Homework 4: Problems (Due June 3)

Remark: for [programing] problems, you are supposed to write a brief report about the solution with code attached. You are also asked to submit a source code. Exercises and Computer Problems are taken from the textbook.

(1) Exercise 9.4 [25 points]

9.4. Consider the ODE $y' = -5y$ with initial condition $y(0) = 1$. We will solve this ODE numerically using a step size of $h = 0.5$.

(a) Are solutions to this ODE stable?

(b) Is Euler's method stable for this ODE using this step size?

(c) Compute the numerical value for the approximate solution at $t = 0.5$ given by Euler's method.

(d) Is the backward Euler method stable for this ODE using this step size?

(e) Compute the numerical value for the approximate solution at $t = 0.5$ given by the backward Euler method.

(2) Exercise 9.8 [20 points]

9.8. Consider the IVP for the ODE $y' = -y^2$ with the initial condition $y(0) = 1$. We will use the backward Euler method to compute the approximate value of the solution y_1 at time $t_1 = 0.1$ (i.e., take one step using the backward Euler method with step size $h = 0.1$ starting from $y_0 = 1$ at $t_0 = 0$). Since the backward Euler method is implicit, and the ODE is nonlinear, we will need to solve a nonlinear algebraic equation for y_1.

(a) Write out that nonlinear algebraic equation for y_1.

(b) Write out the Newton iteration for solving the nonlinear algebraic equation.

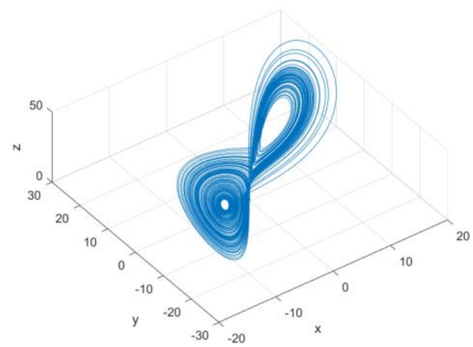
(c) Obtain a starting guess for the Newton iteration by using one step of Euler's method for the ODE.

(d) Finally, compute an approximate value for the solution y_1 by using one iteration of Newton's method for the nonlinear algebraic equation.

(3) [Programming] Consider Lorenz system [15 points]

$$\begin{cases} x' = \sigma(y - x) \\ y' = x(\rho - z) - y \\ z' = xy - \beta z \end{cases}$$

With $\rho = 28, \sigma = 10, \beta = \frac{8}{3}$, the system admits a stranger attractor as below



(a) Apply the classical fourth-order Runge-Kutta method to solve the initial-value problem of this system with the initial condition below and step size $h = 0.01$ for the time interval $[0,100]$

$$x(0) = 0.1, y(0) = 0.1, z(0) = 0.$$

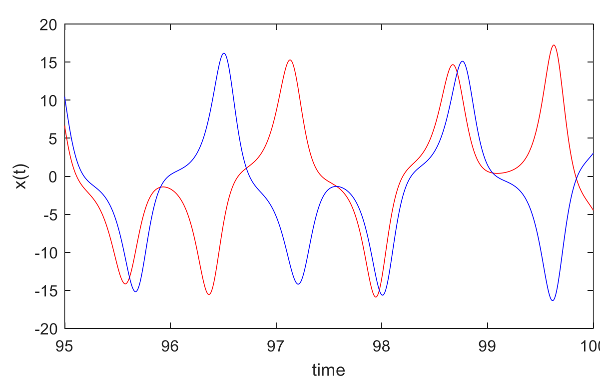
Plot the generated trajectory in phase space (x, y, z). [Hint: you can use plot3 in MATLAB to make the plot]

(b) Apply the classical fourth-order Runge-Kutta method to solve the initial-value problem of this system with the initial condition below and step size $h = 0.01$ for the time interval $[0,100]$

$$x(0) = 0.1, y(0) = 0.100001, z(0) = 0.$$

Plot the generated trajectory in phase space (x, y, z). [Hint: you can use plot3 in MATLAB to make the plot]

(c) Plot the trajectory $x(t)$ against t for $95 \leq t \leq 100$ obtained from (a) and (b), what do you find? You should be able to see that the difference between the two results is significant. This implies high sensitivity to initial conditions, a fundamental property of chaos. [Hint: you should get a plot similar to the one below]



(4) Exercise 10.3 [10 points]

10.3. Consider the two-point BVP

$$u'' = u^3 + t, \qquad a < t < b,$$

with boundary conditions

$$u(a) = \alpha, \qquad u(b) = \beta.$$

To use the shooting method to solve this problem, one needs a starting guess for the initial slope $u'(a)$. One way to obtain such a starting guess for the initial slope is, in effect, to do a "preliminary shooting" in which we take a single step of Euler's method with $h = b - a$.

(a) Using this approach, write out the resulting algebraic equation for the initial slope.

(b) What starting value for the initial slope results from this approach?

[Hint: the use of forward Euler method with $h = b - a$ yield an equation as below

$$u(b) = g(u(a), u'(a))$$

then you can use $u(b) = \beta$ to solve for the initial slope $u'(a)$.]

(5) Exercise 10.4 [25 points]

10.4. Suppose that the altitude of the trajectory of a projectile is described by the second-order ODE $u'' = -4$. Suppose that the projectile is fired from position $t = 0$ and height $u(0) = 1$ and is to strike a target at position $t = 1$, also of height $u(1) = 1$.

(a) Solve this BVP by the shooting method:

1. To determine the initial slope at $t = 0$ required to hit the desired target at $t = 1$, use the trapezoid method with step size $h = 1$ to derive a system of two equations for the unknown initial slope $s_0 = u'(0)$ and final slope $s_1 = u'(1)$.
2. What are the resulting values for the initial and final slopes?
3. Using the initial slope just determined and a step size of $h = 0.5$, use the trapezoid method once again to compute the approximate height of the projectile at $t = 0.5$.

(b) Solve the same BVP again, this time using a finite difference method with $h = 0.5$. What is the resulting approximate height of the projectile at the point $t = 0.5$?

(c) Solve the same BVP once again, this time using collocation at the point $t = 0.5$, together with the boundary values, to determine a quadratic polynomial $u(t)$ approximating the solution. What is the resulting approximate height of the projectile at the point $t = 0.5$?

[Hint: the use of the trapezoid method with $h = b - a$ yield an equation as below

$$u(b) = g_1(u(a), u'(a), u(b), u'(b))$$
$$u'(b) = g_2(u(a), u'(a), u(b), u'(b))$$

then you can use $u(b) = \beta$ to solve for the initial and final slopes $u'(a)$ and $u'(b)$.]

(6) [Programming]: Computer problem 10.1: [15 points]

10.1. Solve the two-point BVP

$$u'' = 10u^3 + 3u + t^2, \qquad 0 < t < 1,$$

with boundary conditions

$$u(0) = 0, \qquad u(1) = 1,$$

by each of the following methods.

(a) *Shooting method.* Use a one-dimensional nonlinear equation solver to find an initial slope $u'(0)$ such that the solution of the resulting initial value problem hits the target value for $u(1)$. Solve each required initial value problem using a library ODE solver or one of your own design. Plot the sequence of solutions you obtain.

(b) *Finite difference method.* Divide the given interval $0 \le t \le 1$ into $n + 1$ equal subintervals,

$$0 = t_0 < t_1 < \cdots < t_n < t_{n+1} = 1,$$

with each subinterval of length $h = 1/(n + 1)$. Let y_i, $i = 1, \ldots, n$, represent the approximate solution values at the n interior points. Obtain a system of n algebraic equations for the y_i by replacing the second derivative in the differential equation by the finite difference approximation

$$y_i''(t) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2},$$

$i = 1, \ldots, n$. Use a library routine, or one of your own design, to solve the resulting system of nonlinear equations. A reasonable starting guess for the nonlinear solver is a straight line between the boundary values. Plot the sequences of solutions you obtain for $n = 1, 3, 7$, and 15.

(c) *Collocation method.* Divide the given interval $0 \le t \le 1$ into $n - 1$ equal subintervals,

$$0 = t_1 < t_2 < \cdots < t_{n-1} < t_n = 1,$$

with each subinterval of length $h = 1/(n-1)$. Take the approximate solution $v(t, \boldsymbol{x})$ to be a polynomial of degree $n - 1$ with coefficients $\boldsymbol{x}$. Forcing $v(t, \boldsymbol{x})$ to satisfy the boundary conditions at the endpoints and to satisfy the ODE at the $n - 2$ interior points yields a system of n equations that determine the n coefficients $\boldsymbol{x}$ of the polynomial $v(t, \boldsymbol{x})$. Use a library routine, or one of your own design, to solve this system of nonlinear algebraic equations. The resulting polynomial can then be evaluated at any point in the interval to obtain an approximate solution value at that point. Print the polynomial coefficients and plot the solutions you obtain for $n = 3, 4, 5$, and 6.

(7) Consider a finite difference solution of the Poisson equation $u_{xx} + u_{yy} = x + y$ on the unit square using the boundary conditions and mesh points shown in the drawing. Use a second-order, accurate, centered finite difference scheme to compute the approximate value of the solution at the center of the square. [10 points]