SHORT-MT: Optimal Solution of Linear Ordinary Differential Equations by Conjugate Gradient Method

Wenqiang Yang1, Wenyuan Wu1**, and Robert M. Corless2

1 Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China
2 Department of Applied Mathematics, Western University, Canada

Abstract. Solving initial value problems and boundary value problems of Linear Ordinary Differential Equations (ODEs) plays an important role in many applications. There are various numerical methods and solvers to obtain approximate solutions represented by points. However, few work about optimal solution to minimize the residual can be found in the literatures. In this paper, we first use Hermit cubic spline interpolation at mesh points to represent the solution, then we define the residual error as the $L^2$ norm of the residual obtained by substituting the interpolation solution back to ODEs. Thus, solving ODEs is reduced to an optimization problem in certain solution space which can be solved by conjugate gradient method with taking advantages of sparsity of the corresponding matrix. The examples of IVP and BVP in the paper show that this method can find a solution with smaller global error without additional mesh points.

Keywords: ODEs · global error · Hermit cubic spline · optimization problem · conjugate gradient method.

1 Introduction

Differential equations (DEs) are one of the most fundamental tools in physical world to model the dynamics of a system. In machine learning, we may model the learner as some dynamical system, for example a neural network with weights changing according to certain rules. For example, the continuous time recurrent neural network using a system of ODE to simulate the effects on a neuron of the incoming spike train works successfully in evolutionary robotics. Spiking neural networks as third generation of neural network increase the level of realism in a neural simulation where the neural voltage is usually described as DEs.

Interestingly, people also exploit reversely a variety of neural network methods for solving DEs arising in science and engineering [13]. In this paper, starting

* Supported by NSFC 11471307 and CAS Research Program of Frontier Sciences (QYZDB-SSW-SYS026).
** Corresponding author: wuwenyuan@cigit.ac.cn.
with the most fundamental case: linear ODE, we attempt to study the optimal solution first by symbolic transformation to an optimization problem and then attain the solution by numerical methods.

The standard form of a Linear ODEs is

\[ \mathbf{x}' = A(t) \cdot \mathbf{x}(t) + q(t) \]  

(1)

Where \( \mathbf{x} = \mathbf{x}(t) : \mathbb{R} \to \mathbb{R}^n \) is vector of solutions as a function time, the \( n \times n \) matrix \( A(t) \) is the state matrix, the \( n \) dimensional vector \( q(t) \) corresponds to the inhomogeneous part of the system.

Such ODE often appearing in numerical modelling and simulation is of great importance in mechanical engineering and industrial design. Numerical solving of linear ODE is well studied, especially IVP[12] and BVP[1][10][2] for ODE. Even a number of efficient solvers have been developed e.g. Matlab. To qualify and measure the reliability of these solvers without knowing the exact solution is an important question[13]. A natural choice is to consider the residual error of the equation after substituting the approximate solution[9][3][6].

Accuracy of the solution depends on the used discrete mesh. If we fix the mesh and consider the solution with first-order continuity approximated by degree 3 piecewise polynomials. The key question is to look for a solution with smallest residual error in such solution space which is different from the goal of [3] to search optimal interpolant with a given numerical solution of ODE. Our main contribution of this paper is to convert the IVP/BVP of ODEs to a quadratic programming problem. Moreover, the optimal solution can be obtained efficiently by the conjugate gradient method since the corresponding matrix is symmetric and positive definite.

2 Preliminaries

2.1 Hermite Cubic Spline

For first-order continuity, we apply Hermite cubic spline interpolation to construct an approximate solution of ODEs [7].

**Theorem 1.** Let \( \{t_0, ..., t_m\} \) be a set of knots in the interval \([t_0, t_m]\) with \( t_0 < t_1 < ... < t_m \) and \( m \geq 2 \), there is a unique cubic spline \( \bar{x} \in C^1[t_0, t_m] \) on the interval \([t_i, t_{i+1}]\), for \( i = 0, ..., m - 1 \) such that

\[ \bar{x}(t_i) = x_i, \quad \bar{x}'(t_i) = x_i' \]

Let \( h_i = t_{i+1} - t_i \), and \( \tau = (t - t_i)/h_i, \quad \tau \in [0, 1] \), then writing the spline \( \bar{x} \) as

\[ \bar{x}_i(\tau) = \alpha_0(\tau) \cdot x_{i-1} + h_i \alpha_1(\tau) \cdot x'_{i-1} + \beta_0(\tau) \cdot x_i + h_i \beta_1(\tau) \cdot x'_i \]

Where \( \alpha_0(\tau) = 2\tau^3 - 3\tau^2 + 1, \quad \alpha_1(\tau) = \tau^3 - 2\tau^2 + \tau, \quad \beta_0(\tau) = -2\tau^3 + 3\tau^2, \quad \beta_1(\tau) = \tau^3 - \tau^2. \)
2.2 Residual and Forward Error

To evaluate the accuracy of this solution, here we first define the residual and forward error which is introduced in [13]. More detailed study about the error analysis can be found in the book [3].

**Definition 1.** Let \( n \)-dimensional vector \( \tilde{x}(t) \in C^1[t_0, t_m] \) be approximate solution of Eq. (1). Then its **residual** is defined to be

\[
\delta(t) = \tilde{x}'(t) - A(t) \cdot \tilde{x}(t) - q(t)
\]  

(2)

The residual of \( C^1 \) solution is also called **defect** in [3].

The \( L_2 \) norm of the residual

\[
\sqrt{\int_{t_0}^{t_m} \delta(t)^T \delta(t) dt}
\]

is called the **residual error**.

Let \( n \)-dimensional vector \( x^*(t) \) be the exact solutions of linear ordinary system Eq. (7), such that

\[
(x^*)' = A(t) \cdot x^* + q(t)
\]

(3)

The difference \( \Delta x(t) = x^* - \tilde{x} \) is called the **forward error**.

In addition, we define the \( L_2 \) norm of the forward error

\[
\sqrt{\int_{t_0}^{t_m} \Delta x^T \Delta x \, dt}
\]

as the **global error**.

3 Optimal Solution

In order to minimize the residual error of ODEs solution on the whole time interval, we can define an objective function as follow:

\[
\min \left[ \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} \delta_i(t)^T \delta_i(t) \, dt \right]
\]

(4)

where \( \delta_i(t) \) is the residual on the \( i \)-th time interval. Recall Theorem 1, we can express each \( \delta_i(t) \) in terms of \( x_i, x'_i, x_{i+1}, x'_{i+1} \) at the mesh points.

We denote the matrices \( a_i = \frac{\alpha_i}{h} E - \alpha_0 A, b_i = \alpha'_i E - h_i \alpha_1 A, c_i = \frac{\beta_i}{h} E - \beta_0 A \) and \( d_i = \beta'_i E - h_i \beta_1 A \), where \( E \) is the \( n \times n \) identity matrix. Thus,

\[
\delta_i = a_i x_i + b_i x'_i + c_i x_{i+1} + d_i x'_{i+1} - q.
\]
Additionally, let $I_i = \int_{t_i}^{t_{i+1}} \delta_i(t)^T \delta_i(t) \, dt$. Then we have

$$I_i = y_i^T F_i y_i - 2B_i^T y_i + \int_{t_i}^{t_{i+1}} q^T q \, dt$$

(5)

where $F_i = \int_{t_i}^{t_{i+1}} \begin{pmatrix} a_i^T a_i & a_i^T b_i & a_i^T c_i & a_i^T d_i \\ b_i^T a_i & b_i^T b_i & b_i^T c_i & b_i^T d_i \\ c_i^T a_i & c_i^T b_i & c_i^T c_i & c_i^T d_i \\ d_i^T a_i & d_i^T b_i & d_i^T c_i & d_i^T d_i \end{pmatrix} \, dt$ is a $4n \times 4n$ matrix, and

$$B_i = \int_{t_i}^{t_{i+1}} q^T \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix} \, dt \quad \text{and} \quad y_i = \begin{pmatrix} x_i \\ x_i' \\ x_{i+1} \\ x_{i+1}' \end{pmatrix}$$

are $4n$-dimensional vectors.

Here $x_i$, $x_i'$, $x_{i+1}$, $x_{i+1}'$ are unknowns in the optimization problem Eq.(4).

Since $\sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} q^T q \, dt \geq 0$ is a constant, to minimize $I_i$ is equivalent to minimize $y_i^T F_i y_i - 2q_i^T B_i y_i$. It’s obviously that objective function can be rewritten as a quadratic form

$$\text{opt} = \min y_i^T F y_i - 2B_i^T y_i.$$ 

(6)

where $F$ is highly structured with size $2n(m+1) \times 2n(m+1)$ (see Fig.1).

It is not difficult to verify that matrix $F$ is a symmetric positive definite matrix. According to [1], the minimal value can be attained at the solution of

$$F y = B$$

(7)
It is well-known that such linear system can be solved efficiently by using conju-
gate gradient method. Especially, if we have obtained an approximate numerical
solution by other solvers, we can use it as an initial point of conjugate gradient
method to refine the solution.

It is important to point out here that the same formulation Eq.(6) works
for both initial value problems (IVPs) and boundary value problems (BVPs) of
Linear ODEs. The only difference is that the first \( n \) rows and columns of \( F \) will
be removed after substituting the initial values for IVPs. But for BVPs, there are
\( n \) rows in different places and the corresponding columns which will be removed.
Accordingly, the substitution also happens in \( B \). Thus, the number of unknowns
of Eq.(6) drops by \( n \).

4 Examples

In order to show the performance of our method to IVPs and BVPs for linear
ODEs, we give an RC ladder network system as an example, which can be
used to filter a signal by blocking certain frequencies and passing others. When
we consider its time response, this system can be considered as an IVP. In
addition, when we consider its controllability, this system becomes a BVP. We
use a constant state matrix \( A \) since it is easy to obtain the exact solution for
comparison.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{rc_ladder_network.png}
\caption{An RC Ladder Network}
\end{figure}

4.1 An Initial Value Problem

An RC ladder network is shown in Fig. 2. Assume that the state variables are
\( x = [x_1, x_2]^T \) be the voltage across each capacitor. During the time period \( t \in [0, 2] \), we study the time response function of the system. Suppose the resistances
\( R_1 = R_2 = 10k\Omega \), and the capacitance \( C_1 = C_2 = 100\mu F \). At the beginning,
\( x(0) = [0, 0]^T \), we give an input signal \( u(t) = 2 \cdot \sin(2t) \). So the model can be
described as follows:

\[
x' = \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t)
\]
Obviously, this ODE has an exact solution as $x^* = e^{At} \cdot \int_0^t e^{-As} q(s)ds$. We apply ode45 and our method, which is implemented also in Matlab, to the same 13 mesh-points decided by ode45, and get the approximate solutions respectively. The corresponding $F$ is a $50 \times 50$ matrix and its condition number is $3.949 \times 10^3$. The residual of both methods are almost the same in Fig.3 and residual error of our method $0.4097 \times 10^{-3}$ is slightly less than $0.4275 \times 10^{-3}$ by ode45. However, Fig.4 shows that global error by our method (blue curve) $9.1313 \times 10^{-5}$ is much smaller than the one by ode45 (red dash curve) $1.8232 \times 10^{-4}$. It’s clear that the forward errors of both methods are fluctuating around zero but the blue curve given by our method has a smaller amplitude than ode45. The reason behind would be our method is a kind of global approach compared with the local approach ode45.

![Residual comparison for IVP between ODE45 (red dash) and our method (blue).](image)

**4.2 A Boundary Value Problem**

If we want to control the former RC ladder network to achieve the goal $x_1(0) = 0$ and $x_2(2) = 1$, and input signal $u(t) = 2 \cdot sin(2t)$, how about the beginning voltage of $x_2(0)$? To answer the question, we need to solve bvp for ODEs.

In this example, We apply Matlab’s bvp5c and our method to the same 10 mesh-points decided by bvp5c. In our method, $F$ is a $38 \times 38$ matrix with the condition number $1.640 \times 10^3$. In order to compare the forward error, we also compute its exact solution by Sec 6.1 [1]. The residual of both methods are almost the same in Fig.5 and residual error of our method $0.7942 \times 10^{-3}$ is less than $0.9595 \times 10^{-3}$ given by bvp5c. Similar to the IVP case, the same comparison result can be observed in Fig.6 that global error by our method (blue curve) $1.7824 \times 10^{-4}$ is much smaller than the one by bvp5c (red dash curve) $4.8521 \times 10^{-4}$.
Fig. 4. Error comparison for IVP between ODE45 (red dash) and our method (blue).

Fig. 5. Residual comparison for IVP between ODE45 (red dash) and our method (blue).
5 Conclusion

In this paper, we provide a conjugate gradient based method for computing optimal solution of linear ordinary differential equations. A fine mesh in time interval leads to a large but sparse matrix $F$. Due to nice properties of $F$ the iteration approach for linear system can be applied here and it can take advantages of sparsity. This method can solve both IVPs and BVPs of Linear ODEs.

As an illustrative example, we use RC ladder network to generate IVP and BVP of Linear ODEs, whose exact solution can be found easily. By comparison, we find that our method can give solutions with smaller forward error and global error than solvers provided by Matlab at the same mesh-points.

References

1. U.M. Ascher, L.R. Petzold: Computer Methods for Ordinary Differential Equations and Differential Equations and Differential-algebraic Equations. SIAM, Philadelphia, 1998
2. U.M. Ascher, R. Mattheij and R. Russell: Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. SIAM, Vancouver(1988)
3. R.M. Corless, N. Fillion: A Graduate Introduction to Numerical Methods. Springer, New York, 2013
4. W.H. Enright: Continuous Numerical Methods for ODEs with Defect Control. Journal of Computational and Applied Mathematics 125:159-170 (2000)
5. G.H. Golub, F. Van Loan Charles: Matrix Computations. The Johns Hopkins University Press, New York, 2012
6. D.J. Higham: Robust Defect Control with Runge-Kutta Schemes. Siam Journal on Numerical Analysis 26(5):1175-1183 (1989)
7. E. Kreyszig: Advanced Engineering Mathematics. 9th edn. Wiley, Columbus (2005)
8. J.R. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Journal of Comparative Physiology 186(3), 219–20 (1994)
9. L.F. Shampine: Solving ODEs and DDEs with Residual Control: Applied Numerical Mathematics 52(1):113-127 (2005)
10. L.F. Shampine, M.W. Reichelt, and J. Kierzenka: Solving Boundary Value Problems for Ordinary Differential Equations in Matlab with bvp4c. ftp://ftp.mathworks.com/pub/doc/papers/bvp/ (2000)
11. E. Süli, D. Mayers: An introduction to numerical analysis. Cambridge University Press, Cambridge, 2003
12. W. Walter: Ordinary Differential Equations. Springer, New York, 1998
13. W.Y. Wu, W.Q. Yang: Error Estimation of Numerical Solvers for Linear Ordinary Differential Equations. https://arxiv.org/abs/1804.03363 (2018)
14. N. Yadav, A. Yadav, M. Kumar: An Introduction to Neural Network Methods for Differential Equations. Springer, 2015