NUMERICAL INTEGRATION OF MATRIX RICCATI DIFFERENTIAL EQUATIONS WITH SOLUTION SINGULARITIES

by

CHARLES K GARRETT

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2013

To my wife Alissa and my gaggle of parents:

William and Tammy Glaze, Charles and Rita Garrett, David and Shirley Grissom.

ACKNOWLEDGEMENTS

Who does one acknowledge for a lifetime of learning? Certainly my parents were instrumental for always letting me be an odd child who enjoys reading technical books. Also, my mother always took the time when moving to make certain we lived in a good school district.

Many of my teachers were also instrumental. Of special importance were the math club sponsors during the 6th grade in Hoover elementary school in Azle, TX who made me realize my talent for mathematics. Then several teachers in my high school, North Crowley High School, in Fort Worth, TX were instrumental in my upbringing. Particularly my science teachers and band directors.

I owe my love of mathematics to the professors at Texas Christian University. I did not know what field I wanted to major in, but the mathematics department there swayed me. Finally, I am indebted to my advisor, Ren-Cang Li, for not only teaching me, but for the contacts he has enabled me to establish.

Last but not least, I thank my wife Alissa for two reasons. The first is, if one doesn't thank their wife in a dissertation that could be bad. Second, when I get in to work mode, I'm not always the most pleasant person to be around, but she always took it well.

January 18, 2013

ABSTRACT

NUMERICAL INTEGRATION OF MATRIX RICCATI DIFFERENTIAL EQUATIONS WITH SOLUTION SINGULARITIES

CHARLES K GARRETT, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: REN-CANG LI

A Matrix Riccati differential equation (MRDE) is a quadratic ODE of the form

$$X' = A_{21} + A_{22}X - XA_{11} - XA_{12}X,$$

where X is a function of t with $X : \mathbb{R} \longrightarrow \mathbb{R}^{n\times m}$ and the A_{ij}'s are constant or functions of t with matrix sizes to respect the size of X. It is well known that MRDEs may have singularities in their solution even if all the A_{ij} are constant.

In this dissertation, several different ideas for the meaning of the solution of an MRDE past a solution singularity are analysed and it is shown how all these ideas are related. Then, a class of numerical methods are given which respect all these ideas. Finally, a robust numerical integration scheme is given based on these numerical methods and several examples are shown to validate the numerical integration scheme.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The matrix Riccati differential equation is an ordinary differential equation of the type

$$X' = A_{21} + A_{22}X - XA_{11} - XA_{12}X, \tag{MRDE}$$

where $X : t \to \mathbb{R}^{n\times m}$ and A_{11}, A_{12}, A_{21}, A_{22} are of size $m \times m$, $m \times n$, $n \times m$, $n \times n$ respectively and can be constant or functions of t. It will always be assumed in this dissertation that A_{11}, A_{12}, A_{21}, A_{22} are at least continuous functions of t. The sizes of the A_{ij} can also be deduced through the partitioning of the matrix A defined as:

$$A := \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{1.1}$$

where A_{21} has the same size as X and A is a square matrix of size $(n+m)\times(n+m)$. The matrix A also plays a key role in solving MRDEs as will be seen later.

Matrix Riccati differential equations are named after Count Jacopo Francesco Riccati [1, §3.2] who was looking at the differential equation satisfied by the slope of the line from the origin to the point (p, q) where p and q satisfy the linear differential equation

$$\begin{pmatrix} p \\ q \end{pmatrix}' = A \begin{pmatrix} p \\ q \end{pmatrix}. \tag{1.2}$$

Then, letting $w = \frac{q}{p}$

$$w' = \frac{q'p - p'q}{p^2} \tag{1.3}$$

$$= \frac{(A_{21}p + A_{22}q)p - (A_{11}p + A_{12}q)q}{p^2} \tag{1.4}$$

$$= A_{21} + A_{22}w - A_{11}w - A_{12}w^2. \tag{1.5}$$

Hence the slope w satisfies the scalar Riccati differential equation.

MRDEs are interesting in a purely mathematical setting as they generalize linear ODEs and Sylvester ODEs

$$X' = A_{21} + A_{22}X \qquad \text{(Linear ODE)}$$

$$X' = A_{21} + A_{22}X - XA_{11}, \qquad \text{(Sylvester ODE)}$$

and they are a class of quadratic differential equations. However MRDEs do not comprise the entire set of quadratic differential equations except in the scalar case.

To see this, take a look at the quadratic factor in (MRDE) for the case when X is a 2 by 1 column vector. Then A_{12} has the form $A_{12} = (a \quad b)$. Therefore,

$$XA_{12}X = \begin{pmatrix} ax_1^2 + bx_1x_2 \\ ax_1x_2 + bx_2^2 \end{pmatrix}. \tag{1.6}$$

This shows that the first entry of $XA_{12}X$ can never have a factor of x_2^2. Thus it is clear that MRDEs in general cannot encompass all quadratic differential equations.

1.2 Applications

Matrix Riccati differential equations arise in numerous fields such as Control Theory [2, 3], linear boundary value problems for ODEs [4, 5, 6], and quantitative finance [7] to name a few. A control theory example and linear boundary value problem example will be shown in detail as these examples are the most common examples for MRDEs.

1.2.1 Control Theory

This section will give a classic example from control theory following the derivation from [2]. The deterministic linear quadratic problem solves the "actuator-plant" model. In this model there is a plant with the governing equation

$$x'(t) = f(x(t), u(t)), \tag{1.7}$$

where $x(t) = (x_1(t), ..., x_n(t))$ is the plant state, and $u(t) = (u_1(t), ..., u_m(t))$ is the control variable.

The engineer knows what trajectory he would like the plant state to operate on. We call this ideal state $x_0(t), u_0(t)$ with $x(t_0) = x_0(t_0)$. Defining

$$\delta x(t) = x(t) - x_0(t) \tag{1.8}$$

$$\delta u(t) = u(t) - u_0(t) \tag{1.9}$$

and linearizing the plant state equation (1.7) via Taylor's theorem, we get

$$\delta x'(t) \approx A_0(t)\delta x(t) + B_0(t)\delta u(t). \tag{1.10}$$

The matrices A_0 and B_0 represent $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ respectively.

To have confidence in the linear approximation, one minimizes with respect to $u(t)$ the quadratic form

$$J_0 = \delta x(T)^T F_0 \delta x(T) + \int_{t_0}^{T} \left(\delta x(t)^T Q_0(t)\delta x(t) + \delta u(t)^T R_0(t)\delta u(t)\right) dt, \tag{1.11}$$

where $F_0(t)$ and $Q_0(t)$ are symmetric positive semidefinite (or definite) and $R_0(t)$ is symmetric positive definite. The functions $Q_0(t)$ and $R_0(t)$ are bounds of the second derivatives which were truncated in the linearisation of $\delta x'(t)$. Thus the integral part of the functional is used to minimize the error in the approximation of linearisation. The part not in the integral is there to ensure $\delta x(T)$ stays near zero for T near t_0.

This finally brings us to the mathematical formulation of the problem. Given

$$\delta x'(t) = A_0(t)\delta x(t) + B_0(t)\delta u(t) \tag{1.12}$$

and a fixed time interval $t \in [t_0, T]$, find $\delta u(t)$ such that J_0 in (1.11) is minimized. The solution to this is

$$\delta u(t) = -G_0(t)\delta x(t) \tag{1.13}$$

where

$$G_0(t) = R_0^{-1}(t)B_0'(t)K_0(t) \tag{1.14}$$

and

$$K_0' = -K_0A_0 - A_0'K_0 - Q_0 + K_0B_0R_0^{-1}B_0'K_0, \quad K_0(T) = F_0. \tag{1.15}$$

So solving the Riccati equation for K_0 is the key to solving the "actuator-plant" problem.

1.2.2 Linear Boundary Value Problems

Another common example where one encounters MRDEs is in solving linear boundary value problems for ODEs. A linear boundary value problem with separated boundary conditions takes the form

$$y' = A(t)y + q(t), \tag{1.16}$$

with boundary conditions

$$B_a y(a) = \beta_1, \qquad B_b y(b) = \beta_2, \tag{1.17}$$

where $A \in \mathbb{R}^{n\times n}$, $y \in \mathbb{R}^{n\times 1}$, $B_a \in \mathbb{R}^{k\times n}$, and $B_b \in \mathbb{R}^{(n-k)\times n}$, and B_a and B_b have full rank. Therefore, we can write

$$B_a = \begin{pmatrix} C_a & D_a \end{pmatrix}, \qquad B_b = \begin{pmatrix} C_b & D_b \end{pmatrix}, \tag{1.18}$$

where D_a and C_b are square and D_a is non-singular. If D_a is singular, then apply a permutation to the ordering of the elements in y so that the corresponding D_a is nonsingular. This is possible since B_a has full rank.

1.2.2.1 Riccati Method

In this section, the Riccati method for solving linear boundary value problems will be summarized. We follow the exposition in [4, §4.5]. To solve the linear boundary value problem using the Riccati method, introduce the transformation $Tw = y$, where

$$T = \begin{pmatrix} I_k & 0 \\ R(t) & I_{n-k} \end{pmatrix}. \tag{1.19}$$

Then $y' = T'w + Tw'$. Solving for w' yields:

$$\begin{aligned} w' &= T^{-1}(y' - T'w) && (1.20)\\ &= T^{-1}(Ay + q - T'w) && (1.21)\\ &= T^{-1}(ATw + q - T'w) && (1.22)\\ &= (T^{-1}AT - T^{-1}T')w + T^{-1}q. && (1.23) \end{aligned}$$

Denote $U = T^{-1}AT - T^{-1}T'$ and $g = T^{-1}q$ to get $w' = Uw + g$. Writing U in block matrix form, we have:

$$\begin{aligned} U &= \begin{pmatrix} I & 0 \\ -R & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ R & I \end{pmatrix} - \begin{pmatrix} I & 0 \\ -R & I \end{pmatrix} \begin{pmatrix} 0 & 0 \\ R' & 0 \end{pmatrix} && (1.24)\\ &= \begin{pmatrix} A_{11} + A_{12}R & A_{12} \\ A_{21} + A_{22}R - RA_{11} - RA_{12}R & -RA_{12} + A_{22} \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ R' & 0 \end{pmatrix}. && (1.25) \end{aligned}$$

If we set $R' = A_{21} + A_{22}R - RA_{11} - RA_{12}R$, then

$$U = \begin{pmatrix} A_{11} + A_{12}R & A_{12} \\ 0 & -RA_{12} + A_{22} \end{pmatrix}, \tag{1.26}$$

and the ODE $w' = Uw + g$ partially decouples as

$$w_1' = (A_{11} + A_{12}R)w_1 + A_{12}w_2 + g_1 \tag{1.27}$$

$$w_2' = (-RA_{12} + A_{22})w_2 + g_2. \tag{1.28}$$

The idea is then to solve (1.28) forward in time and then (1.27) backward in time with an appropriate initial condition for the Riccati equation $R(t)$.

CHAPTER 2

Theory of Solution Singularities of MRDEs

It is well known that singularities can occur in the solution of an MRDE in finite time, even for constant coefficients. A simple example of this is the scalar Riccati differential equation

$$x' = 1 + x^2, \quad x(0) = 0. \tag{2.1}$$

The solution is $x(t) = \tan(t)$ on $t \in [0, \pi/2)$, which goes to infinity as $t \to \pi/2$.

Traditionally, in the theory of ODEs, this is where the solution ends. After all, what would it mean for there to be a solution of x past $t = \pi/2$? Three ideas for the meaning of x past a solution singularity will now be presented given the special structure of MRDEs.

2.1 Linear Transformation

There is a special relationship between a certain linear ODE problem and (MRDE). This relationship has been known since at least the 1920s by Radon [1, §3.1] [8, §2.2].

Theorem 2.1.1. *Consider the linear ODE*

$$P' = AP, \quad P(t) = \begin{pmatrix} S(t) \\ T(t) \end{pmatrix}, \tag{2.2}$$

where A is from (1.1)*, S has size $m \times m$ and T has size $n \times m$. If $T(t_0)S(t_0)^{-1} = X_0$ where X_0 is the initial condition of* (MRDE)*, then*

$$X(t) = T(t)S(t)^{-1} \tag{2.3}$$

where $X(t)$ is the solution of (MRDE).

Proof. First, we will need an identity for $(S^{-1})'$. To get it, notice

$$0 = (SS^{-1})' = S'S^{-1} + S(S^{-1})'. \tag{2.4}$$

Hence

$$(S^{-1})' = -S^{-1}S'S^{-1}. \tag{2.5}$$

Next, if we expand the linear ODE in (2.2), we get

$$S' = A_{11}S + A_{12}T \tag{2.6}$$

$$T' = A_{21}S + A_{22}T. \tag{2.7}$$

Now, to prove the theorem, it will be shown that TS^{-1} satisfies (MRDE).

$$(TS^{-1})' = T'S^{-1} - TS^{-1}S'S^{-1} \tag{2.8}$$

$$= (A_{21}S + A_{22}T)S^{-1} - TS^{-1}(A_{11}S + A_{12}T)S^{-1} \tag{2.9}$$

$$= A_{21} + A_{22}(TS^{-1}) - (TS^{-1})A_{11} - (TS^{-1})A_{12}(TS^{-1}) \tag{2.10}$$

Since, TS^{-1} satisfies (MRDE) and $T(t_0)S(t_0)^{-1} = X_0$, we have $X(t) = T(t)S(t)^{-1}$. □

The formula (2.3) reveals when a singularity in the solution of an MRDE occurs. A singularity in X can occur only if $S(t)$ becomes singular. This is clear since if $A(t)$ is continuous, then it is well known [9] that $P(t)$ exists and is finite on the interval $[t_0, \infty)$. Therefore, when $S(t)$ is not singular, $X(t) = T(t)S(t)^{-1}$ is clearly defined.

To recap, it was just shown that if X has a singularity at t, then $S(t)$ is singular. The converse is also true. I have not seen a proof of this in the literature, so I have the following proof of this.

Theorem 2.1.2. *The solution X to* (MRDE) *with initial condition $X(t_0) = X_0$ has a singularity at t if and only if S is singular at t for the linear ODE* (2.2) *with initial condition satisfying $T(t_0)S(t_0)^{-1} = X_0$.*

Proof. One direction of the theorem was already shown above, so only the other direction will be proven here. Suppose $S(t)$ is singular. Then $S(t)$ has an eigenvector corresponding to an eigenvalue of zero. Thus, there exists an invertible matrix R such that $S(t)R$ has all zeros in its first column. (To see this, put the eigenvector of $S(t)$ corresponding to zero in the first column of R and ensure the columns of R form a basis.) Since R is an invertible matrix, we have

$$\hat{P} = PR = \begin{pmatrix} SR \\ TR \end{pmatrix} = \begin{pmatrix} \hat{S} \\ \hat{T} \end{pmatrix}$$

with $X = TS^{-1} = TRR^{-1}S^{-1} = \hat{T}\hat{S}^{-1}$. Also, it is trivial that $\hat{P}$ solves the linear ODE (2.2) with initial condition $\hat{T}(t_0)\hat{S}(t_0)^{-1} = X_0$.

By construction $\hat{S}(t)$ has all zeros in its first column. Let Φ be the fundamental solution of $P' = AP$ with $\Phi(t_0) = I$. Then $\begin{pmatrix} \hat{S}(t) \\ \hat{T}(t) \end{pmatrix} = \Phi(t) \begin{pmatrix} \hat{S}_0 \\ \hat{T}_0 \end{pmatrix}$. Let P_1 denote the first column of P and in general a subscript of 1 denote the first column of a matrix.

Since, $X(s) = \hat{T}(s)\hat{S}^{-1}(s)$, we have $X(s)\hat{S}(s) = \hat{T}(s)$. Now assume by contradiction that X does not have a singularity at t or more rigorously,

$$\lim_{s \to t} ||X(s)||_\infty < \infty.$$

Then

$$\hat{T}_1(t) = \lim_{s \to t} X(s)\hat{S}_1(s) = \lim_{s \to t} X(s)\hat{S}_1(t) = \mathbf{0},$$

since $\hat{S}_1(t)$ is a column of zeros. This implies $\hat{T}_1(t)$ is a column of zeros as well. From this we get

$$\mathbf{0} = \begin{pmatrix} \hat{S}(t) \\ \hat{T}(t) \end{pmatrix}_1 = \Phi(t) \begin{pmatrix} \hat{S}_0 \\ \hat{T}_0 \end{pmatrix}_1 .$$

But $\hat{S}_0 = S(t_0)R$ is invertible because $S(t_0)$ and R are invertible. Hence, the first column of $\hat{S}_0$ cannot be all zeros. This implies $\Phi(t)$ has an eigenvalue of 0, which contradicts Φ being a fundamental solution of $P' = AP$. Thus

$$\lim_{s \to t} ||X(s)||_\infty = \infty,$$

and hence X has a singularity at t. □

So, what does all this mean in the end? The linear ODE (2.2) can be solved for $t \in [t_0, \infty)$ even if (MRDE) has singularities. Hence, we have our first possible definition of what it means to have a solution beyond a solution singularity of an MRDE. Simply solve the linear ODE (2.2), and then use the transformation (2.3), to get a solution to (MRDE) even past solution singularities.

2.2 Flow on a Grassmannian

In the paper by Schiff and Shnider [10], the solution of the Riccati equation is viewed as a flow on the Grassmannian $Gr(m, m+n)$. The idea of the paper is that the flow on the Grassmannian is always defined, since the Grassmannian is a differentiable compact manifold. But the solution to the Riccati equation is viewed in one local coordinate system, and in this coordinate system a singularity may appear even though the flow of the Riccati equation is still well defined.

Analytically, the solution of (MRDE) past a solution singularity becomes equivalent to solving the linear ODE (2.2) and then transforming the solution via (2.3) just as in the previous section, Section 2.1. Therefore, although there is no analytical

difference, the idea of the flow on the Grassmannian being always well defined gives a geometric justification for using the analytical procedure from Section 2.1 to integrate (MRDE) past solution singularities.

2.3 Generalized Inverse Property

Consider the case when the solution of an MRDE X is a square matrix. If X is invertible for $t \in [t_0, t_f]$, then

$$\begin{aligned}(X^{-1})' &= -X^{-1}X'X^{-1} \\ &= -X^{-1}(A_{21} + A_{22}X - XA_{11} - XA_{12}X)X^{-1} \\ &= -X^{-1}A_{21}X^{-1} - X^{-1}A_{22} + A_{11}X^{-1} + A_{12}.\end{aligned}$$

Making the substitution $U = X^{-1}$ and rearranging terms, we get a new MRDE, which will be called the complementary MRDE

$$U' = A_{12} + A_{11}U - UA_{22} - UA_{21}U. \tag{CMRDE}$$

Just as the MRDE was defined by the matrix A in (1.1), the CMRDE is defined by

$$A_c := \begin{pmatrix} A_{22} & A_{21} \\ A_{12} & A_{11} \end{pmatrix}, \tag{2.11}$$

which is equal to

$$A_c = K^T A K \quad \text{where} \quad K = \begin{pmatrix} 0 & I_m \\ I_n & 0 \end{pmatrix}. \tag{2.12}$$

Even if X is not square, (CMRDE) still makes sense. A generalized inverse property was proposed by Li and Kahan [11]. As there are several definitions of a generalized inverse, in this document we will define a generalized inverse as follows.

Definition 2.3.1. *A generalized inverse of a matrix A is a matrix B such that either $BA = I$ or $AB = I$.*

A few notes should be said about this definition. Suppose A from the definition above has size $n \times m$.

- For A to have a generalized inverse, A must have full rank.
- If $n = m$ and A has full rank, then there is only one generalized inverse which is A^{-1}.
- If $n > m$ and A has full rank, then there are infinitely many generalized inverses all of which are left inverses only, i.e. $BA = I$ but there is no B such that $AB = I$.
- If $n < m$ and A has full rank, then there are infinitely many generalized inverses all of which are right inverses only, i.e. $AB = I$ but there is no B such that $BA = I$.

Going back to the generalized inverse property for MRDEs proposed by Li and Kahan, the following theorem will be given with two proofs, since the proof methods are very different. The first proof is by Li and Kahan from [11] and the second proof is by me.

Theorem 2.3.2. *If $U_0X_0 = I$ (or $X_0U_0 = I$), and if the solutions $X(t)$ to* (MRDE) *and $U(t)$ to* (CMRDE) *have no singularities in some interval $\mathcal{I} = [t_0, T]$, then $U(t)X(t) = I$ (or $X(t)U(t) = I$, respectively).*

Proof. If $U_0X_0 = I$, then $(UX) = I$ solves the following initial value problem for (UX):

$$\begin{aligned}(UX)' &= (A_{12} - UA_{22} + A_{11}U - UA_{21}U)X + U(A_{21} - XA_{11} + A_{22}X - XA_{12}X) \\ &= A_{12}X - (UX)A_{12}X + A_{11}(UX) - (UX)A_{11} - UA_{21}(UX) + UA_{21} \\ &= [I - (UX)]A_{21}X + A_{11}[(UX) - I] - [(UX) - I]A_{11} - UA_{21}[(UX) - I].\end{aligned}$$

Since $(UX) = I$ is an equilibrium point and $(U_0X_0) = I$, we have $(UX) = I$ on the entire interval $\mathcal{I}$. □

This second proof uses the linear transformation from Section 2.1.

Proof. Let A be the matrix associated with (MRDE) and K^TAK the matrix associated with (CMRDE). Then the associated linear systems of (MRDE) and (CMRDE) respectively are

$$P' = AP, \qquad P_0 = \begin{pmatrix} I_m \\ X_0 \end{pmatrix} \tag{2.13}$$

$$Q' = K^TAKQ, \qquad Q_0 = \begin{pmatrix} I_n \\ U_0 \end{pmatrix}. \tag{2.14}$$

Then

$$P = \Phi P_0 \tag{2.15}$$

$$Q = \Phi_c Q_0, \tag{2.16}$$

where Φ and Φ_c are the fundamental matrix solutions for the previous linear ODEs (2.13), (2.14).

First, notice that $\Phi_c = K^T\Phi K$. This can be seen from $I = \Phi_0 = K^T\Phi_0 K$ and by showing that $K^T\Phi K$ satisfies (2.14):

$$\begin{aligned} (K^T\Phi K)' &= K^T\Phi' K \\ &= K^TA\Phi K \\ &= K^TAK(K^T\Phi K). \end{aligned}$$

So $Q = \Phi_c Q_0 = K^T\Phi K Q_0$. Break Φ up into four blocks just as A in (1.1) is broken into four blocks. Then

$$P = \begin{pmatrix} S \\ T \end{pmatrix} = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix} \begin{pmatrix} I \\ X_0 \end{pmatrix}, \tag{2.17}$$

and therefore

$$X = TS^{-1} = (\Phi_{21} + \Phi_{22}X_0)(\Phi_{11} + \Phi_{12}X_0)^{-1}. \tag{2.18}$$

Similarly

$$U = (\Phi_{12} + \Phi_{11}U_0)(\Phi_{22} + \Phi_{21}U_0)^{-1}. \tag{2.19}$$

Thus

$$\begin{aligned} UX &= (\Phi_{12} + \Phi_{11}U_0)(\Phi_{22} + \Phi_{21}U_0)^{-1}(\Phi_{21} + \Phi_{22}X_0)(\Phi_{11} + \Phi_{12}X_0)^{-1} \\ &= (\Phi_{12} + \Phi_{11}U_0)X_0(\Phi_{11} + \Phi_{12}X_0)^{-1} \\ &= (\Phi_{12}X_0 + \Phi_{11})(\Phi_{11} + \Phi_{12}X_0)^{-1} \\ &= I. \end{aligned}$$

The step with $(\Phi_{22} + \Phi_{21}U_0)^{-1}(\Phi_{21} + \Phi_{22}X_0) = X_0$ was not shown. To see this just notice $(\Phi_{21} + \Phi_{22}X_0) = (\Phi_{22} + \Phi_{21}U_0)X_0$. □

2.3.1 Using the Generalized Inverse Property

The idea proposed by Li and Kahan for the solution of an MRDE past a solution singularity is thus. Suppose there is a singularity at $X(t^*)$. If X is square, one can consider solving the CMRDE for $U = X^{-1}$. If U does not have a singularity in a neighborhood of t^*, then one switches to solving for U in this neighborhood, and then switches back to X after the singularity.

If X is not square, we need more to be a bit more careful. First, suppose $t^* \in \mathcal{I} = [t^* - a, t^* + b]$, and X does not have a singularity at $t^* - a$. Also suppose X has size $n \times m$ where $n > m$ (the case $n < m$ is similar). Then we consider the family of generalized inverses at $t = t^* - a$

$$\mathbb{U} = \{U : UX(t^* - a) = I\}.$$

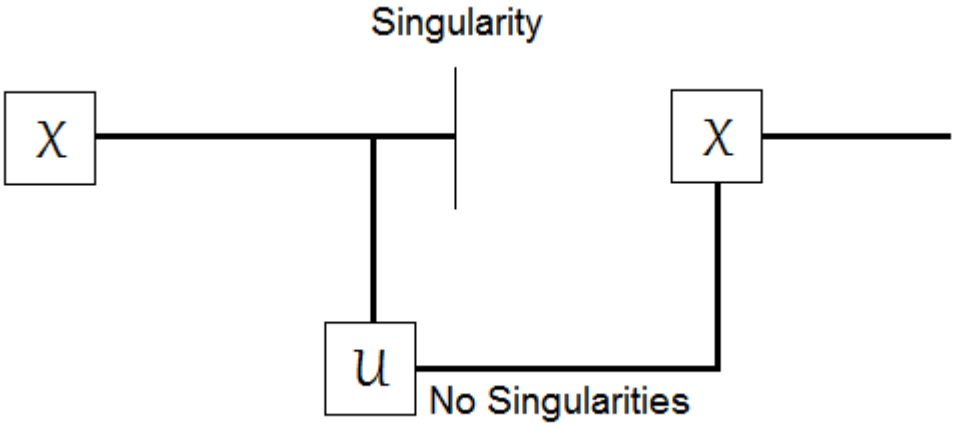

Figure 2.1. A figure denoting the idea of going to (CMRDE) if (MRDE) has a singularity.

Then, solve (CMRDE) an infinite number of times using the family of Us in $\mathbb{U}$ as initial values. Assuming this infinity of solutions has no singularities in $\mathcal{I}$, then we have an infinite family of solutions to (CMRDE) for time $t^* + b$. Then if this family of solutions at $t^* + b$ has enough information, one can reconstruct X after the singularity t^*.

This idea of a family of generalized inverses having enough information to reconstruct X was made more precise by the following lemma from Li and Kahan in [11].

Lemma 2.3.3. *Suppose the $n \times m$ matrix X has full row rank, and let $\mathbb{U} = \{U : XU = I\}$. If $\hat{X}U = I$ for all U in a nonempty relatively open set of $\mathbb{U}$, then $\hat{X} = X$, or in other words, X is uniquely determined by a nonempty relatively open set in the collection of its right generalized inverses.*

I have also created a concept for the size of the set of Us needed to reconstruct X. Although I am not certain of its usefulness for the ideas here, it may be of some theoretical value to others. First, I need a technical lemma.

Lemma 2.3.4. *Let $X \in \mathbb{C}^{n\times m}$ be of full rank with $n < m$. Then $\mathcal{U} = \{U \in \mathbb{C}^{m\times n} | XU = I\}$ is nonempty. Let $\mathcal{V} = \{V \in \mathbb{C}^{m\times n} | XV = 0\}$. Then $\forall U_0 \in \mathcal{U}$ we have $\mathcal{U} = \mathcal{V} + U_0$. Furthermore, $\mathcal{V}$ is a subspace of $\mathbb{C}^{m\times n}$ with $\dim \mathcal{V} = n(m-n)$.*

Proof. First, since X has full rank, it is trivial that $\mathcal{U}$ is nonempty. Now, suppose $U_0 \in \mathcal{U}$. If $U \in \mathcal{U}$ then $U = (U - U_0) + U_0 \in \mathcal{V} + U_0$ since $X(U - U_0) = I - I = 0$. If $V \in \mathcal{V}$ then $V + U_0 \in \mathcal{U}$ since $X(V + U_0) = 0 + I = I$. Hence $\mathcal{U} = \mathcal{V} + U_0$.

Let V_1 and V_2 be in $\mathcal{V}$. Then $X(\alpha V_1 + \beta V_2) = \alpha XV_1 + \beta XV_2 = 0$, which implies $\alpha V_1 + \beta V_2 \in \mathcal{V}$. So $\mathcal{V}$ is a subspace of $\mathbb{C}^{m\times n}$. Now $\mathcal{V} = ker(X)$ where X is the linear map from $\mathbb{C}^{m\times n}$ to $\mathbb{C}^{n\times n}$. This linear map is equivalent to the linear map $\tilde{X}$ from $\mathbb{C}^{nm\times 1}$ to $\mathbb{C}^{n^2\times 1}$ via

$$XA \sim \tilde{X}\tilde{A} = \begin{pmatrix} X & 0 & \cdots & 0 \\ 0 & X & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & X \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}, \tag{2.20}$$

where A_k denotes the kth column of the matrix A. X can be written in Jordan form as $X = PJP^{-1}$, so

$$\tilde{X} = \begin{pmatrix} P & 0 & \cdots & 0 \\ 0 & P & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P \end{pmatrix} \begin{pmatrix} J & 0 & \cdots & 0 \\ 0 & J & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J \end{pmatrix} \begin{pmatrix} P^{-1} & 0 & \cdots & 0 \\ 0 & P^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P^{-1} \end{pmatrix} \tag{2.21}$$

And now it is easy to see that $\dim\ker(\tilde{X}) = n \dim\ker(J) = n \dim\ker(X) = n(m-n)$ where X is viewed as a linear map of column vectors. □

Theorem 2.3.5. *Let $X \in \mathbb{C}^{n\times m}$ with $n < m$. Suppose $\mathcal{V}'$ is a subspace of $\mathbb{C}^{m\times n}$ with the property $\forall V \in \mathcal{V}'$, $XV = 0$ and suppose there exists U_0 such that $XU_0 = I$. If $\dim \mathcal{V}' = n(m-n)$ then $\mathcal{V} = \mathcal{V}'$ and X is unique.*

Proof. Clearly $\mathcal{V}'$ is a subspace of $\mathcal{V}$. Since there exists U_0 such that $XU_0 = I$, we know X has full rank. From the previous lemma, we know $\dim \mathcal{V} = n(m-n) = \dim \mathcal{V}'$. Hence $\mathcal{V}' = \mathcal{V}$ and thus $\mathcal{U} = \mathcal{V} + U_0$ as shown in Lemma 2.3.4.

To see X is unique, notice that $\mathcal{V}' = \ker(X)$ where X is the linear transformation from $\mathbb{C}^{m\times n}$ to $\mathbb{C}^{n\times n}$. Let $B_1, B_2, \ldots, B_{n(m-n)}$ be a basis of $\mathcal{V}'$. Now, define $C_1, C_2, \ldots, C_{n^2}$ in the following way.

- $C_1 = B_1$ but replace the first column of B_1 with the first column of U_0.
- $C_2 = B_1$ but replace the first column of B_1 with the second column of U_0.
- $\vdots$
- $C_n = B_1$ but replace the first column of B_1 with the last column of U_0.
- $C_{n+1} = B_1$ but replace the second column of B_1 with the first column of U_0.
- $\vdots$
- $C_{n^2} = B_1$ but replace the last column of B_1 with the last column of U_0.

Let $\alpha_1, \ldots, \alpha_{n^2}$ be scalars. Then,

$$X(\alpha_1 C_1 + \ldots + \alpha_{n^2} C_{n^2}) = \begin{pmatrix} \alpha_1 & \alpha_{n+1} & \ldots & \alpha_{n^2-n+1} \\ \alpha_2 & \alpha_{n+2} & \ldots & \alpha_{n^2-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_n & \alpha_{2n} & \ldots & \alpha_{n^2} \end{pmatrix}. \tag{2.22}$$

Notice, the matrix above is zero if and only if $\alpha_1 = \ldots = \alpha_{n^2} = 0$. Hence $\alpha_1 C_1 + \ldots + \alpha_{n^2} C_{n^2} = 0$ if and only if $\alpha_1 = \ldots = \alpha_{n^2} = 0$ which implies $C_1, C_2, \ldots, C_{n^2}$ are linearly independent. $C_1, C_2, \ldots, C_{n^2}$ are also clearly independent from $B_1, B_2, \ldots, B_{n(m-n)}$. Therefore, these two sets of matrices define a basis of X. Hence X is uniquely determined by $\mathcal{V}'$ and U_0. □

Li and Kahan tried to give a continuity argument that if one starts close enough to the singularity, the family of Us will not contain a singularity in a neighborhood

of t^* and there will be enough information to recreate X from the Us. The following counterexample shows this is not true.

Let

$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$$

solve the MRDE

$$X' = I + X^2,$$

where x_{11}, x_{12}, x_{21}, x_{22} are scalar functions of t. Consider the initial conditions

$$x_{21}(0) - x_{12}(0) = 0 \quad x_{11}(0) = 1 \quad x_{22}(0) = -1.$$

The solution to this system is

$$X(t) = \begin{pmatrix} \tan(t + \frac{\pi}{4}) & 0 \\ 0 & \tan(t - \frac{\pi}{4}) \end{pmatrix}.$$

But

$$\lim_{t \to \pi/4^-} X(t) = \begin{pmatrix} \infty & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \lim_{t \to \pi/4^-} X^{-1}(t) = \begin{pmatrix} 0 & 0 \\ 0 & \infty \end{pmatrix}.$$

This shows that when X has a singularity at some time t^*, the solution to (CMRDE), X^{-1}, may also have a singularity at t^*.

2.4 Spacing of Singularities

Under certain conditions, it can be shown that in a finite interval, there are only a finite number of singularities to the solution of (MRDE).

Theorem 2.4.1. *Assume $A(t)$ in* (1.1) *is analytic on the open interval U containing a closed finite interval $\mathcal{I}$. Then there are at most a finite number of singularities in the solution of* (MRDE).

Proof. Since each entry of $P(t)$ from (2.2) is analytic, we know that each entry of $S(t)$ is analytic. The determinant function is comprised of a finite number of multiplications, additions, and subtractions of the entries of $S(t)$. Thus $\det S(t)$ is analytic as well.

But every analytic function has the property that it is either zero everywhere, or that it has isolated zeros. Since $S(t_0) = I$, we have $\det S(t_0) = 1$. Hence, $\det S(t_0)$ must have only isolated zeros. □

Given that X has singularities exactly when S has zeros, this theorem implies that X has isolated singularities.

CHAPTER 3

Theory of Numerical Solutions to MRDEs with Solution Singularities

Three theoretical ideas were discussed in the last chapter about the theoretical meaning of a solution past a singularity of an MRDE. In this chapter, we go through those theoretical ideas and give an analysis of numerical solutions to (MRDE) respecting each idea.

3.1 Using Radon's Transformation

From the last chapter, the first and simplest theory to solve an MRDE past solution singularities is simply to solve the linear ODE $P' = AP$ (2.2) and then apply the Radon transformation $X = TS^{-1}$ (2.3).

This idea naturally yields a numerical method. Approximate the solution of the linear ODE (2.2) with a standard numerical method, and then apply the transformation (2.3). Unfortunately, this is not always stable when P is a matrix with more than one column.

For instance, if A is symmetric with largest eigenvalue λ_1 and corresponding eigenvector v_1 with all other eigenvalues smaller than λ_1, then all the columns of P will converge to a multiple of v_1 (assuming each column of P_0 is not orthogonal to v_1). To be exact, when I say a vector converges to a multiple of some other vector, I mean the angle between the vectors approaches zero or for this case

$$\lim_{t\to\infty} \frac{v_1^T P_k(t)}{|v_1||P_k(t)|} = 1, \tag{3.1}$$

where P_k is the kth column of P.

This occurs because, if A is symmetric, then A has an orthogonal basis of eigenvectors and hence can be diagonalized as

$$A = VDV^{-1}. \tag{3.2}$$

Therefore the solution to $P' = AP$ is

$$P = Ve^{D(t-t_0)}V^{-1}P_0. \tag{3.3}$$

Each column of P can be written as

$$P_k(t_0) = \alpha_{1k}v_1 + \ldots + \alpha_{Nk}v_N, \tag{3.4}$$

where $N = (n + m)$ which implies

$$P_k(t) = \alpha_{1k}e^{\lambda_1(t-t_0)}v_1 + \ldots + \alpha_{Nk}e^{\lambda_N(t-t_0)}v_N. \tag{3.5}$$

Assuming that

- $\lambda_1 > \lambda_j$ for all $j \neq 1$
- $\alpha_{1k} \neq 0$ for all k
- $||v_k||_2 = 1$ for all k

then

$$\frac{v_1^T P_k(t)}{|v_1||P_k(t)|} = \frac{\alpha_{1k}e^{\lambda_1(t-t_0)}}{\sqrt{\alpha_{1k}^2 e^{2\lambda_1(t-t_0)} + \ldots + \alpha_{Nk}^2 e^{2\lambda_N(t-t_0)}}} \longrightarrow 1. \tag{3.6}$$

Now we know that P's columns approach a multiple of v_1. Then we attempt to find $S(t)^{-1}$ where S is defined in (2.2). But the columns of S will all be close to a multiple of the same vector. Hence S will become *nearly* singular, cond(S) will grow large, and the computation for S^{-1} will become unstable as $t \longrightarrow \infty$.

3.1.1 Fixing the Problem of S^{-1}

It is possible to amend the procedure above to take care of the instability in S. Using the previous method, which will be called Radon's method here, for computing the solution of (MRDE), we have the following idea:

$$X_0 \to P_0 = \begin{pmatrix} I \\ X_0 \end{pmatrix} \to P_1 \to P_2 \to \ldots \to P_n = \begin{pmatrix} S_n \\ T_n \end{pmatrix} \to X_n = T_n S_n^{-1} \tag{Radon Method}$$

where $P_k \to P_{k+1}$ means using some numerical method to approximate $P(t_{k+1})$ given the initial condition $P(t_k) = P_k$. As stated before, $\text{cond}(S_n)$ may be large making this method unstable.

By Theorem 2.1.1, we know that $X(t_k) = T(t_k)S(t_k)^{-1}$. So if we have an approximation of the linear system P_k at time t_k, we have that P_k and $P_k R$ where R is a square invertible matrix yield the same solution to (MRDE). To see this, notice the approximation to $X(t_k)$ is

$$X_k = T_k S_k^{-1} \tag{3.7}$$

and using $P_k R = \begin{pmatrix} S_k R \\ T_k R \end{pmatrix}$ yields the solution

$$(T_k R)(S_k R)^{-1} = T_k R R^{-1} S_k^{-1} = T_k S_k^{-1} = X_k. \tag{3.8}$$

Now the question becomes, what value of R should be used? If $R = S_k^{-1}$, then $P_k R = \begin{pmatrix} I \\ X_k \end{pmatrix}$, making the top square matrix of $P_k R$ as well conditioned as possible. Another possibility is to use a 'skinny' QR decomposition. With this, we have $P_k = QR$ where $Q \in \mathbb{R}^{(n+m)\times m}$ (or $Q \in \mathbb{C}^{(n+m)\times m}$) with $Q^T Q = I$ (or $Q^* Q = I$) and R is a square upper triangular matrix. Then $Q = P_k R^{-1}$ is used. The merit in this transformation is that if the MRDE has a singularity near t_k, then

the QR decomposition will not give large numbers for the solution of $P_k R^{-1}$, but $P_k S_k^{-1} = \begin{pmatrix} I \\ X_k \end{pmatrix}$ will give large numbers.

Hence, we have the two modified Radon Methods as shown below.

$$X_0 \to P_0 \to P_1 \to P_1 S_1^{-1} \to P_2 \to P_2 S_2^{-1} \to \ldots \to P_n \to X_n = T_n S_n^{-1}$$

(Radon Inverse Method)

$$X_0 \to P_0 \to Q_0 \to P_1 \to Q_1 \to P_2 \to Q_2 \to \ldots \to P_n \to X_n = T_n S_n^{-1}$$

(Radon QR Method)

The Q_k in (Radon QR Method) is the Q in the skinny QR decomposition of P_k.

3.1.2 Example

As an example of all three Radon methods will be given. The example uses the matrix $A = VDV^{-1}$, where A defines the MRDE as in (2.2), V is a random 4 by 4 matrix created by Matlab's rand(4) function and D is a diagonal matrix with entries 4, 1, 1, and 1. Note, in this example A is not symmetric. The initial condition is $X_0 = \mathbf{0}_2$ and we will use $P_0 = \begin{pmatrix} I_2 \\ \mathbf{0}_2 \end{pmatrix}$ as the initial condition for P.

The numerical method used to approximate the solution to the linear ODE at each time step is Newton's method for simplicity

$$P^{n+1} = (I + \Delta t A) P^n. \tag{3.9}$$

To solve for the associated solution of the MRDE, we have the three following methods. The first method is the "Radon Method".

```
t = t0

P = P0

while t < tf
```

```
    P = (I + dt * A) * P
    t = t + dt
end while
X = P(3:4,:) * P(1:2,:)^(-1)
```

The second method is the "Radon Inverse Method".

```
t = t0
X = X0
while t < tf
    P = [I; X]
    P = (I + dt * A) * P
    X = P(3:4,:) * P(1:2,:)^(-1)
    t = t + dt
end while
```

The third method is the "Radon QR Method".

```
t = t0
X = X0
while t < tf
    Q = skinny_qr(P)
    P = Q
    P = (I + dt * A) * P
    t = t + dt
end while
X = P(3:4,:) * P(1:2,:)^(-1)
```

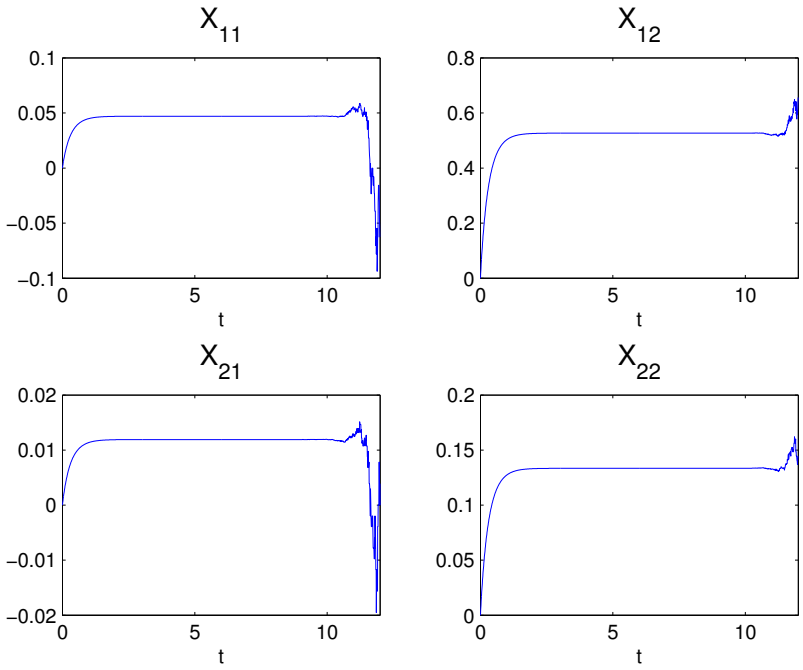

Figure 3.1. $X(t)$ computed using the Radon method.

The pseudo code uses the notation $[I; X]$ to mean $\begin{pmatrix} I \\ X \end{pmatrix}$, the notation $P(3:4,:)$ to mean the 3rd and 4th rows of P, and the notation $P(1:2,:)$ to mean the 1st and 2nd rows of P.

The numerical instabilities in using the plain Radon method are evident when viewing the computed solution for $X(t)$ in Figure 3.1 as well as when viewing the plot of the relative error in Figure 3.4. Also as predicted by the theory, the condition number of $S(t)$ becomes huge as t increases when using the plain Radon method as shown in Figure 3.5. This numerical instability does not seem to occur however for the Radon inverse method and Radon QR method as shown in Figures 3.2, 3.3 of the computed solutions and Figure 3.4 of the relative error.

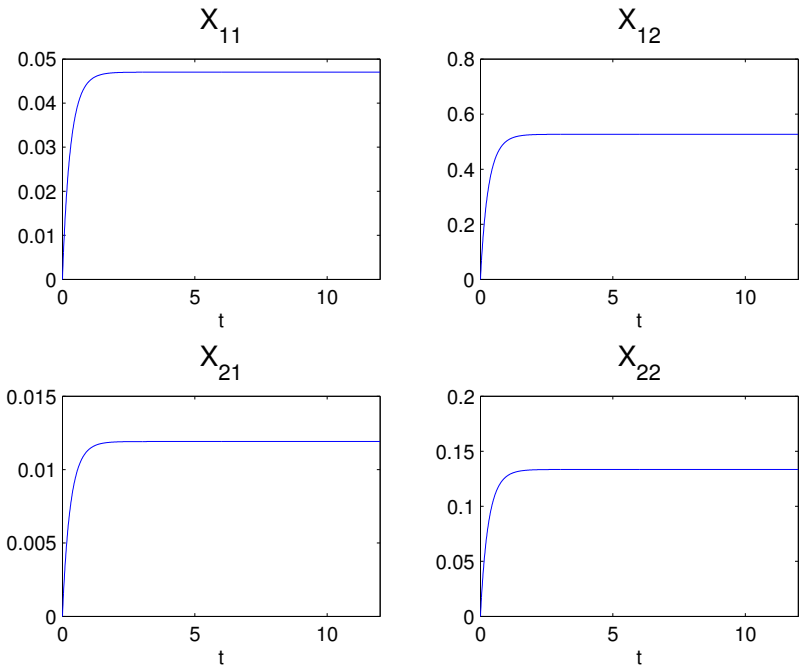

Figure 3.2. $X(t)$ computed using the Radon inverse method.

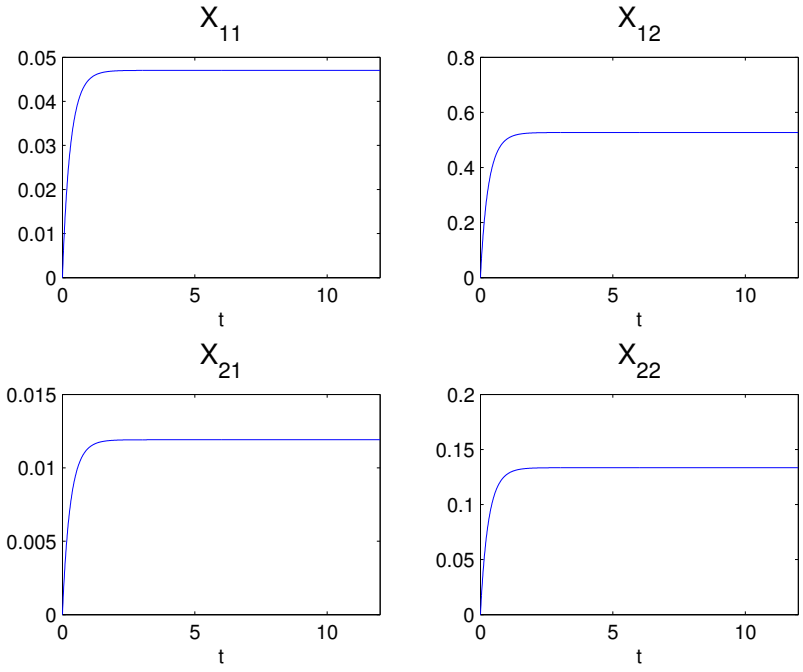

Figure 3.3. $X(t)$ computed using the Radon QR method.

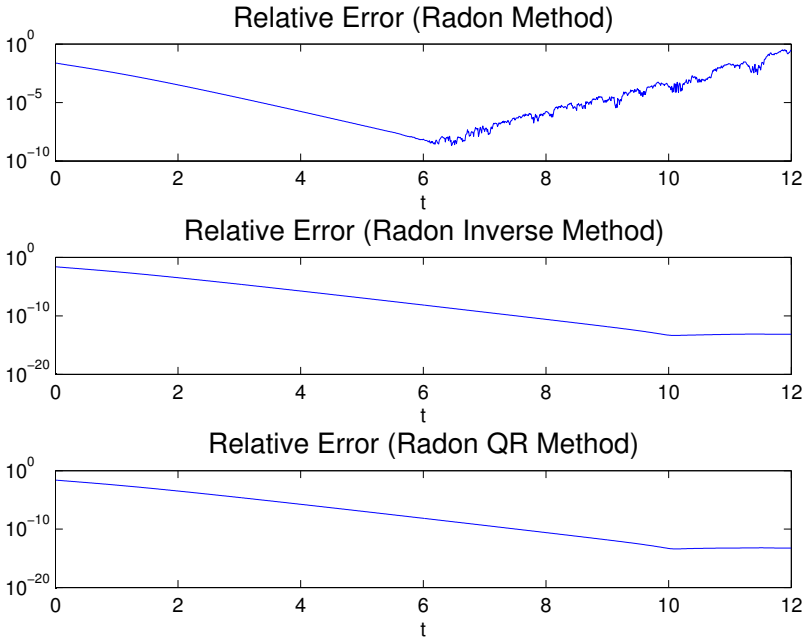

Figure 3.4. Relative error using the three Radon methods.

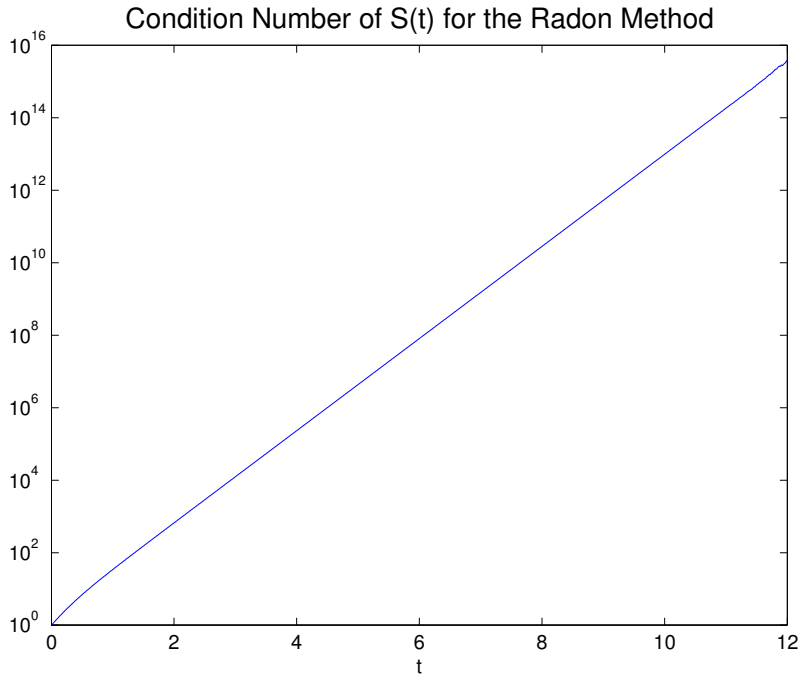

Figure 3.5. Condition number of $S(t)$ using the Radon method.

3.2 Möbius Schemes

In the paper by Schiff and Shnider [10], a set of methods called Möbius Schemes was used to integrate past solution singularities of MRDEs. The scheme uses the one-step method:

$$X_{i+1} = \left[\hat{A}_{21} + \hat{A}_{22}X_i\right]\left[\hat{A}_{11} + \hat{A}_{12}X_i\right]^{-1}, \tag{3.10}$$

where

$$\begin{aligned}
\hat{A}_{11} &= I + hA_{11} + o(h),\\
\hat{A}_{12} &= hA_{12} + o(h),\\
\hat{A}_{21} &= hA_{21} + o(h),\\
\hat{A}_{22} &= I + hA_{22} + o(h).
\end{aligned}$$

Now, it will be shown how any Radon inverse method based on a one step numerical method is a Möbius Scheme.

Almost any one-step numerical method to solve $P' = AP$ can be described [12] as

$$\alpha_1 P_1 + \alpha_0 P_0 = h\phi(P_1, P_0, t_0, h), \tag{3.11}$$

where $\alpha_1 = 1$. For the method to be consistent, we must have

1. $\alpha_0 + \alpha_1 = 0$,
2. $P'(t_0) = \phi(P_0, P_0, t_0, 0)$.

The first consistency criterion requires $\alpha_0 = -1$. The second requires $\phi(P_0, P_0, t_0, 0) = A(t_0)P_0$. Assuming ϕ is a continuous function of the 1st and 4th parameters and assuming y is continuous, we have

$$\lim_{h\to 0} \phi(P_1, P_0, t_0, h) = \phi(P_0, P_0, t_0, 0) = A(t_0)P_0, \tag{3.12}$$

or equivalently

$$h\phi(P_1, P_0, t_0, h) = \hat{A}P_0, \tag{3.13}$$

where $\hat{A} = hA(t_0) + o(h)$. Putting all of this together yields

$$P_1 = P_0 + \hat{A}P_0 = (I + \hat{A})P_0 = \begin{pmatrix} I + \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & I + \hat{A}_{22} \end{pmatrix} \begin{pmatrix} I \\ X_0 \end{pmatrix}. \tag{3.14}$$

Then using the Radon transformation (2.3) we get exactly (3.10).

3.3 GIP Integrators

Following the order of the last chapter, we still have to look at methods preserving the generalized inverse property. Suppose we have a numerical method of the form:

$$X_{i+1} = \mathbb{F}(X_i, A, t_i, h). \tag{3.15}$$

Definition 3.3.1. *A GIP integrator is defined as any method $\mathbb{F}$ which has the following property. If $X_iU_i = I$ (or $U_iX_i = I$) then $X_{i+1}U_{i+1} = I$ (or $U_{i+1}X_{i+1} = I$) where $X_{i+1} = \mathbb{F}(X_i, A, t_i, h)$ and $U_{i+1} = \mathbb{F}(U_i, K^TAK, t_i, h)$.*

The generalized inverse property is an important property of MRDEs and hence we wish to preserve this property numerically. The following theorem gives a condition under which a Radon method using a one-step numerical method to solve an MRDE is a GIP integrator.

Theorem 3.3.2. *Suppose a one-step numerical method to solve the linear system* (2.2) *is of the form*

$$Q = P + hf(A)P, \tag{3.16}$$

where $P \approx P(t_i)$, $Q \approx P(t_i + h)$, and f is some function of A satisfying $f(A) = A(t_0) + o(1)$. If

$$f(K^TAK) = K^Tf(A)K, \tag{3.17}$$

then the one-step numerical method (3.16) *yields a GIP Integrator.*

Proof. First, note that the definition of f yields the most general type of one-step method for a linear ODE as discussed in the previous section. Now, let $B = f(A)$ and partition B in the same way as A is partitioned in (1.1). Denote $X = P_2P_1^{-1}$ and $Z = Q_2Q_1^{-1}$, where $P = \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}$, $Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$, and P_1 and Q_1 are square matrices. The approximate solution to the MRDE is given by

$$Z = Q_2Q_1^{-1} \tag{3.18}$$

$$= (P_2 + hB_{21}P_1 + hB_{22}P_2)(P_1 + hB_{11}P_1 + hB_{12}P_2)^{-1} \tag{3.19}$$

$$= (P_2 + hB_{21}P_1 + hB_{22}P_2)P_1^{-1}P_1(P_1 + hB_{11}P_1 + hB_{12}P_2)^{-1} \tag{3.20}$$

$$= (X + hB_{21} + hB_{22}X)(I + hB_{11} + hB_{12}X)^{-1}. \tag{3.21}$$

Consider $\hat{Q} = \hat{P} + hf(K^TAK)\hat{P}$ to solve the complementary MRDE where $U = \hat{P}_2\hat{P}_1^{-1}$ and $W = \hat{Q}_2\hat{Q}_1^{-1}$. Using the assumption $f(K^TAK) = K^Tf(A)K$, we have $\hat{Q} = \hat{P} + hK^Tf(A)K\hat{P}$. The approximate solution to (CMRDE) is given by (using the same logic as above)

$$W = (U + hB_{12} + hB_{11}U)(I + hB_{22} + hB_{21}U)^{-1}. \tag{3.22}$$

For the numerical scheme to be a GIP integrator, we must have $ZW = I$ when $XU = I$. But notice, if $XU = I$ then

$$(I + hB_{11} + hB_{12}X)U = (U + hB_{11}U + hB_{12}). \tag{3.23}$$

So

$$ZW = (X + hB_{21} + hB_{22}X)(I + hB_{11} + hB_{12}X)^{-1}(U + hB_{12} + hB_{11}U)(I + hB_{22} + hB_{21}U)^{-1} \tag{3.24}$$

$$= (X + hB_{21} + hB_{22}X)U(I + hB_{22} + hB_{21}U)^{-1} \tag{3.25}$$

$$= (I + hB_{21}U + hB_{22})(I + hB_{22} + hB_{21}U)^{-1} \tag{3.26}$$

$$= I. \tag{3.27}$$

□

It turns out that almost all one-step numerical methods that one naturally creates are GIP Integrators. For instance, if $f(A)$ consists of multiplications, additions, and derivatives of A evaluated at different points, which is the case for all the examples in Schiff and Shnider [10], Li and Kahan [11], and this paper, then $f(K^T AK) = K^T f(A)K$. Therefore, since it is a simple matter to numerically preserve the generalized inverse property, all numerical methods in this document will be GIP integrators.

CHAPTER 4

Numerical Implementation of GIP Integrators

In this chapter, it will be shown that embedded Runge-Kutta methods can be used to solve MRDEs to get good local error estimates for time stepping criteria. Also, a posteriori global error estimates will be derived. These methods may even be made into a black box solution for solving MRDEs.

4.0.1 Runge-Kutta Methods

A general s-stage Runge-Kutta method [13, 14, 15, 12] for $y' = f(t, y)$ is defined by

$$\mathbf{k}_i = f\left(\tau + \gamma_i h, \mathbf{y} + h\sum_{j=1}^{s} \alpha_{ij}\mathbf{k}_j\right), \quad i = 1, 2, \ldots, s \tag{4.1}$$

$$\mathbf{Y} = \mathbf{y} + h\sum_{i=1}^{s} \beta_i \mathbf{k}_i \tag{4.2}$$

where $\mathbf{y} \approx y(t)$, $\mathbf{Y} \approx y(t+h)$, and $\gamma_i = \sum_{j=1}^{s} \alpha_{ij}$ for $i = 1, 2, \ldots, s$. Conveniently, Runge-Kutta methods are identified as a *Butcher array:*

$$\begin{array}{c|cccc} \gamma_1 & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1s} \\ \gamma_2 & \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_s & \alpha_{s1} & \alpha_{s2} & \cdots & \alpha_{ss} \\ \hline & \beta_1 & \beta_2 & \cdots & \beta_s \end{array} . \tag{4.3}$$

Theorem 4.0.3. *A Radon method with any Runge-Kutta method as its one-step numerical solver is a GIP integrator.*

Proof. To prove this, we will use Theorem 3.3.2. In particular, we need to know how to cast Runge-Kutta methods into the framework of $\mathbf{Q} = \mathbf{P} + f(A)\mathbf{P}$ for the ODE $P' = AP$ as in Theorem 3.3.2 where $\mathbf{P} \approx P(t)$ and $\mathbf{Q} \approx P(t+h)$.

First, let $B_i = A(t + \gamma_i h)$. We have for $i = 1, 2, \ldots, s$

$$\mathbf{k}_i = B_i \left(\mathbf{P} + h \sum_{j=1}^{s} \alpha_{ij} \mathbf{k}_j \right) \quad \Rightarrow \quad \mathbf{k}_i - hB_i \sum_{j=1}^{s} \alpha_{ij} \mathbf{k}_j = B_i \mathbf{P}. \tag{4.4}$$

Together for all i, they lead to

$$\left[I - h \begin{pmatrix} \alpha_{11}B_1 & \alpha_{12}B_1 & \cdots & \alpha_{1s}B_1 \\ \alpha_{21}B_2 & \alpha_{22}B_2 & \cdots & \alpha_{2s}B_2 \\ \vdots & \vdots & & \vdots \\ \alpha_{s1}B_s & \alpha_{s2}B_s & \cdots & \alpha_{ss}B_s \end{pmatrix} \right] \begin{pmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_s \end{pmatrix} \mathbf{P}. \tag{4.5}$$

Denote the BIG coefficient matrix in brackets by C, and let $F = (B_1^T, B_2^T, \ldots, B_s^T)^T$. Then

$$\begin{pmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{pmatrix} = C^{-1} F P \tag{4.6}$$

and hence

$$\mathbf{Q} = \left[I + h(\beta_1 I, \beta_2 I, \ldots, \beta_s I) C^{-1} F \right] \mathbf{P}. \tag{4.7}$$

Therefore,

$$f(A) = (\beta_1 I, \beta_2 I, \ldots, \beta_s I) C^{-1} F. \tag{4.8}$$

Then setting $\tilde{C} = \text{diag}(K^T, K^T, \ldots, K^T)C\text{diag}(K, K, \ldots, K)$ and $\tilde{F} = \text{diag}(K^T, K^T, \ldots, K^T)FK$ yields:

$$f(K^T AK) = (\beta_1 I, \beta_2 I, \ldots, \beta_s I)\tilde{C}^{-1}\tilde{F} \tag{4.9}$$

$$= (\beta_1 K^T, \beta_2 K^T, \ldots, \beta_s K^T)C^{-1}FK \tag{4.10}$$

$$= K^T(\beta_1 I, \beta_2 I, \ldots, \beta_s I)C^{-1}FK \tag{4.11}$$

$$= K^T f(A)K. \tag{4.12}$$

The requirements of Theorem 3.3.2 are satisfied and the proof is finished. □

Therefore all the classic Runge-Kutta methods are GIP integrators. A few examples of these are the explicit and implicit Euler methods:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad , \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad ,$$

the implicit trapezoidal and midpoint rules:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \quad , \qquad \begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array} \quad ,$$

and the classical Runge-Kutta method of order 4:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & \\ 1/2 & 0 & 1/2 & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \quad .$$

4.1 Embedded Runge-Kutta methods

In this era of numerically approximating ODEs, it is common to use embedded Runge-Kutta methods. In fact, one of the most famous methods, *ode45* from the Matlab suite of ODE methods, is an embedded Runge-Kutta method. The idea behind such methods is to create 2 approximations to the ODE at each time step, one of which is higher order than the other. Then subtract the two approximations to estimate the local error.

To be more explicit, the ODE

$$x' = f(t, x) \tag{4.13}$$

will be approximated. An s-stage embedded Runge-Kutta method uses an extended Butcher array

$$\begin{array}{c|cccc} \gamma_1 & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1s} \\ \gamma_2 & \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_s & \alpha_{s1} & \alpha_{s2} & \cdots & \alpha_{ss} \\ \hline & \beta_1 & \beta_2 & \cdots & \beta_s \\ & \hat{\beta}_1 & \hat{\beta}_2 & \cdots & \hat{\beta}_s \end{array}$$

or more succinctly

$$\begin{array}{c|c} \gamma & A \\ \hline & \beta \\ & \hat{\beta} \end{array} .$$

Assume we have computed an approximation of the ODE (4.13), $x_k \approx x(t_k)$. To use the extended Butcher array, one does two approximations. The only difference between the approximations is the use of either β or $\hat{\beta}$. Both methods use the same matrix A and the same vector γ. The idea behind this is to minimize the number of

function evaluations of f. Another way to see this is that the k_i's from (4.1) for both methods are the same.

Using these two methods will give two approximations to $x(t_{k+1})$ which will be denoted x_{k+1} and $\hat{x}_{k+1}$. The first method using β has local truncation error $p+1$ and the second method using $\hat{\beta}$ has local truncation error p. Subtracting the two approximations then gives an approximation of the local error

$$\hat{x}_{k+1} - x_{k+1} = (\text{local error of } \hat{x}_{k+1}) + O(h^{p+1}). \tag{4.14}$$

Although we have a local error estimate for the lower order method, the higher order method is always taken as the approximation to the ODE in this paper as it is usually more accurate.

4.1.1 Step Size Control

We can use the local error approximation (4.14) for automatic step size control as detailed in [15, §II.4]. The local error will be approximated as above by

$$\text{err} = ||\hat{x}_{k+1} - x_{k+1}||_F \tag{4.15}$$

where $||\cdot||_F$ denotes the Frobenius norm. Then, we need a tolerance criteria for the error. The tolerance criteria used in the code for this paper is $\text{tol} = ||T||_F$ where

$$T = \text{ones}(\text{size}(x_{k+1}))\text{atol} + |x_{k+1}|\text{rtol}, \tag{4.16}$$

ones(size(...)) gives a matrix of 1's of the size of the input matrix, max and $|\cdot|$ are evaluated entry-wise for the given matrix, and atol/rtol are the absolute and relative tolerances.

Assuming h is small enough so that $\text{err} \approx Ch^{p+1}$, we want to find h_{new} such that when h_{new} is used in the next step, $\text{err} \approx \text{tol}$. A simple approximation yields

$$h_{\text{new}} \approx h(\text{tol}/\text{err})^{1/(p+1)}. \tag{4.17}$$

To add a little *wiggle room* to this since all of this relies on approximations, we multiply the right hand side by the factor 0.8. Also, to ensure we do not increase/decrease h by too large a factor, we also ensure h cannot change size by more than a factor of 2. Finally, this gives the time stepping control criteria used in all the codes used for this paper

$$h_{\text{new}} = h \min(2, \max(0.5, 0.8(\text{tol}/\text{err})^{1/(p+1)})). \tag{4.18}$$

4.1.2 Embedded Runge-Kutta Formulas

Two embedded Runge-Kutta formulas are used for the examples to be shown. The first is the Dormand and Prince 4(5) method which was created in the paper by the authors Dormand and Prince in [16] and may also be found in [12, 15] . The method uses embedded explicit Runge-Kutta formulas of orders 4 and 5. This method is used for two reasons.

- This Dormand and Prince 4(5) method minimizes the principal local error constant for the order 5 method. This helps to ensure that the local error estimate is an overestimate as opposed to an underestimate. The Fehlberg 4(5) method is also a widely used method, but it minimizes the principal local error constant for the order 4 method which can underestimate the local error and cause many time step rejections in the code.
- Matlab uses the Dormand and Prince 4(5) method for ode45 [17] which is one of the most used and hence well-tested and robust ODE solvers.

The Dormand and Prince 4(5) method is given by the following Butcher array.

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
\beta & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hat{\beta} & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40}
\end{array}
$$

(β is associated with the order 5 method and $\hat{\beta}$ is associated with the order 4 method.)

The other embedded Runge-Kutta method used is an embedded singly diagonally implicit Runge-Kutta (ESDIRK) method of order 4/5 developed by Kværnø [18]. An ESDIRK method has the property that A in the Butcher tableau is lower triangular with all the diagonal elements equal and nonzero. The purpose of having such a method is for stiff ODE problems. The ESDIRK method developed by Kværnø is not a true ESDIRK method actually as $\alpha_{11} = 0$. The Butcher array for this method is given on the next page.

0						
0.26000000000000000	0.26000000000000000					
0.13000000000000000	0.84033320996790809	0.26000000000000000				
0.22371961478320505	0.47675532319799699	−0.06470895363112615	0.26000000000000000			
0.16648564323248321	0.10450018841591720	0.03631482272098715	−0.13090704451073998	0.26000000000000000		
0.13855640231268224	0	−0.04245337201752043	0.02446657898003141	0.61943039072480676	0.26000000000000000	
0.13659751177640291	0	−0.05496908796538376	−0.04118626728321046	0.62993304899016403	0.06962479448202728	0.26
β	0	−0.05496908796538376	−0.04118626728321046	0.62993304899016403	0.06962479448202728	0.26
$\hat{\beta}$	0	−0.04245337201752043	0.02446657898003141	0.61943039072480676	0.26000000000000000	0

4.2 Global Error Estimation

Although very few ODE codes employ global error estimation, I believe it is essential. Hence the code uses a simple global error estimation as described by Shampine and Watts in [19]. The idea is very simple and will not be described at length here, but only a synopsis will be given.

Two integrations are done simultaneously giving two answers X_n and $\tilde{X}_n$. X_n is computed normally using the time step controls of the embedded Runge Kutta formulas. $\tilde{X}_n$ is computed by then taking two steps of half the size to get a more accurate global answer. Written in diagram form:

$$X_n \longrightarrow P_n \longrightarrow P_{n+1} \longrightarrow X_{n+1} \tag{4.19}$$

and

$$\tilde{X}_n \longrightarrow \tilde{P}_n \longrightarrow \tilde{P}_{n+1/2} \longrightarrow \tilde{P}_{n+1} \longrightarrow \tilde{X}_{n+1}. \tag{4.20}$$

The global error for X_n is then computed as

$$\text{global error} = X_n - \tilde{X}_n. \tag{4.21}$$

4.3 Computation of the Initial Timestep

During the numerical approximation of (MRDE), we have a current time step h except for when the algorithm begins. I have constructed a crude algorithm to construct an initial time step which will always give an accepted local error versus the tolerance. The algorithm is expensive as far as initial time step algorithms are concerned, but I wanted to ensure that rejected step sizes during the main algorithm would not be due to a bad initial value of h.

The initial step size is computed by first using a crude approximation for h as given by 'Phase 1' in the paper by Gladwell, Shampine, and Brankin [20]. Then, the

algorithm checks to see if this h gives a satisfactory local error. If not, h is halved until a satisfactory local error is obtained. The algorithm for this is given below.

```
Input Variables:
    A, t0, tf, P0, atol, rtol, solver

% Phase 1
tol = norm(atol * ones(size(P0)) + rtol * abs(P0), 'fro');
h = min(.1 * abs(tf - t0), tol^(1/5) / norm(A(t0) * P0, 'fro'));

% Find h which gives a sufficient local error.
while 1
    [Q, Q_hat] = solver_1step(f, t0, P0, h);
    tol = norm(atol * ones(size(Q)) + abs(Q) * rtol, 'fro');
    err = norm(Q - Q_hat, 'fro');
    if err < 0.8 * tol
        break;
    h = h / 2;
end
return h
```

4.4 The Algorithm

We now have all the technical details covered to give the algorithm to be used to solve (MRDE).

```
Input variables:
    rtol, atol, solver, use_qr, t0, x0, A(t), tlast
```

```
% Initialize output variables.
T = t0;
X(1) = x0;
X2(1) = x0;
accept = 0;
reject = 0;
P = [I; x0];
P2 = P;
h = .1;
i = 1;
facmax = 2;
facmin = .5;
fac = .8;
rejected_last_time = 0;
[m n] = size(P);

LOOP
    % This makes sure we don't solve for a point past tlast.
    if T + h > tlast
        h = tlast - T;
    end

    % Integrate one step of the ODE.
    [Q, Q_hat] = solver_1step(A(t), T, P, h);

    % Local Error and Tolerance
```

```
E = atol * ones(size(Q)) + abs(Q) * rtol;
nu = norm(Q - Q_hat, 'fro') / norm(E, 'fro');

% If error is acceptable, update all variables.
if nu <= 1
    X(i+1) = Q(n+1:m,:) * Q(1:n,:)^(-1);
    if(use_qr)
        P = skinny_qr(Q);
    else
        P = [I; X(i+1)];

    % More accurate answer.
    Q = solver_1step(A(t), T, P2, h/2);
    Q = solver_1step(A(t), T + h/2, Q, h/2);
    X2(i+1) = Q(n+1:m,:) * Q(1:n,:)^(-1);
    if(use_qr)
        P2 = skinny_qr(Q);
    else
        P2 = [I; X2(i+1)];

    T = T + h;
    i = i + 1;
    accept = accept + 1;

    % If we got the last point, stop the algorithm.
    if T >= tlast * (1 - eps)
```

```
            break LOOP;
    else
        reject = reject + 1;
        rejected_last_time = 2;

    % Change step size.
    if rejected_last_time > 0
        facmax = 1;
        rejected_last_time = rejected_last_time - 1;
    else
        facmax = 2;

    h = h * min(facmax, max(facmin, fac * nu^(-1/5)));
END LOOP

% Return data.
return X, X2, accept, reject
```

CHAPTER 5

Examples

In the upcoming examples there are a few options that can be set. In particular, for each example, one may use QR or inverse Radon methods and one may use the DOPRI or ESDIRK methods for the one-step numerical ODE solver. For all the examples, all 4 combinations of methods are shown to compare and contrast them except for Example 1. Also, no time step rejections occurred in the code, so only the number of time steps taken is shown. Finally, it should be noted that all errors given are relative errors.

5.1 Example 1

This is a 1-dimensional example due to Li and Kahan in [11]. It is

$$x' = t + x^2, \quad x(0) = 0. \tag{5.1}$$

The exact solution to this MRDE is

$$\sqrt{t}\frac{J_{2/3}(2t^{3/2}/3)}{J_{-1/3}(2t^{3/2}/3)} \tag{5.2}$$

where J_α is a Bessel function of the first kind. The following settings were used.

Relative Tolerance	10^{-6}
Absolute Tolerance	10^{-12}
t_0	0
t_{end}	10

Table 5.1. Example 1 parameters

The figures shown for this example are the approximated solution versus the true solution, the relative global error, and the ratio of the approximated relative global error versus the true relative global error. The number of accepted and rejected step sizes is shown for each method. Finally, since no difference was seen between using the QR and inverse Radon methods, only the inverse Radon method is shown here.

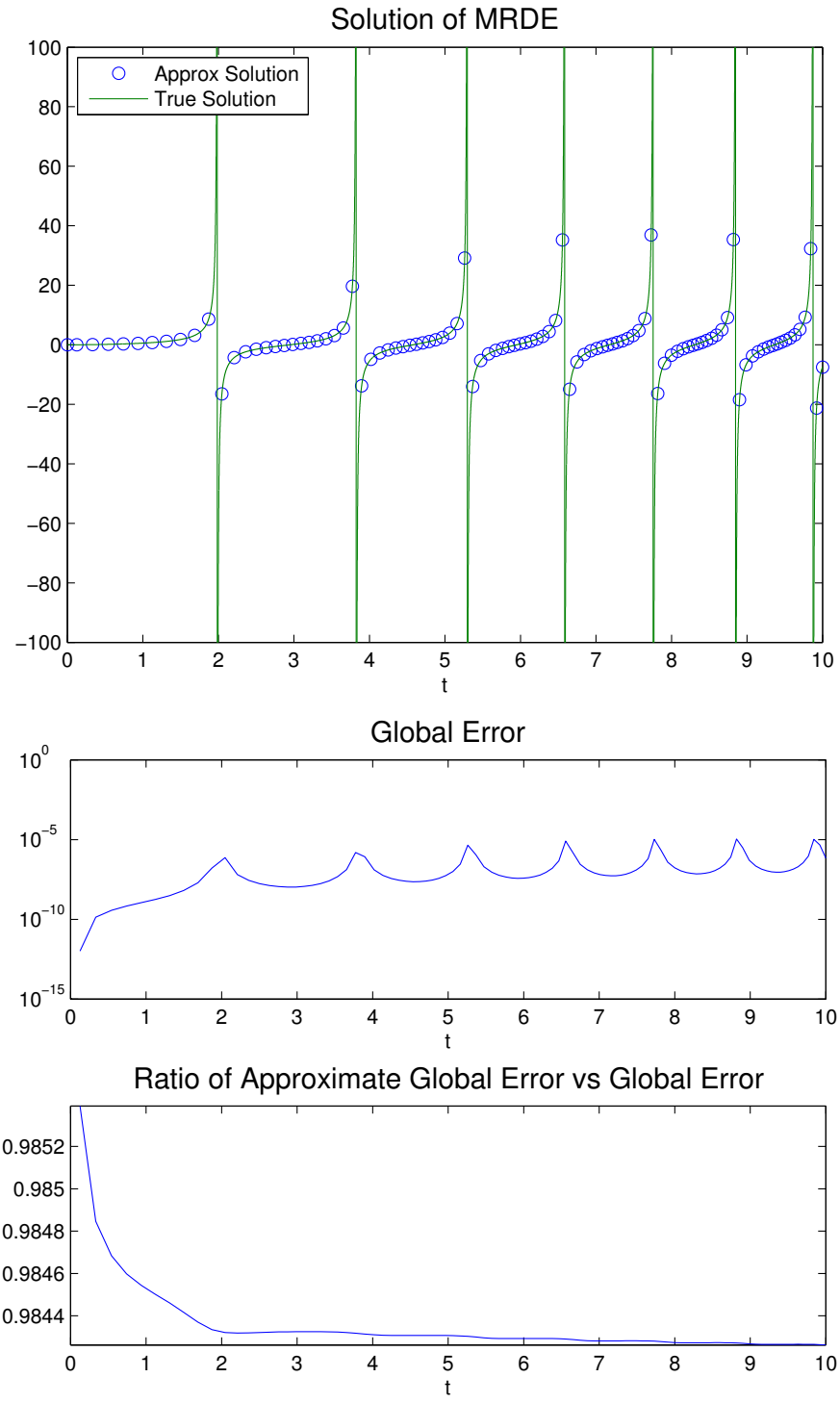

Figure 5.1. Example 1. Radon inverse method with DOPRI. Number of timesteps: 102.

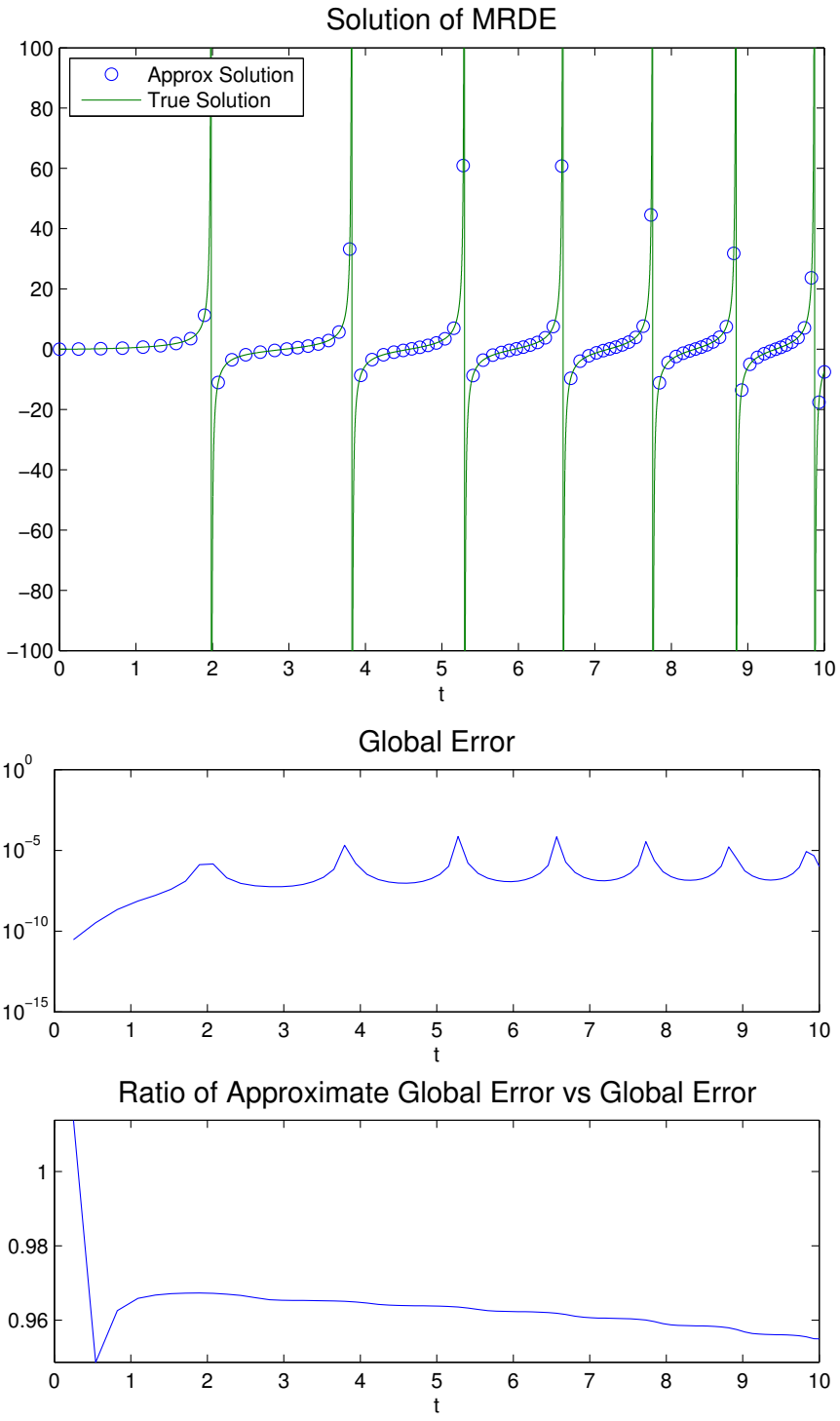

Figure 5.2. Example 1. Radon inverse method with ESDIRK. Number of timesteps: 82.

5.2 Example 2

This example comes from Choi and Laub in [21]. It is the MRDE

$$X' = k^2 I_n - X^2, \quad X(0) = X_0, \tag{5.3}$$

where k is a constant scalar and n may be any positive integer.

Choi and Laub show that if $M^2 = aI_n, a \neq 0$, then

$$e^{tM} = \cosh(\sqrt{a}t)I_n + \frac{1}{\sqrt{a}}\sinh(\sqrt{a}t)M. \tag{5.4}$$

This is used to show that the solution to (5.3) is

$$X(t) = (k\sinh(kt)I_n + \cosh(kt)X_0)(\cosh(kt)I_n + \frac{1}{k}\sinh(kt)X_0)^{-1}. \tag{5.5}$$

Therefore if X_0 is diagonalizable as $X_0 = S\Lambda S^{-1}$ where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, then

$$X(t) = S\mathrm{diag}\left\{\frac{k\sinh(kt) + \lambda_i\cosh(kt)}{\cosh(kt) + \frac{\lambda_i}{k}\sinh(kt)}, i = 1, \ldots, n\right\}S^{-1}. \tag{5.6}$$

For the results shown, $n = 2$, S is a random 2 by 2 matrix created by *rand(2)* in Matlab, $\Lambda = \mathrm{diag}(-2k, -3k)$, and $k = 10$. The other set parameters were as follows.

Relative Tolerance	10^{-6}
Absolute Tolerance	10^{-12}
t_0	0
t_{end}	2

Table 5.2. Example 2 parameters

The same data as shown for example 1 is shown for this example except both the QR and inverse Radon methods are shown. You may also notice that the errors seem erratic starting near $t = 1.2$. This occurs because the relative error is near the machine precision.

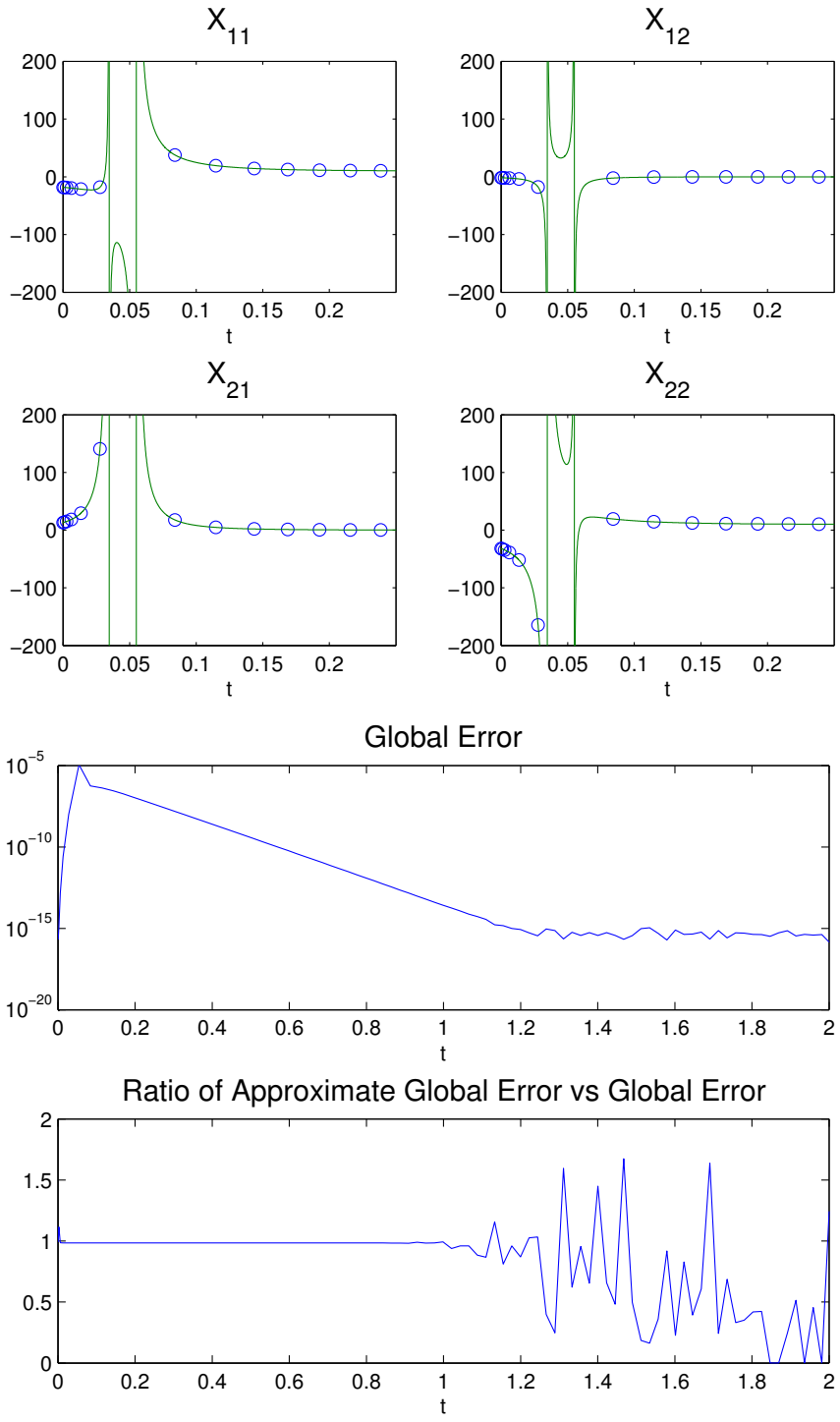

Figure 5.3. Example 2. Radon inverse method with DOPRI. Number of timesteps: 92.

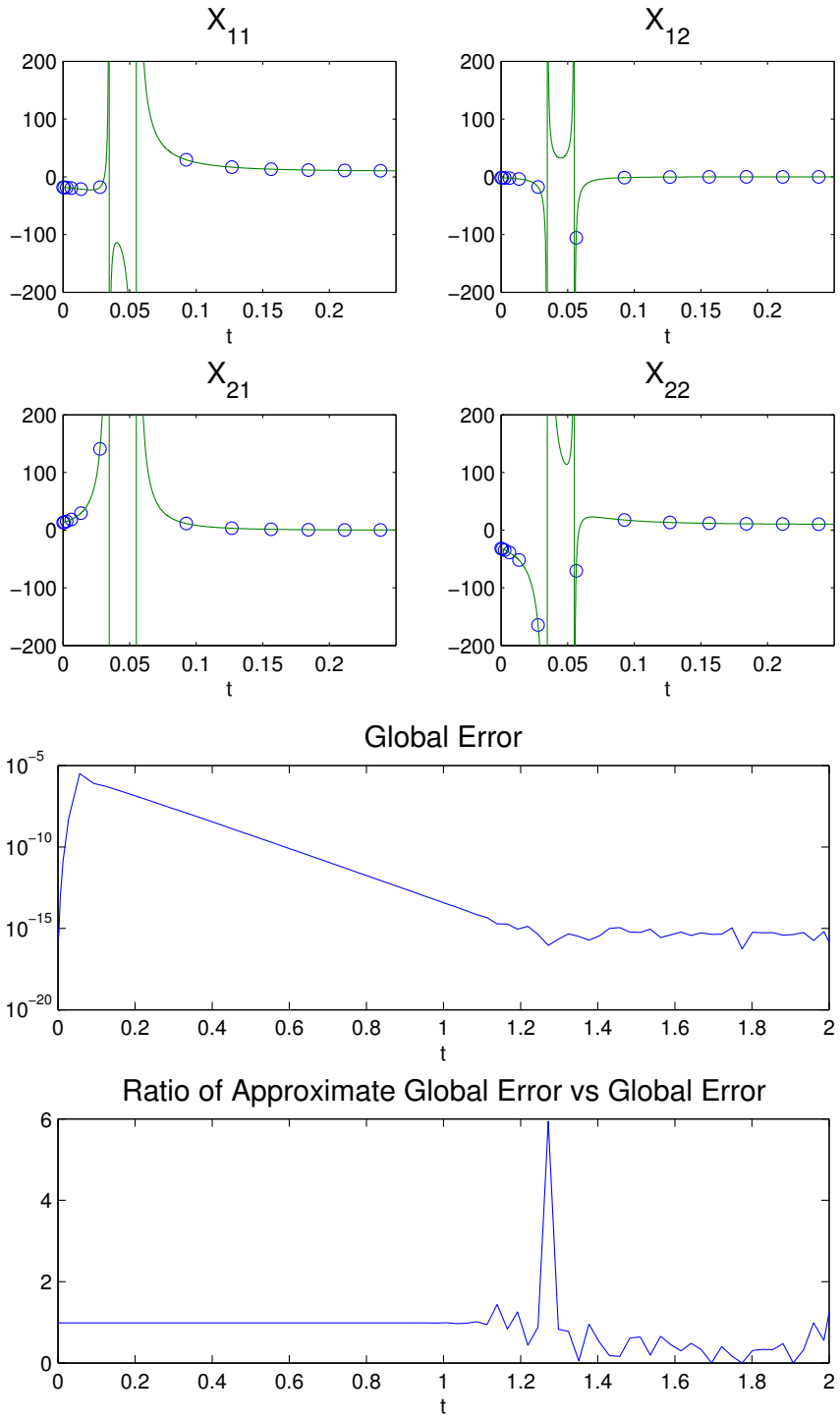

Figure 5.4. Example 2. Radon inverse method with ESDIRK. Number of timesteps: 79.

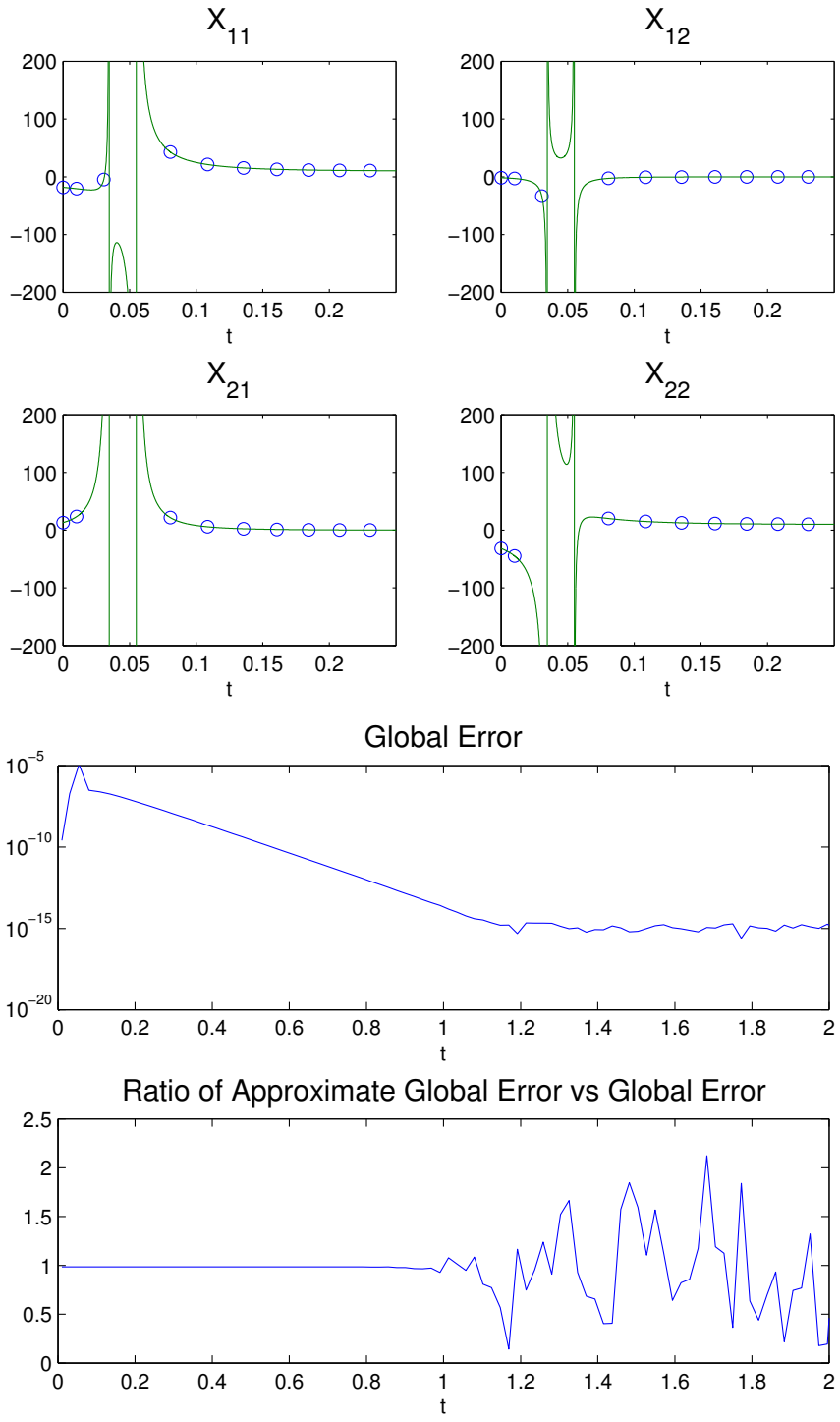

Figure 5.5. Example 2. Radon QR method with DOPRI. Number of timesteps: 90.

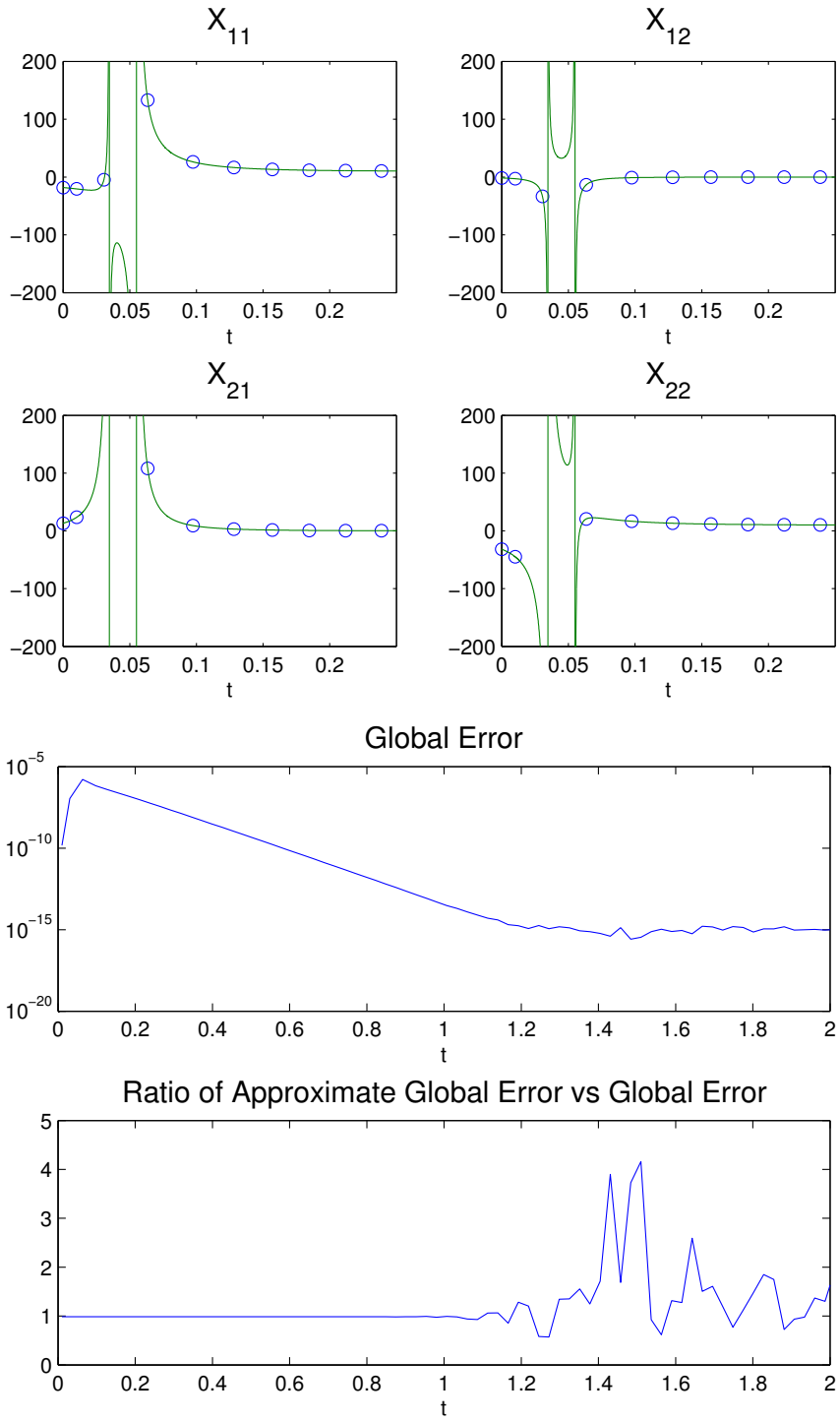

Figure 5.6. Example 2. Radon QR method with ESDIRK. Number of timesteps: 76.

5.3 Example 3

This example comes from the paper by Sorine and Winternitz in [22]. The example is for $X \in \mathbb{R}^{3\times 3}$, with (MRDE) defined by:

$$A_{11}(t) = -A_{22}^T(t) = \begin{pmatrix} .5 & -1 & 0 \\ 1 & .5 & -.5\cos(2t) \\ -.5\sin(2t) & -1 & 0 \end{pmatrix}, \tag{5.7}$$

$$A_{12}(t) = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1+.5\sin(2t) \end{pmatrix}, \tag{5.8}$$

$$A_{21}(t) = \begin{pmatrix} e^{-t/2} & 0 & 0 \\ 0 & e^{-t/2} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.9}$$

Sorine and Winternitz used two initial conditions:

$$X1(0) = \begin{pmatrix} -1 & 0.1 & 0.1 \\ 0.3 & -0.8 & 0.1 \\ 0.3 & 0.3 & -0.6 \end{pmatrix} \quad \text{and} \quad X2(0) = \begin{pmatrix} -1.01 & 0.1 & 0.1 \\ 0.3 & -0.81 & 0.1 \\ 0.3 & 0.3 & -0.61 \end{pmatrix} \tag{5.10}$$

Even though the difference between $X1(0)$ and $X2(0)$ is small, the solution of (MRDE) has no singularities for initial condition $X1(0)$, but contains a singularity for $t \approx 0.8$ for initial condition $X2(0)$ as shown in Figure 5.7. Since we are mainly concerned with MRDEs containing singularities, only graphs of data for $X2$ are shown afterwards. Since the true solution is not known, only the approximated global error is given in the figures. Also, step sizes in t are given for this example. The following parameters were used for the graphs.

Relative Tolerance	10^{-8}
Absolute Tolerance	10^{-16}
t_0	0
t_{end}	2

Table 5.3. Example 3 parameters

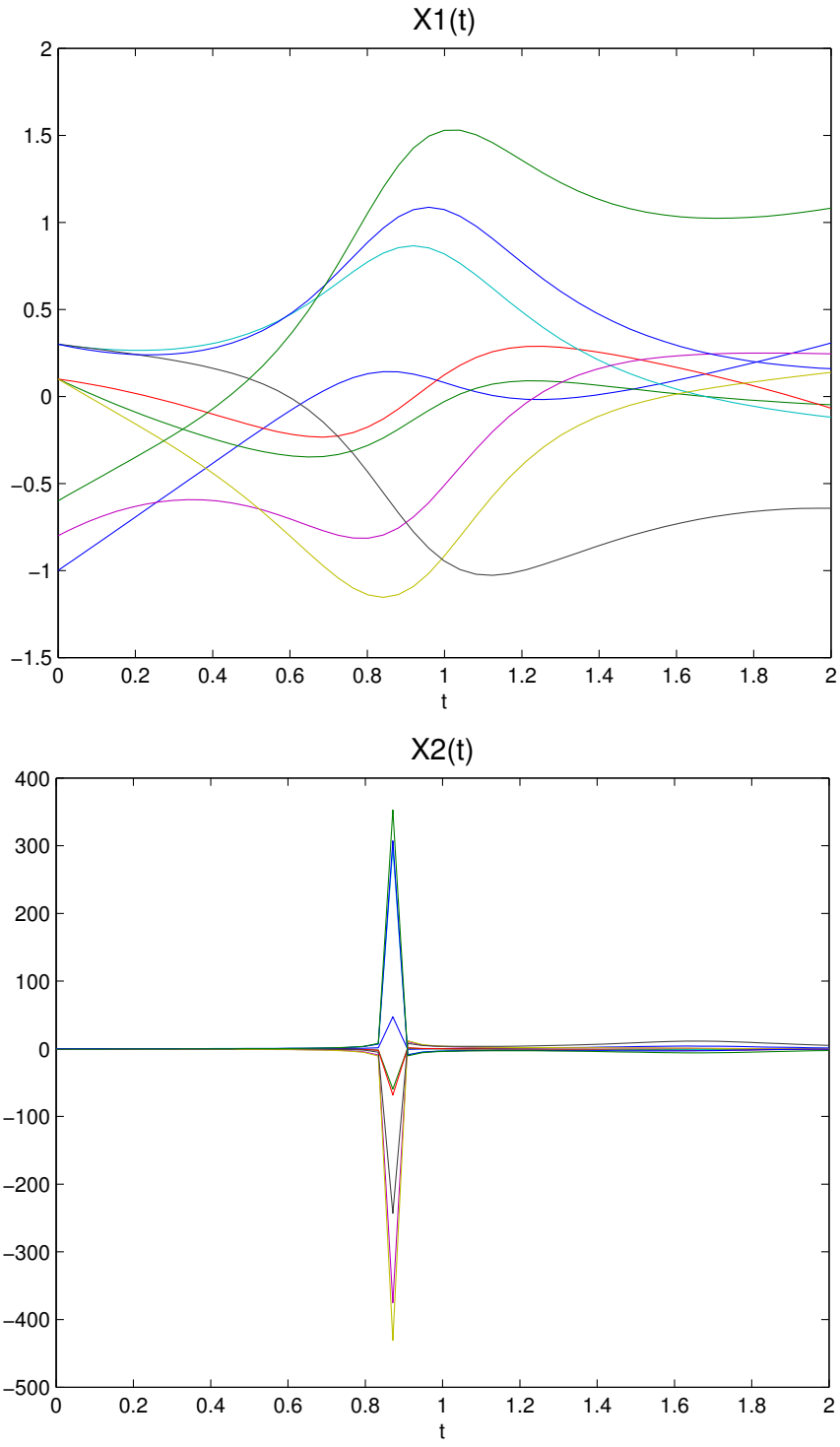

Figure 5.7. Example 3. Solution of (MRDE) using initial conditions $X1(0)$ (top) and $X2(0)$ bottom.

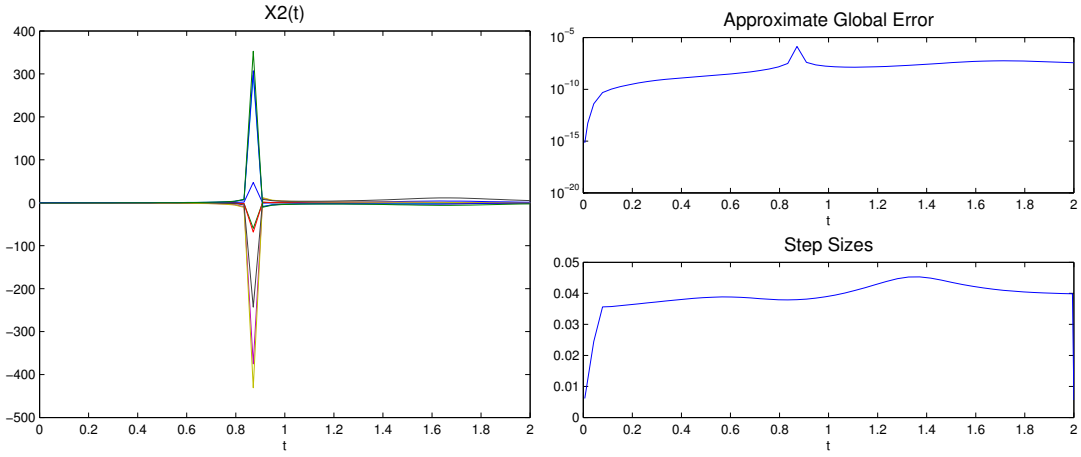

Figure 5.8. Example 3. Radon inverse method with DOPRI for X2(t). Number of timesteps: 53.

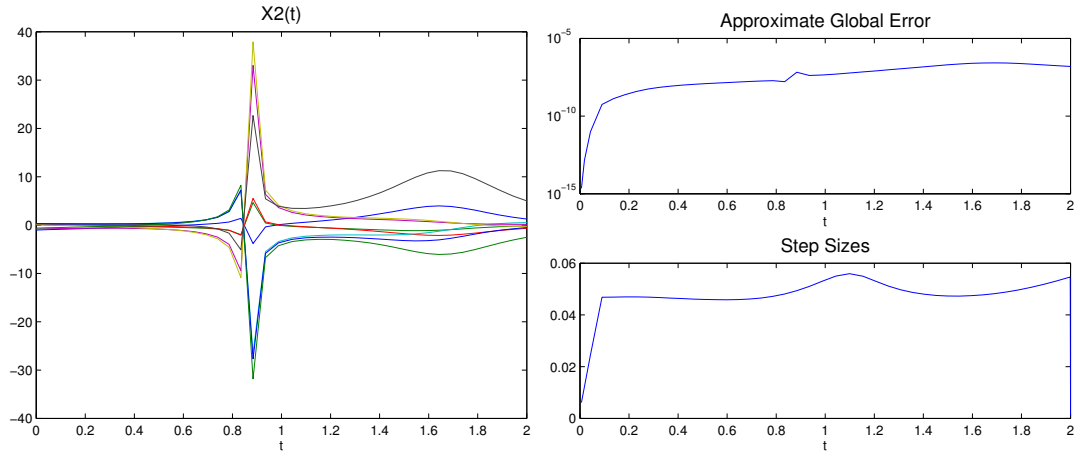

Figure 5.9. Example 3. Radon inverse method with ESDIRK for X2(t). Number of timesteps: 44.

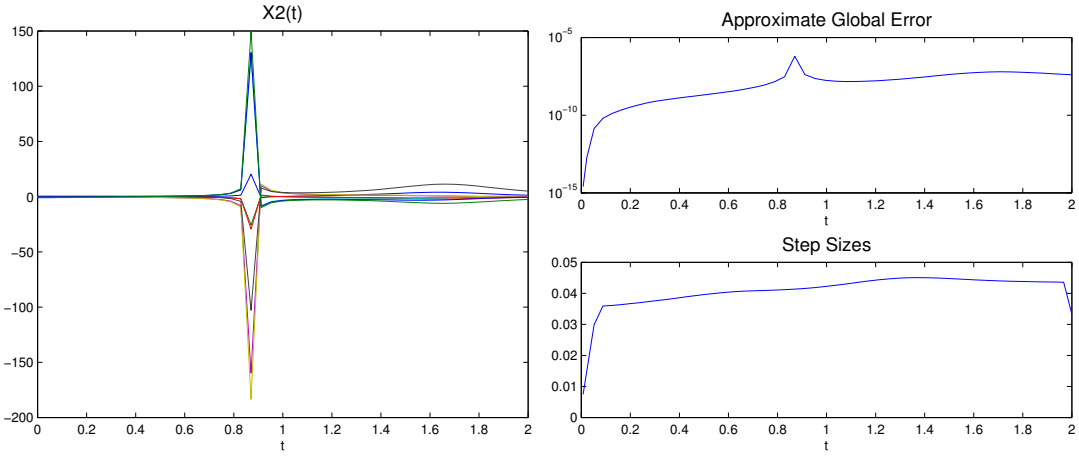

Figure 5.10. Example 3. Radon QR method with DOPRI for X2(t). Number of timesteps: 50.

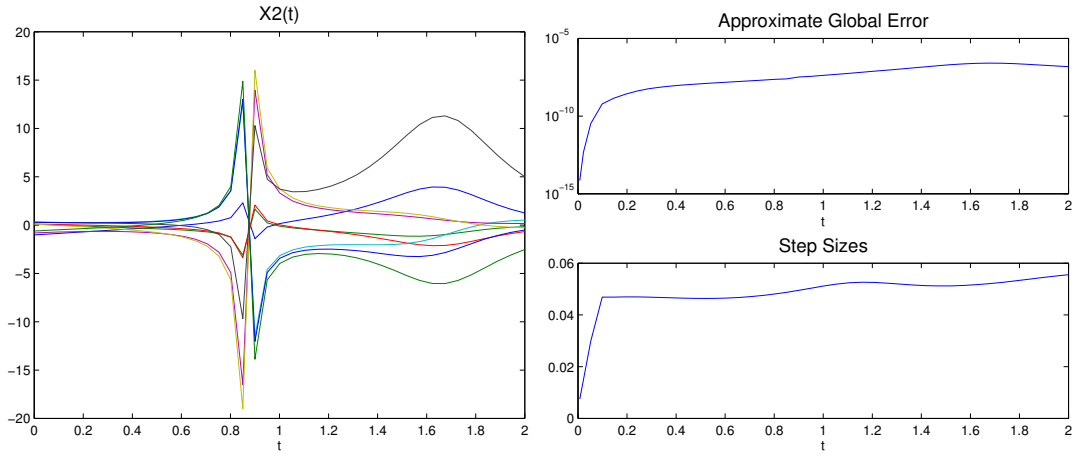

Figure 5.11. Example 3. Radon QR method with ESDIRK for X2(t). Number of timesteps: 42.

5.4 Example 4

This example is the same as example 3, except a nonsquare initial condition was propose by Ren-Cang Li, which still gives singularities in the solution to (MRDE).

The purpose of this example is to show that the MRDE need not be square for the algorithms to work.

The initial condition is:

$$X(0) = \begin{pmatrix} -1.0 & 0.1 & 0.1 & 0.0 \\ 0.3 & -0.8 & 0.1 & 0.0 \end{pmatrix}. \tag{5.11}$$

All parameters were kept the same as in Example 3.

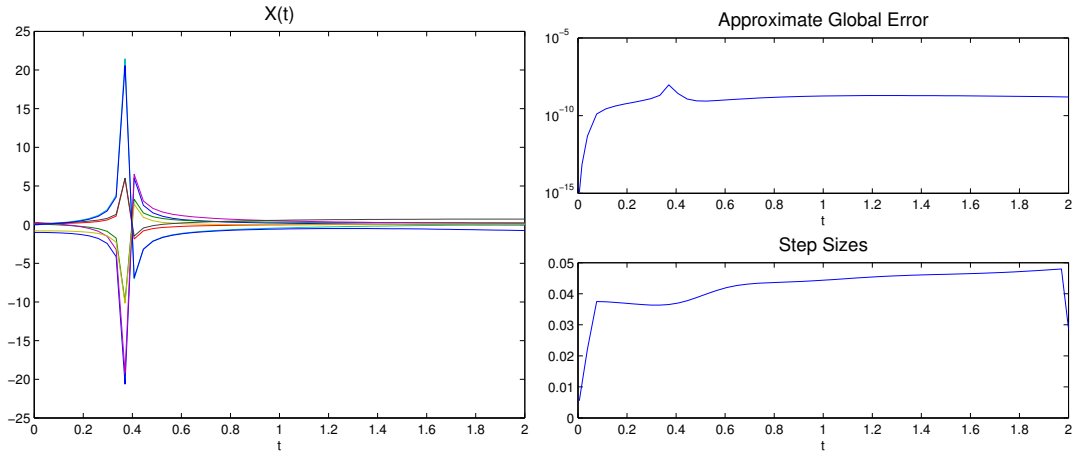

Figure 5.12. Example 4. Radon inverse method with DOPRI. Number of timesteps: 49.

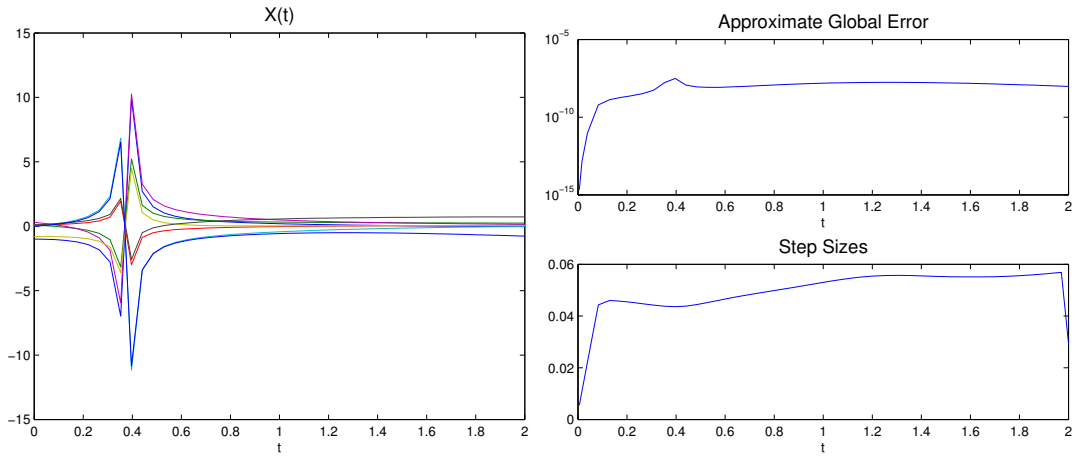

Figure 5.13. Example 4. Radon inverse method with ESDIRK. Number of timesteps: 42.

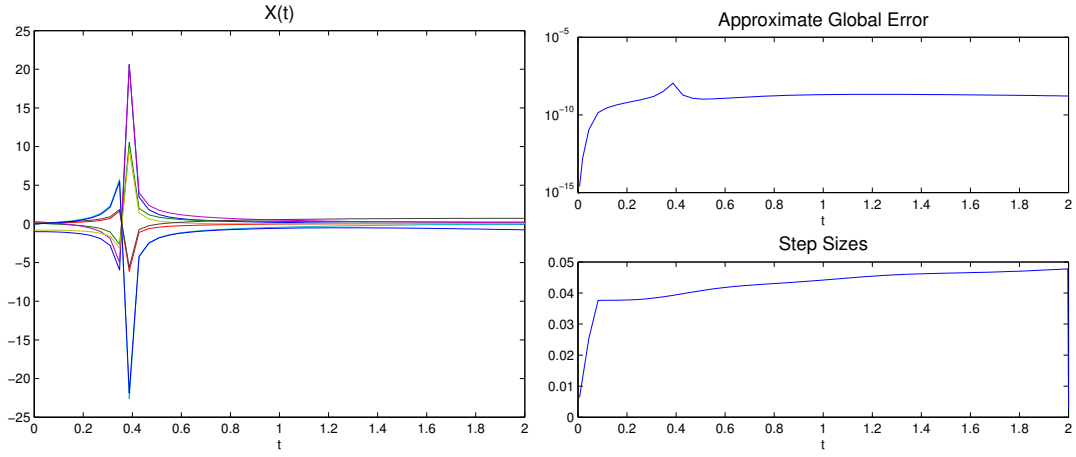

Figure 5.14. Example 4. Radon QR method with DOPRI. Number of timesteps: 49.

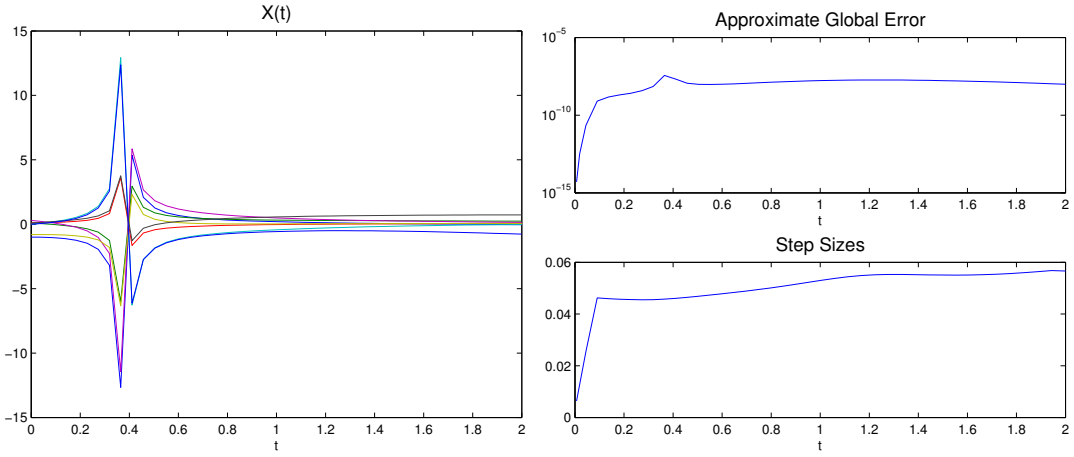

Figure 5.15. Example 4. Radon QR method with ESDIRK. Number of timesteps: 41.

5.5 Example 5

This example was found in the paper by Dieci [23] where numerous MRDE examples were tested using a different algorithm. As Dieci was not attempting to solve MRDEs with singularities in the solution, this example does not have a solution singularity. Rather this example is a stiff ODE problem to highlight the use of ESDIRK over DOPRI for stiff problems. The MRDE is defined by:

$$A(t) = \begin{pmatrix} \frac{-t}{2\epsilon} & 0 & \frac{1}{\epsilon} & 0 \\ 0 & 0 & 0 & \frac{1}{\epsilon} \\ \frac{1}{2} & 1 & 0 & \frac{t}{2\epsilon} \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{5.12}$$

where $0 < \epsilon \ll 1$ and $X(-1) = 0$.

The parameters used for the solver were:

Relative Tolerance	10^{-4}
Absolute Tolerance	10^{-8}
ϵ	0.001
t_0	-1
t_{end}	5

Table 5.4. Example 5 parameters

Notice that after $t = 0$, the ESDIRK method is able to use much larger step sizes than the DOPRI method because of the stiffness of the problem.

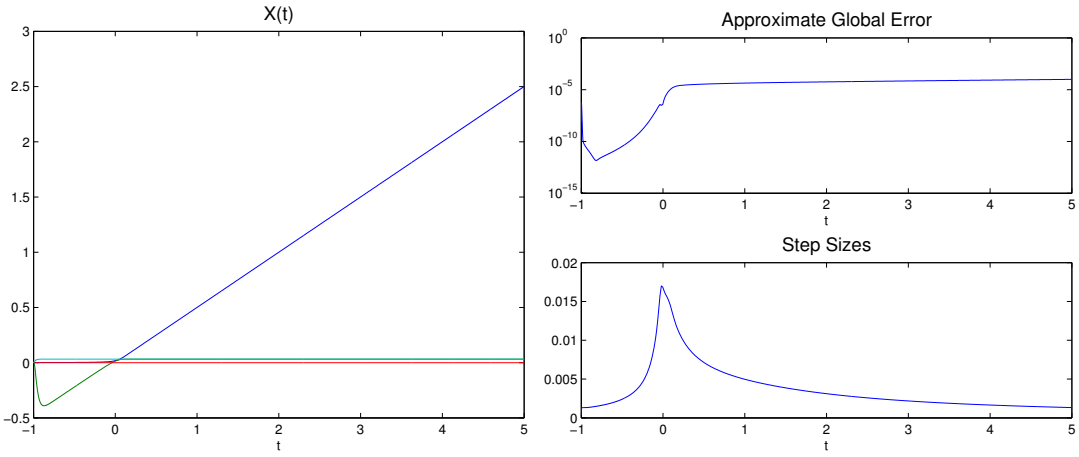

Figure 5.16. Example 5. Radon inverse method with DOPRI. Number of timesteps: 2412.

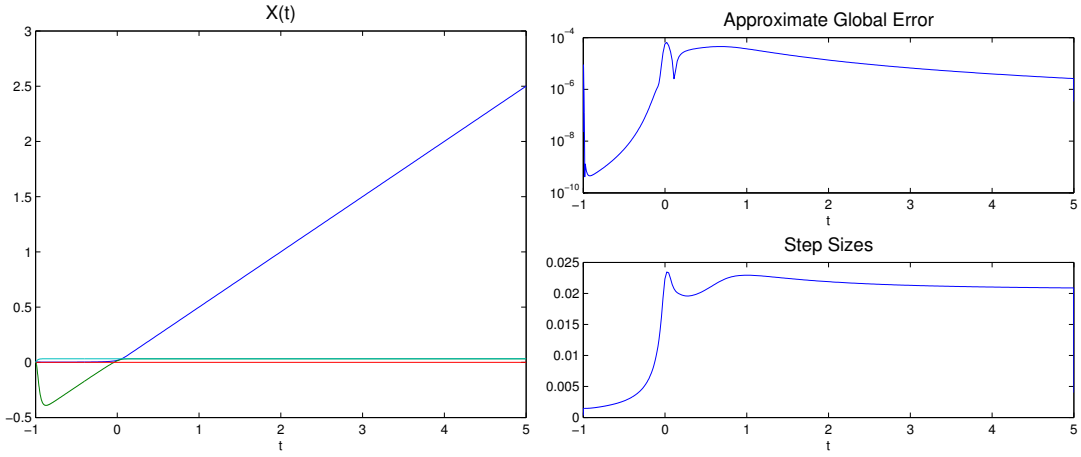

Figure 5.17. Example 5. Radon inverse method with ESDIRK. Number of timesteps: 607.

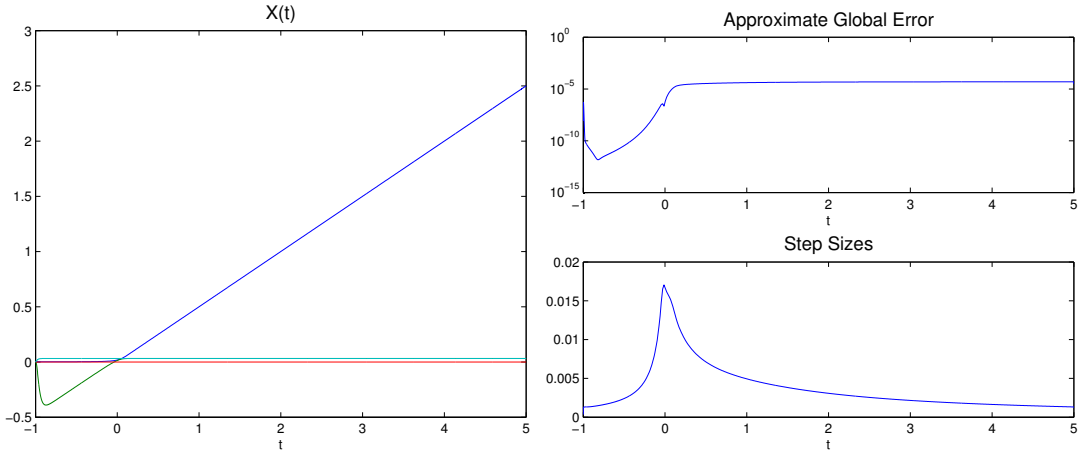

Figure 5.18. Example 5. Radon QR method with DOPRI. Number of timesteps: 2425.

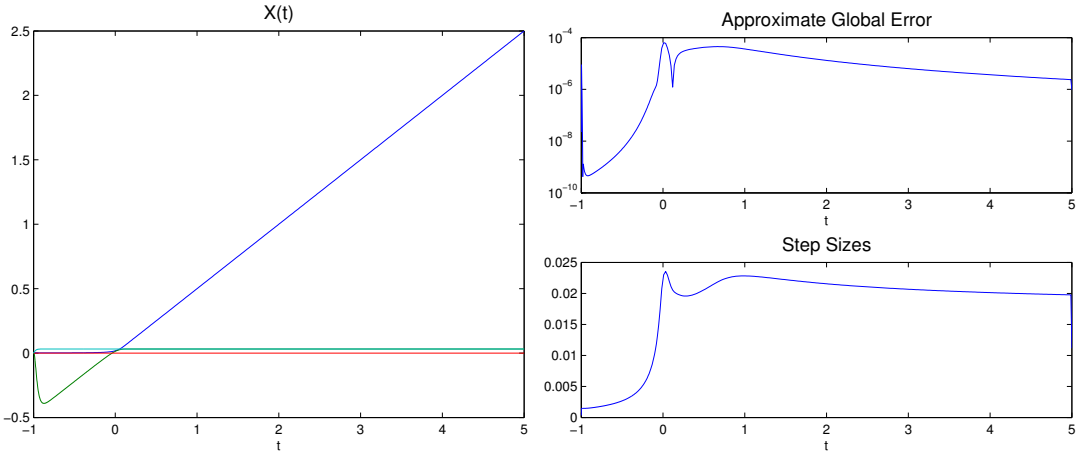

Figure 5.19. Example 5. Radon QR method with ESDIRK. Number of timesteps: 612.

REFERENCES

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory.* Berlin: Birkhäuser Verlag, 2003.

[2] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE Transactions on Automatic Control*, vol. AC-16, p. 6, 1971.

[3] F. L. Lewis, *Optimal Control.* John Wiley & Sons, 1986.

[4] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[5] L. Dieci, M. R. Osborne, and R. D. Russell, "A Riccati transformation method for solving linear BVPs. I: Theoretical aspects," *SIAM J. Numer. Anal.*, vol. 25, no. 5, pp. 1055–1073, 1988.

[6] ——, "A Riccati transformation method for solving linear BVPs. II: Computational aspects," *SIAM J. Numer. Anal.*, vol. 25, no. 5, pp. 1074–1092, 1988.

[7] J. Fonseca, M. Grasselli, and C. Tebaldi, "Option pricing when correlations are stochastic: an analytical framework," *Review of Derivatives Research*, vol. 10, pp. 151–180, 2007, 10.1007/s11147-008-9018-x. [Online]. Available: http://dx.doi.org/10.1007/s11147-008-9018-x

[8] W. T. Reid, *Riccati Differential Equations.* New York: Academic Press, 1972.

[9] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations, Reprint.* Malabar, Florida: Krieger Publishing Company, 1984.

[10] J. Schiff and S. Shnider, "A natural approach to the numerical integration of Riccati differential equations," *SIAM J. Numer. Anal.*, vol. 36, no. 5, pp. 1392–1413, 1999.

[11] R.-C. Li and W. Kahan, "A family of anadromic numerical methods for matrix Riccati differential equations," *Math. Comp.*, vol. 81, no. 277, pp. 233–265, January 2012.

[12] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems.* New York: John Wiley & Sons, 1991.

[13] J. C. Butcher, *Numerical Methods For Ordinary Differential Equations*, 2nd ed. West Sussex, England: Wiley, 2008.

[14] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., ser. Springer Series in Computational Mathematics. Berlin: Springer, 2006, no. 31.

[15] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, 2nd ed. New York: Springer-Verlag, 1993.

[16] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19 – 26, 1980.

[17] L. F. Shampine and M. W. Reichel, "The MATLAB ODE suite," *SIAM J. Sci. Comput.*, vol. 18, pp. 1–22, January 1997.

[18] A. Kværnø, "Singly diagonally implicit runge-kutta methods with an explicit first stage," *BIT*, vol. 44, no. 1, pp. 489–502, 2004.

[19] L. F. Shampine and H. A. Watts, "Global error estimates for ordinary differential equations," *ACM Transactions on Mathematical Software*, vol. 2, no. 2, pp. 172–186, June 1976. [Online]. Available: http://doi.acm.org/10.1145/355681.355687

[20] I. Gladwell, L. Shampine, and R. Brankin, "Automatic selection of the initial step size for an ode solver," *Journal of Computational and Applied Mathematics*, vol. 18, no. 2, pp. 175 – 192, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037704278790015X

[21] C. H. Choi and A. J. Laub, "Constructing Riccati differential equations with known analytic solutions for numerical experiments," *IEEE Trans. Automat. Control*, vol. 35, pp. 437–439, 1990.

[22] M. Sorine and P. Winternitz, "Superposition laws for solutions of differential matrix Riccati equations arising in control theory," *IEEE Trans. Automat. Control*, vol. AC-30, pp. 266–272, 1985.

[23] L. Dieci, "Numerical integration of the differential Riccati equation and some related issues," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 781–815, 1992.

BIOGRAPHICAL STATEMENT

Charles K Garrett has a biography.