CS/MATH 375, Fall 2024 — HOMEWORK # 13
Due : Dec. 6th at 10:00pm on Canvas

Instructions

- **Report:** In general, your report needs to read coherently. That is, start off by answering question 1. Fully answer the question, and provide all the information needed to understand your answer. If Matlab code or output is part of the question, include that code or output (e.g., screenshot) alongside your narrative answer. If discussion is required for a question, include that. Overall, your report is your narrative explanation of what was done, your answers to the specific questions, and how you arrived at your answers. *Your report should include your Matlab scripts, code output, and any figures.*

- **What to hand in:** Submission must be one **single PDF** document, containing your entire report, submitted on Canvas.

- **Partners:** You are allowed to (even encouraged) to **work in pairs**. If you work with a partner, only one member of the group should need to submit a report . On Ca nvas, both partners should join a group (numbered 1 through 15). Then either member can upload the report for the entire group. Groups of more than 2 students are not allowed.

- **Typesetting:** If you write your answers by hand, then make sure that your handwriting is readable. Otherwise, I cannot grade it.

- **Plots:** All plots/figures in the report must be generated in Matlab or Python and not hand drawn (unless otherwise specified in the homework question).

  In general, make sure to (1) title figures, (2) label both axes, (3) make the curves nice and thick to be easily readable, and (4) include a legend for the plotted data sets. The font-size of all text in your figures must be large and easily readable.

---

1. Gaussian Quadrature.
   Remember the integral
   $$I = \int_0^{\pi} x\sin(x)\,dx$$
   from the previous Homework, where you computed the solution by hand and approximated it by the composite trapezoid method. We will now approximate it using Gaussian quadrature.

   (a) Approximate the integral by hand using the Gaussian quadrature rule with three points. The quadrature rule on the interval $[-1,1]$ is given by
   $$\int_{-1}^{1} f(x)dx \approx w_0 f(x_0) + w_1 f(x_1) + w_2 f(x_2)$$

where $x_0 = -\sqrt{3/5}$, $x_1 = 0$, $x_2 = -x_0$, and $w_0 = 5/9$, $w_1 = 8/9$, $w_2 = w_0$. Note: you will need to do the change of interval as described in lecture/book in order to get an approximation over the arbitrary interval $[a, b]$.

(b) Write a MATLAB function `integral = gauss_int(f,a,b,k)` that implements the k-point Gaussian quadrature rule on the interval $[a, b]$. The values `a` and `b` are the limits of integration and `f` is the function to be integrated. Your function only needs to work for $k \leq 8$. I have provided a MATLAB function `lgwt_table.m` on the Canvas page which you should feel free to use. It produces the Gaussian weights and nodes for the k-point quadrature rule on the interval $[-1, 1]$.

To test your code, use it with $k = 3$ and show that it gives the same answer as your by hand calculations in the previous question.

(c) Use your code from the previous question to approximate the integral for $k = 2, 3, \ldots 8$. For each approximation, compute the error. Generate a semi-log plot of error as a function of k. Does your plot look linear? If so, it would imply that the error is $\mathcal{O}(e^{bk})$ for some constant b (i.e. Gaussian quadrature is geometrically convergent).

(d) Write a MATLAB function `integral = comp_gauss_int(f,a,b,k,n)` that implements composite Gaussian quadrature on the interval $[a, b]$ using n sub-intervals, and k-point quadrature on each sub interval. Here, `f` is the function to be integrated.

To write this composite integral function, use `integral = gauss_int(...)` to approximate the integral on each of the (smaller) intervals $[x_{i-1}, x_i]$, for $x_0 = a$, $x_i = x_0 + ih$, $h = (b - a)/n$. Then sum up the contributions on each of the smaller intervals to approximate the entire integral over $[a, b]$, again analogous to composite trapezoid.

(e) Compute an approximate integral using composite Gaussian quadrature with $k = 3$-point quadrature on each sub-interval for $n = 4, 8, 16, 32$. Tabulate your results by showing three columns: the value of the approximate integral computed by composite Gaussian quadrature, the error, and the observed algebraic order of convergence p for each value of n. Recall that the order of convergence p is computed as

$$p = \frac{\log(err^{(n)}/err^{(n-1)})}{\log(h^{(n)}/h^{(n-1)})}$$

where $err^{(n)}$ is the error for n intervals and $h^{(n)} = (b - a)/n$ is the sub-interval length. Plot the error as a function of n on a log-log plot. Does this plot confirm the order of convergence suggested by the table above?

(f) Repeat the previous problem with $k = 4$.

2. Write a Matlab script to compute an Euler approximation to the IVP

$$x' = f(t, x), \quad x(t_0) = x_0.$$

Your method should have the following form: `[t,x] = my_euler(f,x0,t0,tend,n)`. The input parameters are (in order) a function that evaluates the right hand side of the ODE, the initial condition x_0, the left end-point of the interval (t_0), the right end-point of the interval, and the number of steps to take. The outputs should both be size $n+1$ vectors which contain the discretized interval $t_0, t_1, \ldots t_n$ and the approximate solution at these points $x_0, x_1, \ldots x_n$.

3. Consider this initial value problem (IVP)

$$\begin{cases} y'(t) = (y(t))^2 - y(t), & t \in [0, 2], \\ y(0) = 0.5. \end{cases}$$

Approximate the solution to this IVP with the Euler method, using $h = 1.0, 0.5, 0.25,$ and 0.125. Plot the approximate solutions on the same axes.

4. Consider the following IVP

$$y'(t) = (y(t))^{(1/3)} \tag{1}$$
$$y(0) = 0 \tag{2}$$

Show that both $y(t) = 0$ and $y(t) = (2/3t)^{3/2}$ are solutions to this IVP. Trying approximating the solution with Euler's method and $h = 0.1$ Which of these two solutions does Euler's method find?

5. Suppose that an ordinary differential equation is solved numerically on an interval $[a, b]$ and that the local truncation error is ch^p. Show that if all truncation errors have the same sign (the worst possible case), then the total truncation error is $(b-a)ch^{p-1}$, where $h = (b-a)/n$.

6. Consider the following ODE:

$$\begin{cases} u' = -u^2 - 2\sin(2t) + (\cos(2t))^2, & t \in [0, 1] \\ u(0) = 1 \end{cases}$$

   (a) Verify that $u(t) = \cos(2t)$ satisfies both the ODE and the initial condition.

   (b) Use your Euler method to solve the above ODE using $n = 10, 20, 40, 80$ time steps ($h = 0.1, 0.05, 0.025, 0.0125$) Tabulate your results by showing four columns:

the step size h, the value of the approximation at $t = 1$, the error, and the observed order of convergence p for each value of h. Recall that the order of convergence p is computed as

$$p = \frac{\log(err(h_k)/err(h_{k-1}))}{\log(h_k/h_{k-1})}$$

where $err(h_k)$ is the error at $t = 1$ corresponding to a step size of h_k. Do you see the expected convergence? If not, try to explain your results.

7. Write a Matlab script to compute a fourth order Runge-Kutta approximation to the IVP

$$x' = f(t, x), \quad x(t_0) = x_0.$$

Your method should have the following form: `[t,x] = my_rkr4(f,x0,t0,tend,n)`. The input parameters are (in order) a function that evaluates the right hand side of the ODE, the initial condition x_0, the left end-point of the interval (t_0), the right end-point of the interval, and the number of steps to take. The outputs should both be size $n + 1$ vectors which contain the discretized interval $t_0, t_1, \ldots t_n$ and the approximate solution at these points $x_0, x_1, \ldots x_n$.

8. Next, consider an IVP that represents a flame problem (from Mathworks). When combustion starts (say a match is lit), then the ball of flame expands quickly until it reaches a steady state. This steady state represents the balance between the fuel on the surface of the combusting material and the fuel in the interior of the material expanding.

A straight-forward IVP models this process well,

$$\begin{aligned} y'(t) &= (y(t))^2 - (y(t))^3, \qquad 0 \le t \le 2/\delta && (3) \\ y(0) &= \delta, && (4) \end{aligned}$$

where $y(t)$ is the radius of the flame (and $(y(t))^2$ and $(y(t))^3$ are the surface and volume contributions). The value δ is the initial radius.

Use the Runge-Kutta Method you wrote in the previous problem to solve this ODE with $\delta = 0.01$. Try using $h = 10.0$ and $h = 0.1$ For both h-values, provide a plot of the solution, with the x-axis representing time, and the y-axis representing the flame radius.

Provide one or two sentences interpreting the overall shape of the plot for $h = 0.1$. That is, how is the flame radius varying over time?

Provide one or two sentences explaining the reason(s) for any differences between your two plots.

9. Find the region of stability for the RK2a method (from lecture). Express your answer as an inequality using the variable $z = \lambda h$, where h is the step-size of the method and λ is the constant from the Dahlquist problem. Graph the region in the complex plane.