Summer 2021: Numerical Analysis
Assignment 1 (due July 12, 2021 at 11am ET)

Homework submission. Homework assignments must be submitted through Gradescope. Please hand in cleanly handwritten or typed (preferably with LaTeX—I will provide the source files of these assignments if you want to use them to learn LaTeX) homeworks. If you are required to hand in code or code listings, this will explicitly be stated on that homework assignment.

Collaboration. NYU's integrity policies will be enforced. You are encouraged to discuss the problems with other students on Campuswire. However, you must write (i.e., type) every line of code yourself and also write up your solutions independently. Copying of any portion of someone else's solution/code or allowing others to copy your solution/code is considered cheating.

Plotting and formatting. Plot figures carefully and think about what you want to illustrate with a plot. Choose proper ranges and scales (`semilogx, semilogy, loglog`), always label axes, and give meaningful titles. Sometimes, using a table can be useful, but never submit pages filled with numbers. Discuss what we can observe in and learn from a plot. Use `format compact` and other format commands to control MATLAB' outputs. When you create figures in MATLAB (or Python), please export them in a vector graphics format (.eps, .pdf, .dxf) rather than raster graphics or bitmaps (.jpg, .png, .gif, .tif). Vector graphics-based plots avoid pixelation and thus look much cleaner.

Programming. This is an essential part of this class. We will use MATLAB for demonstration purposes in class, but you are free to use other languages (Python, Julia). Basic programming skills are crucial for many jobs, so this is also a good chance to get more comfortable with it, if you aren't already. In your programs, please use meaningful variable names, try to write clean, concise and easy-to-read code and use comments for explanation.

1. **[5pt]** We search for solutions in $[1, 2]$ to the equation

$$x^3 - 3x^2 + 3 = 0.$$

   (a) Compute the first iterates $x_0, \dots, x_5$ of the secant method in $[1, 2]$.

   (b) Compute the first iterates $x_0, \dots, x_5$ using Newton's method with starting value $x_0 = 1.5$.

   (c) Compute the first iterates using Newton's method with starting value $x_0 = 2.1$. Sketch the equation graph and try to explain the behavior.

2. **[10pts]** In this problem, you will prove the rate of convergence for the secant method.

   (a) Show that the secant method

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$$

   can be rewritten in the form:

$$x_{k+1} = \frac{x_k f(x_{k-1}) - x_{k-1} f(x_k)}{f(x_{k-1}) - f(x_k)}. \tag{1}$$

(b) Now, denote the root of f to be ξ, so that $f(\xi) = 0$. Also assume that f is twice continuously differentiable and that $f' > 0$ and $f'' > 0$ in a neighborhood of ξ. Define the quantity ψ to be:

$$\psi(x_k, x_{k-1}) = \frac{x_{k+1} - \xi}{(x_k - \xi)(x_{k-1} - \xi)},$$

where x_{k+1} is as in (1). Compute (for fixed value of x_{k-1})

$$\varphi(x_{k-1}) = \lim_{x_k \to \xi} \psi(x_k, x_{k-1}).$$

(c) Now compute

$$\lim_{x_{k-1} \to \xi} \varphi(x_{k-1}),$$

and therefore show that

$$\lim_{x_k, x_{k-1} \to \xi} \psi(x_k, x_{k-1}) = \frac{f''(\xi)}{2f'(\xi)}.$$

(d) Next, assume that the secant method has convergence order q, that is to say that

$$\lim_{k \to \infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^q} = A < \infty.$$

Using the above results, show that $q - 1 - 1/q = 0$, and therefore that $q = (1 + \sqrt{5})/2$.

(e) Finally, show that this implies that

$$\lim_{k \to \infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^q} = \left(\frac{f''(\xi)}{2f'(\xi)} \right)^{q/(1+q)}.$$

3. **[5pt]** Consider the following ordinary differential equation (ODE):

$$\frac{du}{dt} = g(u).$$

To solve this numerically, you can use the backward Euler method, for some time step $\Delta t > 0$ (we will talk about this later in the semester):

$$\frac{u^{n+1} - u^n}{\Delta t} = g(u^{n+1}).$$

The numerical result from this process is the sequence $u^0, u^1, u^2, \ldots$, which can be interpreted as an approximation to the exact solution sampled at times $0, \Delta t, 2\,\Delta t, \ldots$.

(a) If $g(u) = au$ for some $a < 0$, derive a formula for u^{n+1} as a function of u^n.

(b) If $g(u)$ is a general nonlinear function and is differentiable, write down an iteration which determines u^{n+1} from Newton's method.

(c) The convergence of Newton's method depends on the choice of the initial guess. What would be a sensible choice for an initial guess?

4. **[5pt]** Raytracing is an algorithm that involves finding the point at which a ray (a line with a direction and an origin) intersects a curve or surface. We will consider a ray intersecting with an ellipse. The general equation for an ellipse is

$$\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2 - 1 = 0$$

and the equation for a ray starting from the point $P_0 = [x_0, y_0]$ in the direction $\mathbf{V}_0 = [u_0, v_0]$, is

$$\mathbf{R}(t) = [x_0 + tu_0, y_0 + tv_0]$$

where $t \in [0, \infty)$ parameterizes the ray. In this problem we will take $\alpha = 3$, $\beta = 2$, $P_0 = [0, b]$, $\mathbf{V}_0 = [1, -0.3]$. Using your favorite root finding algorithm write a code which computes the intersection of the given ray and the ellipse and plot your results. .

(a) Plug the equation for the ray, $\mathbf{R}(t)$, into the equation for the ellipse and analytically (with pen and paper) solve for the value of t which gives the point of intersection, call it t_i.

(b) Perform the same calculation numerically using your favorite root finder. Report your answer to within an error of 10^{-6} and justify how you found the minimum number of iterations required to achieve this tolerance. Also report the point of intersection $P_i = \mathbf{R}(t_i)$