathbf{g}_4\tau^4, \quad \mathbf{Q}(\tau) = \mathbf{q}_1\tau + \mathbf{q}_2\tau^2 + \mathbf{q}_3\tau^3 + \mathbf{q}_4\tau^4 \tag{26a}$$

$$\mathbf{F}(\tau) = \mathbf{I} + \mathbf{f}_1\tau + \mathbf{f}_2\tau^2 + \mathbf{f}_3\tau^3 + \mathbf{f}_4\tau^4, \quad \mathbf{E}(\tau) = \mathbf{I} + \mathbf{e}_1\tau + \mathbf{e}_2\tau^2 + \mathbf{e}_3\tau^3 + \mathbf{e}_4\tau^4 \tag{26b}$$

Substituting the first equation in (26a) into (17a) and comparing the coefficients of different powers of τ, we have

$$\begin{aligned}&\mathbf{g}_1 = \mathbf{D},\ \mathbf{g}_2 = (\mathbf{A}\mathbf{g}_1 - \mathbf{g}_1\mathbf{C})/2,\ \mathbf{g}_3 = (\mathbf{A}\mathbf{g}_2 - \mathbf{g}_2\mathbf{C} + \mathbf{g}_1\mathbf{B}\mathbf{g}_1)/3,\\ &\mathbf{g}_4 = (\mathbf{A}\mathbf{g}_3 - \mathbf{g}_3\mathbf{C} + \mathbf{g}_2\mathbf{B}\mathbf{g}_1 + \mathbf{g}_1\mathbf{B}\mathbf{g}_2)/4.\end{aligned} \tag{27}$$

Applying similar procedures to (17b), (17c) and (17d), we obtain

$$\begin{aligned}&\mathbf{f}_1 = \mathbf{A},\quad \mathbf{f}_2 = (\mathbf{A}\mathbf{f}_1 - \mathbf{g}_1\mathbf{B})/2,\quad \mathbf{f}_3 = (\mathbf{A}\mathbf{f}_2 - \mathbf{g}_2\mathbf{B} + \mathbf{g}_1\mathbf{B}\mathbf{f}_1)/3,\\ &\mathbf{f}_4 = (\mathbf{A}\mathbf{f}_3 + \mathbf{g}_3\mathbf{B} + \mathbf{g}_2\mathbf{B}\mathbf{f}_1 + \mathbf{g}_1\mathbf{B}\mathbf{f}_2)/4,\end{aligned} \tag{28}$$

$$\begin{aligned}&\mathbf{e}_1 = -\mathbf{C},\ \mathbf{e}_2 = (\mathbf{B}\mathbf{g}_1 - \mathbf{e}_1\mathbf{C})/2,\ \mathbf{e}_3 = (\mathbf{B}\mathbf{g}_2 - \mathbf{e}_2\mathbf{C} + \mathbf{e}_1\mathbf{B}\mathbf{g}_1)/3,\\ &\mathbf{e}_4 = (\mathbf{B}\mathbf{g}_3 - \mathbf{e}_3\mathbf{C} + \mathbf{e}_2\mathbf{B}\mathbf{g}_1 + \mathbf{e}_1\mathbf{B}\mathbf{g}_2)/4,\end{aligned} \tag{29}$$

$$\begin{aligned}&\mathbf{q}_1 = -\mathbf{B},\ \mathbf{q}_2 = -(\mathbf{B}\mathbf{f}_1 + \mathbf{e}_1\mathbf{B})/2,\ \mathbf{q}_3 = -(\mathbf{B}\mathbf{f}_2 + \mathbf{e}_2\mathbf{B} + \mathbf{e}_1\mathbf{B}\mathbf{f}_1)/3,\\ &\mathbf{q}_4 = -(\mathbf{B}\mathbf{f}_3 + \mathbf{e}_3\mathbf{B} + \mathbf{e}_2\mathbf{B}\mathbf{f}_1 + \mathbf{e}_1\mathbf{B}\mathbf{f}_2)/4.\end{aligned} \tag{30}$$

Higher order approximations can be easily obtained in a similar way, but is unnecessary. Substituting the coefficient matrices given by (27)-(30) into equation (26), we obtain approximations of $\mathbf{G}, \mathbf{Q}, \mathbf{F}$ and $\mathbf{E}$ for the subinterval of length $\Delta t = \tau$.

Note that all formulations before equation (26) are exact. There are truncation errors caused by disregarding terms of order higher than four in equation (26). For stiff problems, a larger N can be used to further reduce the truncation error in (26).

The use of 2^N type algorithm, however, changes the order of integration of equation (17), because the combination of subintervals does not proceed in exactly the same order as from t_{f} backward to 0. For example, to combine three contiguous subintervals numbered 1-3 into a new subinterval C, we can proceed in two obvious ways: 1) Combine subintervals 1 and 2 to get a new subinterval A, then combine subintervals A and 3 to get the final interval C; 2) Combine subintervals 2 and 3 to get subinterval B, then combine subintervals 1 and B to get the final interval C. Based on the matrix inversion lemma [11] and the combination equations in (23), it is not difficult to prove that the results from both combinations are identical.

For practical implementation, noted that the direct use of the combination equations in (25) would cause serious round-off errors when the length of the subintervals τ is very small. To avoid this, the matrices $\mathbf{F}$ and $\mathbf{E}$ should be written as

$$\mathbf{F} = \mathbf{I}_n + \mathbf{F}',\quad \mathbf{E} = \mathbf{I}_m + \mathbf{E}',\quad \mathbf{F}_{\mathrm{c}} = \mathbf{I}_n + \mathbf{F}'_{\mathrm{c}},\quad \mathbf{E}_{\mathrm{c}} = \mathbf{I}_m + \mathbf{E}'_{\mathrm{c}} \tag{31}$$

and equation (25) should be replaced by

$$\mathbf{G}_{\mathrm{c}} = \mathbf{G} + (\mathbf{I}_n + \mathbf{F}')(\mathbf{G}^{-1} + \mathbf{Q})^{-1}(\mathbf{I}_n + \mathbf{E}') \tag{32a}$$

$$\mathbf{Q}_\mathrm{c} = \mathbf{Q} + (\mathbf{I}_m + \mathbf{E}')(\mathbf{Q}^{-1} + \mathbf{G})^{-1}(\mathbf{I}_m + \mathbf{F}') \tag{32b}$$

$$\mathbf{F}'_\mathrm{c} = -(\mathbf{I}_n + \mathbf{F}')[\mathbf{GQ}(\mathbf{I}_n + \mathbf{GQ})^{-1} + (\mathbf{I}_n + \mathbf{GQ})^{-1}\mathbf{GQ}](\mathbf{I}_n + \mathbf{F}')/2 + 2\mathbf{F}' + \mathbf{F}'^2 \tag{32c}$$

$$\mathbf{E}'_\mathrm{c} = -(\mathbf{I}_m + \mathbf{E}')[\mathbf{QG}(\mathbf{I}_m + \mathbf{QG})^{-1} + (\mathbf{I}_m + \mathbf{QG})^{-1}\mathbf{QG}](\mathbf{I}_m + \mathbf{E}')/2 + 2\mathbf{E}' + \mathbf{E}'^2 \tag{32d}$$

5 Conservative Systems

For continuous time optimal control and elastic wave propagation problems, the system being studied are conservative. In these cases, we have $m = n$, and the matrices $\mathbf{D}$ and $\mathbf{B}$ in dual equations (5) are symmetric with

$$\mathbf{C} = -\mathbf{A}^\mathrm{T}, \quad \mathbf{D} = \mathbf{D}^\mathrm{T}, \quad \mathbf{B} = \mathbf{B}^\mathrm{T}. \tag{33}$$

Similarly, matrices $\mathbf{Q}$ and $\mathbf{G}$ in equation (7) are also symmetric with

$$\mathbf{F} = \mathbf{E}^\mathrm{T}, \quad \mathbf{G} = \mathbf{G}^\mathrm{T}, \quad \mathbf{Q} = \mathbf{Q}^\mathrm{T}. \tag{34}$$

Actually, (7) is the integrated form of (5) based on the theory of Hamiltonian systems [9]. Substituting (34) into (7), we have

$$\mathbf{q}_\mathrm{b} = \mathbf{F}\mathbf{q}_\mathrm{a} - \mathbf{G}\mathbf{p}_\mathrm{b}, \quad \mathbf{p}_\mathrm{a} = \mathbf{Q}\mathbf{q}_\mathrm{a} + \mathbf{F}^\mathrm{T}\mathbf{p}_\mathrm{b}, \tag{35}$$

which can be rewritten as

$$\begin{Bmatrix}\mathbf{q}_\mathrm{b}\\ \mathbf{p}_\mathrm{b}\end{Bmatrix} = \mathbf{T}\begin{Bmatrix}\mathbf{q}_\mathrm{a}\\ \mathbf{p}_\mathrm{a}\end{Bmatrix}, \qquad \text{with} \qquad \mathbf{T} = \begin{bmatrix}\mathbf{F} + \mathbf{GF}^{-\mathrm{T}}\mathbf{Q} & -\mathbf{GF}^{-\mathrm{T}}\\ -\mathbf{F}^{-\mathrm{T}}\mathbf{Q} & \mathbf{F}^{-\mathrm{T}}\end{bmatrix} \tag{36}$$

It is easy to verify that

$$\mathbf{T}^\mathrm{T}\mathbf{JT} = \mathbf{J}, \quad \text{with} \quad \mathbf{J} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}, \tag{37}$$

so $\mathbf{T}$ is a symplectic matrix. Therefore it is easy to find the integration invariant of (35) in the form of

$$\Lambda = \mathbf{v}^\mathrm{T}\mathbf{JPv}, \qquad \mathbf{v} = \begin{Bmatrix}\mathbf{q}\\ \mathbf{p}\end{Bmatrix}, \tag{38}$$

where $\mathbf{v}$ is the state vector and $\mathbf{P}$ is a $2n \times 2n$ matrix. It is necessary to find the condition for $\mathbf{P}$ so as to keep Λ invariant. It is not difficult to show that if the multiplication of $\mathbf{P}$ and $\mathbf{T}$ is commutative, i.e.

$$\mathbf{PT} = \mathbf{TP}, \tag{39}$$

then Λ remains invariant under transformation $\mathbf{T}$. This implies that if $\mathbf{P}$ is any polynomial of $\mathbf{T}$ then Λ is invariant. A good numerical scheme should maintain all invariants in order to correctly represent the behavior of a conservative system.

6 Solution of the Riccati Differential Equation

Based on the discussions in the previous sections, the matrices $\mathbf{G}(\tau)$, $\mathbf{Q}(\tau)$, $\mathbf{F}(\tau)$, and $\mathbf{E}(\tau)$ of the small subinterval of length τ can be computed first by the equations in (26). Then the 2^N type algorithm can be used to calculate these matrices for a typical time interval $[t_a, t_b]$ of length η. Based on these typical interval matrices, the final matrix function $\mathbf{Q}(t)$ can be calculated, which satisfies the differential equation (18d), the same as equation (1). However, the boundary condition that $\mathbf{Q}(t)$ satisfies is given in (14) as

$$\mathbf{Q} = \mathbf{0}, \qquad \text{at } t = t_{\mathrm{f}}, \tag{40}$$

which is not the same as that given in condition (2) for $\mathbf{S}(t)$. On the other hand, the differential equation (17a) for the matrix function $\mathbf{G}(t)$ is the same as equation (3) for the matrix function $\mathbf{T}(t)$, however, the initial condition for $\mathbf{G}(t)$ is (14)

$$\mathbf{G}(0) = \mathbf{0}, \qquad \text{at } t = 0, \tag{41}$$

which is again different from the condition in (4) for $\mathbf{T}(t)$.

To satisfy the boundary condition (2), we need to construct the matrix function $\mathbf{S}(t)$ from the functions $\mathbf{G}(t), \mathbf{Q}(t), \mathbf{F}(t)$ and $\mathbf{E}(t)$ by the equation

$$\mathbf{S}(t) = \mathbf{Q} + \mathbf{E}(\mathbf{I}_m + \mathbf{S}_{\mathrm{f}}\mathbf{G})^{-1}\mathbf{S}_{\mathrm{f}}\mathbf{F}. \tag{42}$$

Since $\mathbf{E} \to \mathbf{I}_m$, $\mathbf{F} \to \mathbf{I}_n$, $\mathbf{G} \to \mathbf{0}$ and $\mathbf{Q} \to \mathbf{0}$ as $t \to t_{\mathrm{f}}$, it can be easily verified that the $\mathbf{S}(t)$ given in equation (42) satisfy the boundary condition given in (2). To show that $\mathbf{S}(t)$ in (42) satisfies equation (1), we need to use the relation $\mathrm{d}\mathbf{X}^{-1}/dt = -\mathbf{X}^{-1}\dot{\mathbf{X}}\mathbf{X}^{-1}$ and equations (18). The physical interpretation for the above equation is the use of combination equation (23) for the interval (t, t_{f}) with matrices $[\mathbf{G}, \mathbf{Q}, \mathbf{F}, \mathbf{E}]$ being treated as interval 1, and at the end $t = t_{\mathrm{f}}$ a fictitious interval with matrices $[\mathbf{0}, \mathbf{S}_{\mathrm{f}}, \mathbf{I}_n, \mathbf{I}_m]$ being treated as interval 2. Here, only the equation (23b) is used to obtain equation (42).

The matrix function $\mathbf{T}(t)$ can be constructed similarly by

$$\mathbf{T} = \mathbf{G} + \mathbf{F}(\mathbf{I}_n + \mathbf{G}_0\mathbf{Q})^{-1}\mathbf{G}_0\mathbf{E}. \tag{43}$$

Since $\mathbf{E} \to \mathbf{I}_m$, $\mathbf{F} \to \mathbf{I}_n$, $\mathbf{G} \to \mathbf{0}$ and $\mathbf{Q} \to \mathbf{0}$ as $t \to 0$, it can be easily verified that $\mathbf{T}(t)$ in (43) satisfies the initial condition given in (4). The verification that $\mathbf{T}(t)$ in (43) satisfies differential equation (3) can be done similarly as for $\mathbf{S}(t)$, except that the equations in (17) should be used. Let $\mathbf{S}(t) = \mathbf{S}_\infty$ when $t \to \infty$, we have

$$-\mathbf{B} + \mathbf{S}_\infty\mathbf{A} - \mathbf{C}\mathbf{S}_\infty + \mathbf{S}_\infty\mathbf{D}\mathbf{S}_\infty = \mathbf{0}, \tag{44}$$

which is the algebraic Riccati equation. For non-conservative systems, such as the source free transport system and elastic wave propagation with damping, the matrix

$\mathbf{S}_\infty$ can also be calculated using the 2^N type algorithm. In this case, the procedure described in the previous sections should be carried out until $\mathbf{E}$ and $\mathbf{F}$ are nearly zero matrices. The matrices $\mathbf{Q}$ and $\mathbf{G}$ are then $\mathbf{S}_\infty$ and $\mathbf{T}_\infty$, respectively. The algebraic Riccati equation for $\mathbf{T}_\infty$ is

$$-\mathbf{D}-\mathbf{T}_\infty\mathbf{C}+\mathbf{A}\mathbf{T}_\infty+\mathbf{T}_\infty\mathbf{B}\mathbf{T}_\infty=\mathbf{0} \tag{45}$$

7 Numerical Examples

Although the 2^N type precise time integration is applicable to problems with finite integration domain $\left[0, t_{\mathrm{f}}\right]$, we choose a infinite domain in these examples to demonstrate its application to algebraic Riccati equation.

Example 1: $n=4$; $m=4$; the system matrices are given as

$$\mathbf{A}=\begin{bmatrix} -0.3379 & 0.5821 & -.1579 & 0.2771 \\ -26.7825 & -.1705 & 0 & 0 \\ -.11821 & -.3059 & -.5523 & 0.9694 \\ 0 & 0 & 0 & 7.6923 \end{bmatrix}, \quad \mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -10-10\mathrm{i} & 0 & 0 \\ 0 & 10 & -100-100\mathrm{i} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{C}=\begin{bmatrix} 0.4+0.4\mathrm{i} & 20+10\mathrm{i} & 0.1 & 0 \\ -0.5 & 0.2 & 0.3 & 0 \\ 0.1 & 0 & 0.5 & 0 \\ -.2 & 0 & -0.1 & -7.7 \end{bmatrix}, \quad \mathbf{D}=\mathrm{diag}\left[0 \quad 0 \quad 0 \quad -10.0-1.0\mathrm{i}\right]\cdot$$

The algebraic Riccati equation (46) was solved by using the 2^N type precise time integration algorithm with $\eta=1.0$ and 4.0, respectively. The calculated matrices $\mathbf{S}_\infty$ are exactly the same. This indicates that the accuracy has reached machine precision, so using a smaller η will not further improve the accuracy. Substituting $\mathbf{S}_\infty$ into (44), we found that the entries in the residual matrix are all smaller than 10^{-10}. Similarly, the calculated matrix $\mathbf{T}_\infty$ also satisfies equation (45) with entries in the residual matrix smaller than 10^{-10}.

Example 2. In this example, we have $n=5; m=1$. The system matrices are

$$\mathbf{A}=\begin{bmatrix} -0.8 & 0.5 & -0.4 & 0.2 & 0.4 \\ 0.3 & -2.1 & 0 & 0 & 0 \\ 0.1 & 0.3 & -0.5 & 0.2 & 0.6 \\ 0 & 0 & 0 & -0.8 & 0.5 \\ 0.3 & 1.0 & 0 & 0 & -0.9 \end{bmatrix}, \quad \mathbf{B}=\begin{bmatrix} 0 \\ -5.0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{C}=[0.5], \quad \mathbf{D}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 2.0 \\ 0 \end{bmatrix}.$$

The algebraic Riccati equations (44) and (45) were solved using $\eta = 0.4$ and $\eta = 5.0$, respectively. The numerical results are exactly the same for both cases. Substitute the matrix $\mathbf{S}_\infty$ and $\mathbf{T}_\infty$ into (44) and (45), respectively, we found again that the entries in the residual matrices are smaller than 10^{-10}.

8 Concluding Remarks

The 2^N type precise time integration algorithm discussed in this paper is very efficient for calculating accurate solutions to matrix Riccati equations. The computer programming for this method is also straightforward since it uses only matrix operations.

References

1. Bellman, R.: Methods of non-linear analysis, vol. 2, Academic Press, NY, 1973.
2. Bittanti, S., Laub, A. J. Willems, J. C.: The Riccati Equation. Springer-Verlag, NY, 1991.
3. Green, M., Limebeer, D. J. N.: Linear Robust control. Prentice-Hall, Englewood Cliff, NJ, 1995.
4. Basar, T., Bernland, P.: H_∞ Optimal Control and Related Mini-Max Design Problems--A dynamic game approach, 2nd Ed. Birkhauser, Boston, 1995.
5. Kenney, C. S., Leipnik, R. B.: Numerical integration of the differential Riccati equation. IEEE Trans, *AC,* 30 (1985) 962.
6. Zhong, W. X., Williams, F. W.: A precise time integration method. Proc. Inst. Mech. Engrs. 208 (1994) 427-430.
7. Zhong, W. X.: Precise integration of eigen-waves for layered-media, in Proc. EPMESC-5. 2 (1995) 1209-1220.
8. W.X. Zhong, 'The method of precise integration of finite strip and wave guide problems', Proc. Intern. Conf. on Computational Method in Struct. and Geotech. Eng. (1994) 50-60.
9. Zhong, W. X., Lin, J. H., Qiu, C. H.: Computational structural mechanics and optimal control---The simulation of substructural chain theory to linear quadratic optimal control problems, Intern. J. Num. Meth. Eng. 33 (1992) 197-211.
10. Angel, E., Bellman, R.: Dynamic Programming and Partial Differential Equations. Academic Press, New York, 1972.
11. Stengel , R. F.: Stochastic Optimal Control. John Wiley and Sons, New York, 1986.