Kevin Wang
79639666

PHYS 410 - Computational Physics - HW 1 Writeup

Introduction

In this homework I develop an integrator for solving a general system of non-linear differential equations. In particular, I implement a fourth order Runge-Kutta integrator, with the option of having adaptive steps sizes.

Review of theory

Fourth Order Runge-Kutta Approximation

Runge-Kutta approximations are a family of iterative methods for numerically approximating the solution to a system of non-linear differential equations. The methods use Simpson's rule to approximate the integration of the differential equations.

The most popular method is the fourth order Runge-Kutta method (RK4). Given an initial value problem $\frac{dy}{dx} = \boldsymbol{f}(t, y), \boldsymbol{y}(t_0) = \boldsymbol{y}_0$, and a step size $h > 0$, the solution to the problem can be approximated iteratively as

$$y(x_n + h) = y(x_n) + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{1}$$

where

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k2}{2}\right) \\ k_4 &= f(t_n + h, y_n + hk_3) \\ t_{n+1} &= t_n + h. \end{aligned}$$

Error Approximation

Let Δt be the current step size. Let the coarse RK4 approximation be

$$y_C(t_0 + \Delta t) \approx y_{exact}(t_0 + \Delta t) + k(t_0)\Delta t^5 \tag{2}$$

and the fine RK4 approximation be

$$y_F\left(t_0 + \frac{\Delta t}{2}\right) \approx y_{exact}\left(t_0 + \frac{\Delta t}{2}\right) + k(t_0)\left(\frac{\Delta t}{2}\right)^5. \tag{3}$$

Then

$$y_C(t_0 + \Delta t) - y_F(t_0 + \Delta t) \approx \frac{15}{16} e_C, \tag{4}$$

where $e_C = y_C - y_{exact}$.
The relative error is then

$$\frac{e_C^n}{y_C^n}, \tag{5}$$

where the superscripts denote some step *n*.

Physical Phenomenon Described by ODEs

The Simple Harmonic Oscillator (SHO). This is an oscillator where the magnitude of the restoring force is directly proportional to the displacement. It is governed by the IVP

$$\frac{d^2x(t)}{dt^2} = -x(t). \tag{6}$$

The Van der Pol (VDP) Oscillator. This is a non-conservative, oscillating system with non-linear damping. Its equation of motion is

$$\frac{d^2x}{dt^2} + a(x^2 - 1)\frac{dx}{dt} + x = 0. \tag{7}$$

Numerical Approach

Oscillators

The equation of motion for the SHO can be rewritten as

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(x, t) = \begin{pmatrix} x_2 \\ -x_1 \end{pmatrix}. \tag{8}$$

Similarly, for the VDP oscillator:

$$\frac{dx}{dt} = f(x, t) = \begin{pmatrix} x_2 \\ a(1 - x_1^2)x_2 - x_1 \end{pmatrix}. \tag{9}$$

While the VDP oscillator has no analytic solution, the SHO has the solution

$$x = sin(t) \tag{10}$$

given the initial conditions $x(0) = 0, x'(0) = 1$.

Problem 1

I implement a function, *rk4step*, that finds the next value in the solution after one RK4 step using **Eq. 1**. I then find the asymptotic behaviour of the local error as a function of the time step by calling the *rk4step* function and plotting the result. Specifically, I use the difference between the approximated solution to an SHO described by **Eq. 6** with initial conditions $x(0) = 0, x'(0) = 1$, and the exact solution in **Eq. 10.**

Problem 2

I implement *rk4* by calling *rk4step* iteratively, starting using the initial conditions and then using the output of each iteration as the input to each following iteration. Each time step was the difference between time values given in the list of times to be used. This function was then used to approximate the solution to an SHO with the same initial conditions as described above, using three different levels of time discretization. e_C was found for each approximation and plotted after scaling.

Problem 3

In *rk4ad*,I use the same number of subdivisions for each time step, doubling subdivisions until the relative error (**Eq. 5**) satisfies requirements. I compute each step by calling *rk4* using the subdivisions, and taking the result at the last subdivision. I did this to ensure allow uniform convergence, especially in cases where zero or near-zero values of *x* cause the relative error to blow up. The downside is that some time steps far exceed the required relative error.

Results

The local error of the integrator is $O(\Delta t^5)$. Notice that in **Fig. 1**, the fifth root of the error is linear with respect to the time step. The integrator *rk4* can be used to model an SHO and VDP oscillator. The e_C was found to obey the overall error of $O(\Delta t^4)$ as a scaling factor of $\rho = 2^4$ was used (**Fig. 2**). The Van der Pol oscillator matched theoretical results (**Fig. 3**). Similarly, *rk4ad* was also used to model the two oscillators. Curiously, the solution found for different relative error thresholds were identical (**Fig. 4**). This is because near zero points, the approximated solution was so small it required a nearly zero error. As a result, all cases resulted in a time discretization of 9. The Van der Pol oscillator modeled using *rk4ad* matches with the one modeled using *rk4*.

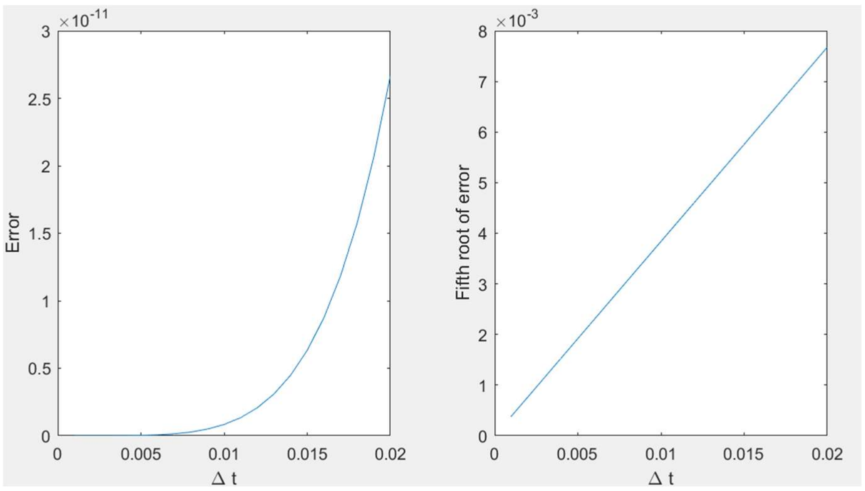

Figure 1: the difference between one step of approximation and the exact solution, for different time steps.

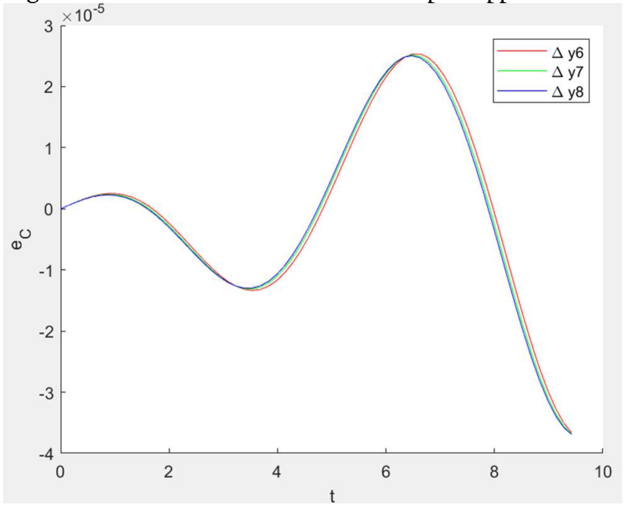

Figure 2: the e_C values corresponding to discretization levels 6, 7, and 8 for an SHO using *rk4*.

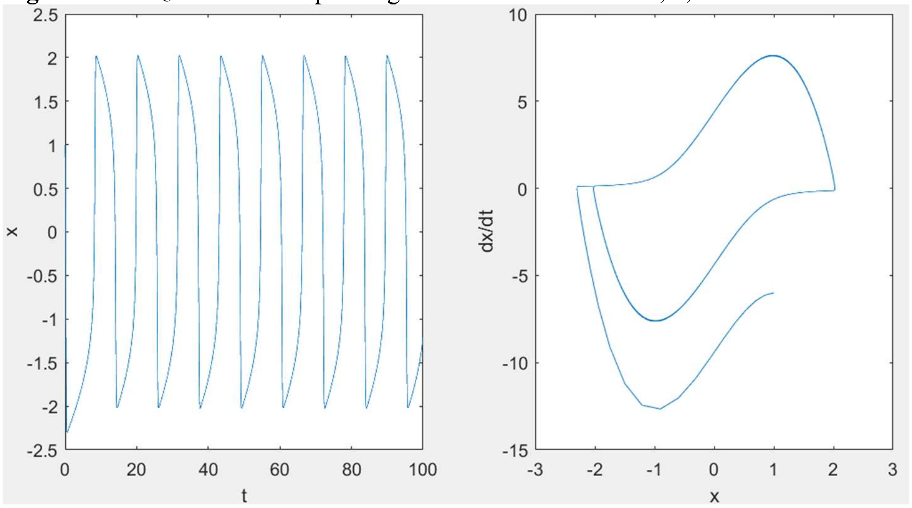

Figure 3: The VDP oscillator position over time and its phase-space evolution using *vk4*.

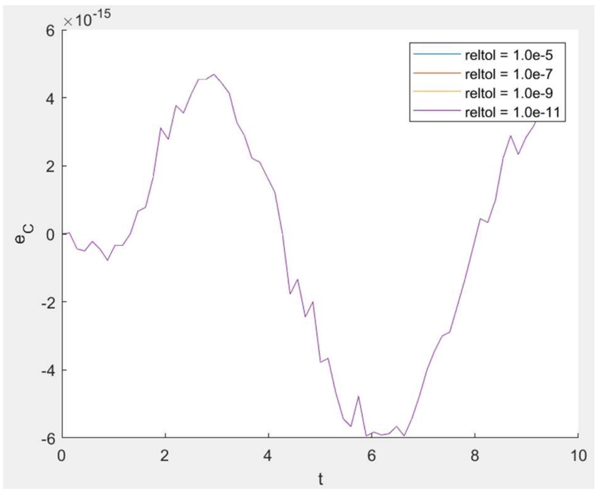

Figure 4: the e_C values corresponding to various *reltol* values for an SHO using *rk4ad*.

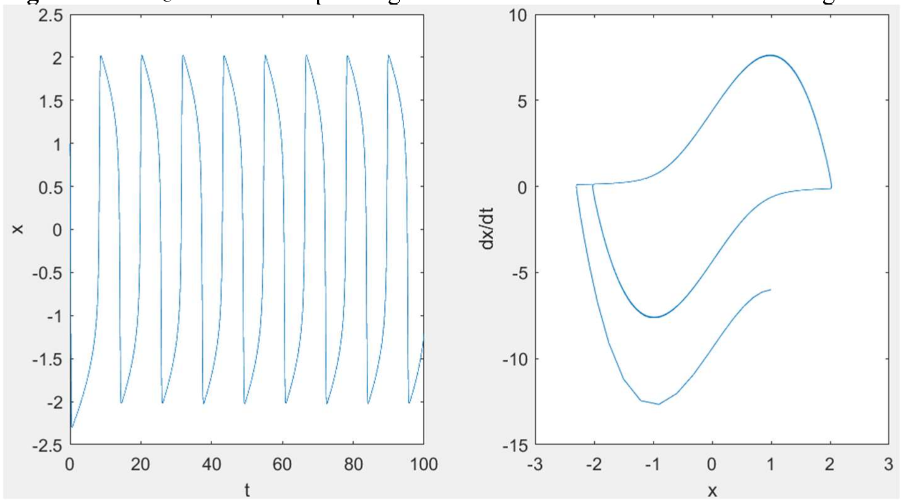

Figure 5: The VDP oscillator position over time and its phase-space evolution using *vk4ad*.

Discussion/Conclusions

Runge-Kutta methods were used to develop an integrator to approximate solutions to systems of ODEs. Error estimation methods were also used to allow for adaptive time discretization to attain required relative errors. The integrator was then used to model simple harmonic oscillator and Van der Pol oscillators.

No generative AI was used.