

Applications of Spline Interpolation to Solve ODE's

Andrew M. Chuen

December 22, 2017

1 Introduction

Mathematical splines were originally introduced as a method to interpolate a set of points with lower-degree polynomials to avoid Runge's phenomenon. Cubic splines, in particular, were ideal not only due to computational efficiency with tridiagonal matrices, but also because they were low-order while still being able to account for inflection points within the intervals of interpolation. This can potentially be exploited to solve second order ODE's.

In this report, cubic spline interpolation is presented as a method to solve initial value problems. Section 2 provides a general overview of the algorithm. Test case with which to evaluate the performance of the method are presented in section 3. The results are shown and discussed in Sections 4 and 5, respectively.

2 Cubic Spline Time-Stepping Algorithm

We start by assuming that a function $f(x)$ has continuous derivatives in the range $a \leq x \leq b$. The range is broken up into n intervals through insertion of knot points x_0, x_1, \dots, x_n , where the knot points are unique ($a = x_0 < x_1 < \dots < x_n = b$). $s(x)$ is a cubic spline interpolating the function $f(x)$ if

1. $s(x)$ is a cubic polynomial in each interval $[x_i, x_{i+1}]$
2. $s(x_i) = f(x_i)$
3. $s'(x)$ and $s''(x)$ are continuous

The general formulation for a cubic spline function and its derivatives are given by the following:

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (1)$$

$$s'_i(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \quad (2)$$

$$s''_i(x) = 2c_i + 6d_i(x - x_i) \quad (3)$$

Additionally, we will assume that $f(x)$ is governed by the general 2nd order ODE.

$$f'' + p(x)f' + q(x)f = r(x) \quad (4)$$

with initial conditions

$$\begin{aligned} f(0) &= f_0 \\ f'(0) &= f'_0 \end{aligned}$$

Substituting equations (1)-(3) into equation (4) results in the following at each knot:

$$2c_i + p(x_i)b_i + q(x_i)a_i = r(x_i) \quad (5)$$

When enforcing continuity between the i th and $i + 1$ th knots, equations (1)-(3) and (5) form a system of equations from which the coefficients of the spline at x_i can be obtained, assuming the function value and first and second derivatives at the i th knot are known. Note the spacing h_i , is defined as $h_i = x_{i+1} - x_i$.

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 \quad (6)$$

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2 \quad (7)$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad (8)$$

$$c_{i+1} = r(x_{i+1}) - p(x_{i+1})b_{i+1} - q(x_{i+1})a_{i+1} \quad (9)$$

Equation (9) implicitly solves for the second derivative at the next knot. As such, iteration is needed at each step before the next time-step can be calculated. To start the iteration, an initial guess of 0 is assigned to d_i . In effect, this generates a quadratic spline that fits through the points at x_i and x_{i+1} , allowing for a_{i+1} and b_{i+1} to be evaluated. These values are then passed into equation (9) and updates the guess for d_i in (8). The process is repeated a few times at each step to ensure convergence.

3 Test Cases

Newton's Cannon and Eulerian rigid-body rotation are presented as the main test cases for this method. These cases were selected because the solutions for these problems are well known and provide a good benchmark to determine accuracy and performance of the method's implementation. Furthermore, because these problems are based in mechanics, conserved quantities such as energy can be evaluated to compare the method's performance.

3.1 Newton's Cannonball

Newton's cannonball describes the thought experiment presented by Isaac Newton hypothesizing the universal nature of gravitational force. This force determines the shape of the trajectories in the case of a two-body problem. Using Newton's second law and only accounting for gravitational force, the following non-dimensionalized system of ODE's are obtained.

$$\ddot{x} + \omega^2 x = 0 \quad (10)$$

$$\ddot{y} + \omega^2 y = 0 \quad (11)$$

where ω^2 describes the gravitational constant.

$$\omega^2 = \frac{\mu}{r^3} = \frac{\mu}{(x^2 + y^2)^{3/2}} \quad (12)$$

The gravity parameters, $\mu \in [0.4, 0.5, 0.6, 1]$, are used, which should recover hyperbolic, parabolic, elliptic, and circular orbits respectively. Additionally, energy and angular momentum will be calculated to ensure that the method is able to reproduce results without introducing noise into the system. The energy equation, H , is given by the following,

$$H = \frac{1}{2} [\dot{x}^2 + \dot{y}^2] - (x^2 + y^2)^{-1/2} \quad (13)$$

and the angular momentum equation is given by the following,

$$M = x\dot{y} - y\dot{x} \quad (14)$$

For the purposes of the report, the time-scale considered is $t \in [0, 50]$ with $\Delta t = 0.01$.

3.2 Euler Equations for Rigid Body Rotation

The Euler equations for rigid body rotation are given by the following.

$$\begin{aligned}\frac{d^2\omega_1}{dt^2} + \beta \frac{d\omega_1}{dt} &= \alpha_1\omega_1 \\ \frac{d^2\omega_2}{dt^2} + \beta \frac{d\omega_2}{dt} &= \alpha_2\omega_2 \\ \frac{d^2\omega_3}{dt^2} + \beta \frac{d\omega_3}{dt} &= \alpha_3\omega_3\end{aligned}\tag{15}$$

where α_i is determined by the following

$$\alpha_1 = \omega_3^2 \frac{(I_2 - I_3)(I_3 - I_1)}{I_1 I_2} + \omega_2^2 \frac{(I_2 - I_3)(I_1 - I_2)}{I_1 I_3}\tag{16}$$

$$\alpha_2 = \omega_1^2 \frac{(I_3 - I_1)(I_1 - I_2)}{I_2 I_3} + \omega_3^2 \frac{(I_2 - I_3)(I_3 - I_1)}{I_2 I_1}\tag{17}$$

$$\alpha_3 = \omega_2^2 \frac{(I_1 - I_2)(I_2 - I_3)}{I_3 I_1} + \omega_1^2 \frac{(I_1 - I_2)(I_3 - I_1)}{I_3 I_2}\tag{18}$$

and $I_1 = 3$, $I_2 = 2$, and $I_3 = 1$. Like the case of the Newton's cannon, the energy calculation is used to compare the performance of the spline time-stepping. The energy equation is given by the following,

$$E = \frac{1}{2} (\omega_1^2 I_1 + \omega_2^2 I_2 + \omega_3^2 I_3)\tag{19}$$

The initial conditions are given by the following,

$$\vec{\omega}(0) = \begin{bmatrix} 1 \\ 0.01 \\ 0.01 \end{bmatrix} \quad \frac{d\vec{\omega}}{dt}(0) = \vec{0}$$

and the time-scale considered is $t \in [0, 80]$ with a time-step of $\Delta t = 0.016$.

4 Results

The results of the implementation are presented in this section. The method is compared to an explicit fourth-order Runge-Kutta method that uses the same interval size (note that the RK4 method computes intermediate values within each Δt step-size). Figures 1a and 1b also show that the spline interpolation method is capable of generating an approximate solution to the problems presented in Section 3.

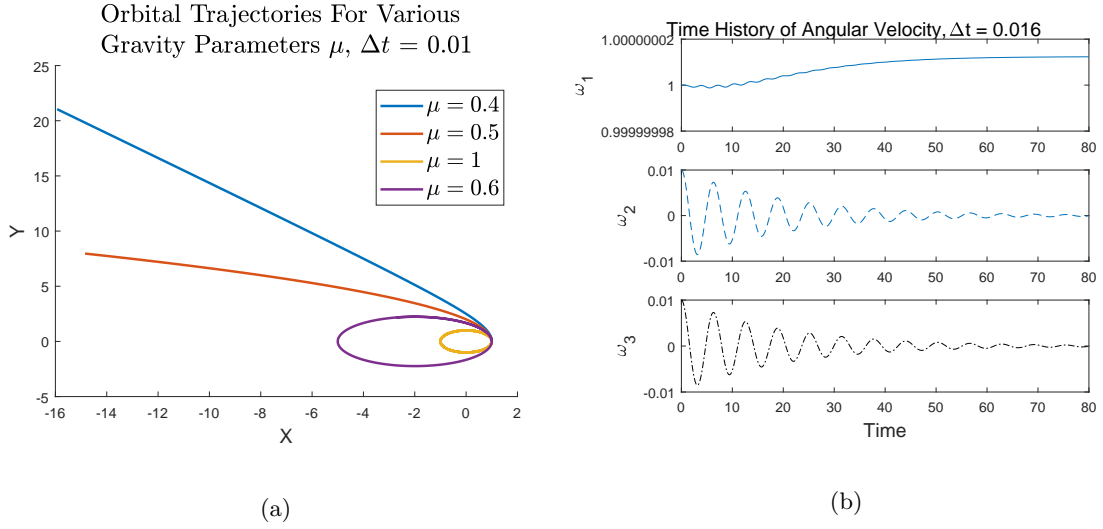


Figure 1: Solutions to systems of equations presented in Section 3.

The following figures also show the performance of the cubic-spline method in solving the ODE's and calculating energy.

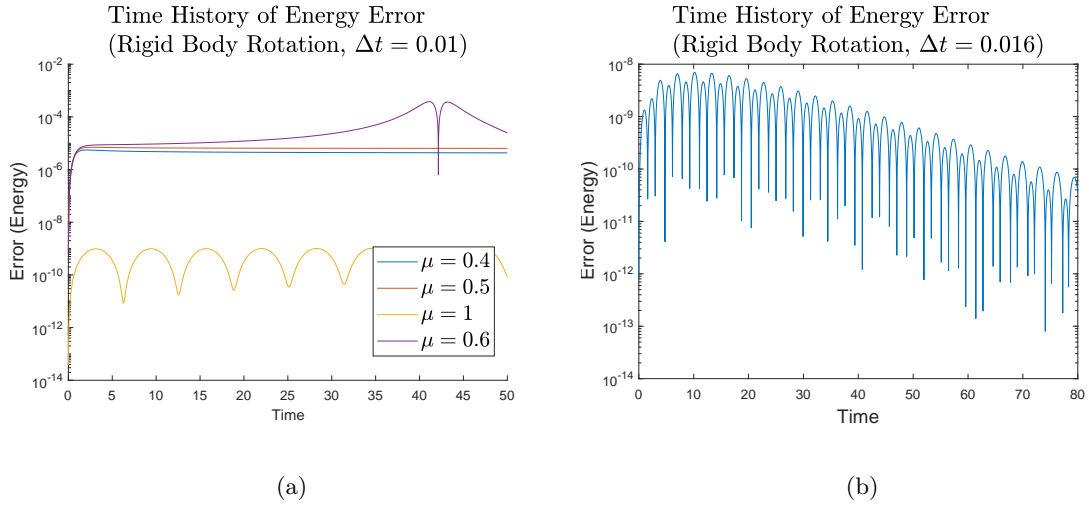


Figure 2: The plots above show the time-histories of the cubic spline's error performance.

Figure 3 also shows the convergence behavior of cubic splines. In the case of Newton's Cannonball, it has second-order convergence.

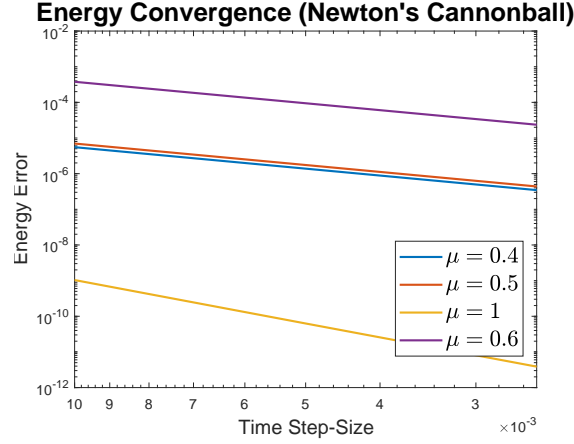


Figure 3: Convergence performance of cubic-spline in solving Newton’s Cannonball.

5 Discussion

Figures 2a and 2b show that the errors for the spline interpolation method are generally minimal, as they do not exceed 10^{-3} . In the case of rigid-body rotation, the method performs quite well as it is able to calculate the result with a nominal tolerance. In the case of Newton’s cannonball, however, the results exhibit unusual error corresponding with the trajectory of the orbit itself. It is likely that the spline is having more difficulty with proper calculations in regions with higher curvatures. Figure 3 shows that the cubic-spline does converge with smaller step-size, and it exhibits a second-order convergence rate. This shows that the method is stable and able to produce usable results.

Some possible directions to improve this method include improving the way values of d_i and the coefficients at x_{i+1} are calculated. There may be a way to feasibly obtain the coefficients at all points in a single sweep or set of iterations through the use of Newton’s method and LU-decomposition. Additionally, it may be useful to consider obtaining improved accuracy through adaptability. Accumulation of errors could be a result of insufficient points to account for difficult curvatures in the function.

Using higher-order polynomial splines is another possible direction for improvement. Higher-order polynomials may provide the additional continuity of derivatives through the knots, which can improve the order of approximation of the spline and increase the convergence rate.

A formal proof of its convergence may lend some insight into its performance and potential strategies to improve the method.

6 Conclusion

Cubic splines are presented as a method to solve ordinary differential equations. The method utilizes iteration at every time-step to implicitly calculate the third derivative and the next time-step’s coefficients. Using splines produces approximations that are capable of reaching desired tolerance levels with 2nd-order convergence. Further improvements will need to be made to make the method a viable solver.