I. BTM Chapter 3 Practice and MATLAB 3D Plots (15 minutes)

1. One of the two partners should log on to your Physics Department account (flip a coin to decide), locate the 1094 Session 3 files on the 263 home page (http://www.physics.ohio-state.edu/~ntg/263), and save them in your own 263 directory. You’ll also need the files from Session 2. Start up MATLAB and set the current directory to your 263 folder.

2. For \( f(x, y) = x^3 y - e^{xy} \), evaluate \( f_{xx}, f_{xy}, \) and \( f_{yx} \). Check your answers using MATLAB, following the “MATLAB Cheat Sheet III” as needed. Write the MATLAB commands.

3. Consider the function \( f(x, y) = x^2 + 2y^2 - 3xy + x \). Where are the stationary points and what type are they (maximum, minimum, saddlepoint)? Follow the ”MATLAB Cheat Sheet III” instructions to plot the function using surf and contourf. Correct your answers if they disagree with the plot. [Note: For the surface plot, use the ”Rotate 3D” button to look at it from different viewpoints.]

II. Numerical Derivatives and Round-Off Error (15 minutes)

1. Run deriv_test.m by typing deriv_test at the >> prompt, enter 2 for \( x_0 \), and look at the plot that is produced. This plot is of the error made by using approximation 1 (see the Session 2 “Cheat Sheet”) for the derivative of a test function: \( f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \). If \( y = Cx^\alpha \), then \( \log y = \log C + \alpha \log x \), and a log-log plot is a straight line with slope \( \alpha \). What does the slope of the line in the plot tell you about how the error depends on \( \Delta x \)? Apply the Taylor expansion of \( f(x + \Delta x) \) to approximation 1 through \( O(\Delta x)^2 \) to derive the theoretical result for the error. Are the two results consistent?

2. Find deriv_test.m and test_function1.m in the directory listing and bring them up in the “Editor”. Look through the listings and try to explain to your partner how they work. What is the test function?
3. Modify `deriv_test.m` so that the lower limit of $\Delta x$ is $10^{-20}$ and then run `deriv_test` again. Now there are three regions in the plot. What you are seeing is evidence for “round-off error” in the calculation. Because decimal numbers are represented with a finite number of binary digits (0’s and 1’s), there is a limit on how close two numbers can get and still be represented accurately. This is called the “machine precision”. It means that when $\Delta x$ gets too small, the difference $f(x_0 + \Delta x) - f(x_0)$ reaches a minimum approximately equal to the machine precision. Why does this lead to the slope that you observe below $10^{-5}$? Bonus: Speculate on what is happening at still smaller $\Delta x$.

4. Bonus: Modify `deriv_test.m` to add a second line to the plot that calculates the approximate derivative using method 3. How do the slopes compare? What do you conclude?

III. Numerical Integration and Round-Off Error (15 minutes)

1. Find `integ_test.m`, `integ_naive.m` and `test_integrand1.m` in the directory listing and bring them up in the “Editor”. The “Numerical Integration Background I” handout has a brief description of numerical integration. Look at the program listings and try to figure out how they work. What is the integrand (i.e., what function is it)? Change the maximum number of subintervals from 200 to 1000. What command did you change?

2. Run `integ_test.m` by typing `integ_test` at the >> prompt and look at the plot produced. This plot is of the error made in evaluating the integral of the test function with the crudest possible integration method. What is the slope of the line? What does that imply for how the error scales with $\Delta x$?

3. Modify `integ_test.m` to add a second line to the plot that calculates the approximate integral using the trapezoid rule according to the handout. You’ll need to copy `integ_naive.m` to `integ_trapezoid.m` and change it as well. [Hint: change `weights(N+1)` and `weights(1)`.
How do the slopes of the error for the naive and trapezoid methods compare? What is your interpretation?

4. Bonus: Modify `integ_test.m` so that the maximum number of subintervals is 10000 (you can increase Delta_N so it runs faster). What is happening in the left region of the plot? In `test_integrand1.m`, remove single to switch to the default “double precision” [that is, change to `result = exp(x)`]. How does that affect the behavior for small $\Delta x$?