Work through each of the following problems. Be sure to read the midterm project guidelines found in the resources section on sakai. Turn in all coding portions digitally and the handwritten analysis portions in class. Both portions must be turned in by the due date.

1. (6 points) **Approximating the uniform norm on $C([a, b])$**

   (a) Upgrade your `findroot` function from project 1 so that it approximates derivatives if they aren't specified. In other words, it should be capable of the following call

   findroot(f, x_0) $\leftarrow x$ attempts to find a root of f using Newton iteration with f' approximated numerically.

   (b) Define a function called `findlocalextreme` which calls `findroot` and is capable of the following call

   findlocalextreme(f, x_0) $\leftarrow [x, y]$ where x is near x_0 and $f(x) = y$ is a possible local extreme value for f.

   (c) Define a function called `uniformnorm` which calls `findlocalextreme` and is capable of the following call

   uniformnorm$(f, (a, b))$ $\leftarrow c$ where $c = ||f||_\infty$ for $f \in C([a, b])$.

2. (8 points) **Implementing cubic splines in MATLAB**

   (a) For this exercise you will need to implement a cubic splines algorithm in MATLAB. It should be called `cubicsplines` and should be capable of the following call

   cubicsplines$(\mathbf{x}, \mathbf{y}, \mathbf{b})$ $\leftarrow \mathbf{c}$ is a coefficient matrix for a cubic piecewise polynomial interpolant of the function data, $(\mathbf{x}, \mathbf{y})$, and boundary values, $\mathbf{b}$.

   These arrays have the form

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \mathrm{c} = \begin{pmatrix} c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n,0} & c_{n,1} & c_{n,2} & c_{n,3} \end{pmatrix}$$

   where $y_j = f(x_j)$ for some unknown function, f. The approximation is $S = \{S_1, \ldots, S_n\}$ where for $1 \le j \le n$ the cubic polynomial, $S_j(t) = c_{j,3}t^3 + c_{j,2}t^2 + c_{j,1}t + c_{j,0}$, is a polynomial interpolant for f on $[x_{j-1}, x_j]$, and S satisfies the boundary conditions, $S''(x_0) = b_1, S''(x_n) = b_2$.

   (b) A heuristic which measures the "smoothness" of a function, $f \in C^2[x_0, x_n]$, is given by

$$m(f) = \int_{x_0}^{x_n} (f''(s))^2 \, ds.$$

   Let S denote the natural cubic spline interpolant for some given data, $(x_0, \ldots, x_n)$, and $(y_0, \ldots, y_n)$. Show that S minimizes m among all functions in $C^2[x_0, x_n]$ which interpolate the given data.

   Hint: Let $f = C^2[x_0, x_n]$ be any competing interpolant and define $g : [x_0, x_n] \to \mathbb{R}$ by $g(t) = S(t) - f(t)$. Show that

$$m(f) = m(S) + m(g).$$

(c) Suppose f is a cubic polynomial defined on $[x_0, x_n]$. Is it possible for f to be equal to its own cubic spline interpolant? Specificially, if $S = \{S_1, \ldots, S_n\}$ is a piecewise cubic spline interpolant for f, is it possible that $S(t) = f(t)$ for all $t \in [x_0, x_n]$? If so, what (if any) restrictions must be placed on the interpolation nodes and boundary conditions? If not, show that it is not possible.

(d) Let $f(x) = \frac{1}{1+25x^2}$ be Runge's function defined on $[-1, 1]$ and $\{-1, x_1, \ldots, x_{n-1}, 1\}$ be equally spaced nodes. Let S_n be the natural cubic spline interpolant and p_n the degree n polynomial interpolant for f. For each $n \in \{2, 3, \ldots, 15\}$, compute and plot $m(S_n)$ and $m(p_n)$ together. In another plot, compute and plot $||f - S_n||_\infty$ and $||f - p_n||_\infty$. Do these plots agree with your expectation? Explain why or why not.

3. (5 points) **Error analysis for high order interpolants**

(a) Fix $n \in \mathbb{N}$, let $x_0 = a < x_1 < \ldots < x_{n-1} < x_n = b$ be an equally spaced partition of $[a, b]$, and define the corresponding nodal polynomial by

$$q_n(x) = \prod_{j=0}^{n} (x - x_j) \qquad x \in [a, b].$$

Show that $||q_n||_\infty \leq \frac{(b-a)^{n+1} n!}{4n^{n+1}}$.

(b) Suppose $f \in C^\infty([a, b])$ and let $p_n : [a, b] \to \mathbb{R}$ denote the polynomial of degree n which interpolates f on equally spaced nodes in $[a, b]$. Give a sufficient condition for the interpolation error to converge to zero (i.e. $\lim\limits_{n\to\infty} ||f - p_n||_\infty \to 0$).

(c) Give an example of a function satisfying your condition from (b). Be sure to show that it satisfies your condition.

4. (6 points) **Getting started with Runge-Kutta**

Suppose $y(t)$ is the solution to the differential equation $\dot{u} = f(t, y)$.

(a) Consider the following Runge-Kutta method for approximating $y(t + h)$ given by

$$y(t + h) = y(t) + \frac{1}{3}k_1 + \frac{2}{3}k_2 \quad \text{where } k_1 = hf(t, y(t)) \text{ and } k_2 = hf(t + \frac{3h}{4}, y(t) + \frac{3k_1}{4}).$$

Show that this method has local truncation error on the order of $\mathcal{O}(h^3)$.

(b) Implement the fourth order Runge-Kutta method given in Algorithm 5.2 of B&F in MATLAB in a function named `rk4`. It should be capable of the following call

rk4($f, (t_0, t_f, h), y_0$**)** $\leftarrow$ $(\mathbf{t}, \mathbf{y})$ returns a discretized solution to the IVP, $\dot{y} = f(t, y)$, $y(0) = y_0$ for $t \in [t_0, t_f]$, evaluated with a step size of h.

f should be a function handle which defines a vector field on $\mathbb{R}^n$ and $y_0 \in \mathbb{R}^n$ is a vector of initial data.

(c) Use your `rk4` function to integrate the following vector field

$$f(x) = \begin{pmatrix} 10(y - x) \\ 28x - y - xz \\ xy - \frac{8}{3}z \end{pmatrix}$$

from several initial points chosen randomly on the unit sphere centered at the origin. Use $h = .01$ and integrate each point for 5 time units. Plot the results and use them to make a prediction about the behavior of the system when integrated from an initial point on a sphere of radius 2. Write a short paragraph which explains why you made this prediction.