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. (9 points) **Getting started with multi-step methods**

   (a) Consider a 2 step implicit multi-step algorithm of the form

   $$x_{n+1} = c_1 x_n + c_0 x_{n-1} + h\left(\alpha_2 f(t_{n+1}, x_{n+1}) + \alpha_1 f(t_n, x_n) + \alpha_0 f(t_{n-1}, x_{n-1})\right) \quad n \geq 1.$$

   Show that this method is consistent if and only if

   $$c_0 + c_1 = 1 \quad \text{and} \quad \alpha_0 + \alpha_1 + \alpha_2 - c_0 = 1.$$

   (b) Determine the conditions on the coefficients, $(c_0, c_1, \alpha_0, \alpha_1, \alpha_2)$, for which the method in (a) is consistent, but unstable.

   (c) There is a unique choice for the coefficients, $(c_0, c_1, \alpha_0, \alpha_1, \alpha_2)$, which makes the method in (a) have order 3. Find these coefficients, show that they are unique, that the method has order 3, and that it does not have order 4.

   (d) Implement the Adam's fourth order predictor-corrector method (algorithm 5.4 in B&F) in MATLAB. It should be a function called `ms4` which is capable of the following call

   **ms4**$(f, (t_0, t_f, h), x_0) \leftarrow (\mathbf{t}, \mathbf{x})$ are vectors representing a discretized solution to the initial value problem

   $$\dot{\mathbf{x}} = f(t, \mathbf{x}), \qquad \mathbf{x}(0) = \mathbf{x}_0 \qquad t \in [t_0, t_f]$$

   in the sense that $x_n \approx x(t_n)$, evaluated with a step size of $h$. Use `rk4` from midterm project 2 for steps 2-5 of the algorithm.

## Investigating a nonlinear oscillator

The remaining exercises focus on numerical simulation of a model for nonlinear oscillation. The motion of an idealized point-mass at the end of a rigid pendulum is described by solutions to the $2^{\text{nd}}$ order initial value problem

$$\ddot{\theta} + \gamma\dot{\theta} + \sin(\theta) = 0 \qquad \theta(0) = \theta_0 \qquad \dot{\theta}(0) = \dot{\theta}_0$$

where $\theta(t)$ is the angle formed by the pendulum and an imaginary vertical line after $t$ units of time and $\gamma \geq 0$ is a constant which determines the amount of damping. For each exercise you should consider the equivalent equation written as a vector field on $\mathbb{R}^2$.

2. (7 points) **Linear approximation of the pendulum**

(a) For this exercise you will need to implement a linear solver based on Gaussian elimination with pivoting. It should be called `linearsolver` and should be capable of the following calls

**linearsolver**$(U, b)$ $\leftarrow x$ satisfies the linear equation $Ux = b$ where $U$ is an $n \times n$ real, nonsingular, upper triangular matrix and $b$ is a column vector of length $n$. Your function should compute $x$ directly by back substitution.

**linearsolver**$(A, b)$ $\leftarrow x$ returns the solution to $Ax = b$ where $A$ is any $n \times n$ real nonsingular matrix. In this case your function should use Gaussian elimination with pivoting to obtain an upper triangular linear problem, and then compute the solution by recalling `linearsolver`.

**linearsolver**$(A, B)$ $\leftarrow X$ where both $B$ and $X$ are matrices with $n$ rows. The $j^{\text{th}}$ column of $X$ is the vector $x_j$ which satisfies $Ax_j = b_j$ where $b_j$ is the $j^{\text{th}}$ column of $B$. In this case your function should first compute an $LU$ factorization for $A$, and then compute each column of $X$ using 2 calls to `linearsolver` with upper triangular matrices.

Your function should return an error if the matrix input is singular. You may *not* use the builtin MATLAB linear solver, `lu`, or `inv` functions. However, you may use the builtin functions `triu`, `tril`, and `rank` as needed.

(b) Write a MATLAB function called `solvelinearIVP` which calls `linearsolver` to solve the linear homogeneous initial value problem

$$\dot{\mathbf{x}} = A\mathbf{x} \qquad \mathbf{x}(0) = \mathbf{x}_0$$

where $A$ is a real $n \times n$ matrix and $\mathbf{x}_0$ is a column vector of initial data. It should be capable of the following call

**solvelinearIVP**$(A, \mathbf{x}_0)$ $\leftarrow \mathbf{x}$ where $x$ is a function handle for the exact solution to the initial value problem (i.e. the call $\mathbf{x}(t)$ should evaluate the solution at time $t$).

Your function should call your `linearsolver` function and may also call the builtin functions `eig` and `exp`.

(c) Classify the qualitative dynamics which arise from the solutions to pendulum equation linearized around an equilibrium. Your analysis should consider the possible linearizations at any equilibrium solution, as well as the effect of varying $\gamma \in [0, \infty)$. Use this analysis to make a script called `linearized_pendulum_dynamics` which calls `solvelinearIVP` for each type of dynamics and plots solutions for initial conditions chosen uniformly on the unit circle.

Make a separate plot for each linear equation. Choose enough initial conditions and an integration time for each plot so that it captures the qualitative dynamics. Use each plot to confirm that the trajectories behave as predicted by your analysis.

3. (9 points) **Investigating the nonlinear pendulum**

The remaining exercises focus on studying a model for nonlinear oscillation. The motion of an

(a) Suppose $f$ is a vector field on $\mathbb{R}^2$ of the form $f(x, y) = (y, g(x))$ for some scalar function $g$. Show that the function

$$H(x, y) = \frac{1}{2}y^2 - \int_{x_0}^{x} g(s)\ ds$$

is constant along all trajectories of the equation $(\dot{x}, \dot{y}) = f(x, y)$. In other words, solutions of this ODE lie on level curves of $H$.

Compute $H$ for the undamped pendulum equation and write a script called `calibrate_step_size` which plots level curves of $H$ in the domain $[-\pi, 5\pi] \times [0, 5]$. Begin with a step size $h = 10^{-1}$ and plot trajectories from initial conditions uniformly along the line segment between $(0, 0)$ and $(0, 5)$ and integrated for $t = 30$ time units.

Repeat this procedure, decreasing $h$ by a factor of 10 each time, until all plotted solutions appear to align with the level curves of $H$. Make a separate plot for each attempt and leave all of the code for each plot intact. For the remaining exercises involving `ms4` and the pendulum equation, use your final step size to compute solutions.

(b) Use `ms4` to compute a solution to the IVP with $\gamma = 0$, $\theta_0 = \frac{5\pi}{4}$, $\dot{\theta}_0 = \sqrt{2 - \sqrt{2}}$ for $t \in [0, 40]$, and plot this trajectory segment in the $(\theta, \dot{\theta})$ plane. What is wrong with the solution computed by `ms4`? Does this picture mean that `ms4` is numerically unstable? Can this problem be fixed by decreasing the step size, increasing the order of integration, or other measures? Explain your answers thoroughly.

(c) Suppose the pendulum is held along the vertical line in the "up" position ($\theta_0 = \pi$) and then pushed with some initial velocity so that it begins to swing. If $\gamma > 0$, then the pendulum will lose energy along its trajectory and converge to a stable equilibrium. Hence, no matter how large the initial velocity, the pendulum can only complete a finite number of revolutions around the circle.

Suppose we are interested in trajectories which complete *exactly* 2 revolutions. Define a function, $g : [0, 10] \to (0, \infty)$, by $g(x) = \gamma_x$ where $\gamma_x$ is the maximum possible damping such that for given initial velocity, $\dot{\theta}_0 = x$, the pendulum completes *exactly* 2 revolutions. Make a MATLAB script called `revolution_plot` which produces a plot of the graph of $g$. You may use `ms4` for performing integrations and any code from previous projects as needed.

(d) This portion should be completed after finishing 2(c). Make a script called `pendulum_dynamics` and repeat the plots from `linearized_pendulum_dynamics` for all equilibrium solutions in the domain $[-\pi, 5\pi] \times [-5, 5]$. The solutions for each linearized equation should be centered over the corresponding equilibrium and colored distinctly.

For each initial condition integrated in one of the linearized equations, use `ms4` to compute the trajectory for the nonlinear equation. Integrate each for 20 time units and plot the solutions in a distinct color and with thicker lines to distinguish it from the linearized solutions.

Write a few sentences for each plot which describes the regions for which the nonlinear solutions are well approximated by the linearized solutions, and the regions where they are poorly approximated. What do thee regions have in common? Identify characteristics of the motion for the linear equations which are not present for the nonlinear motions and vice versa. Be specific.