Assignment #3

Due Wednesday, 15 February 2023, at the start of class

There will be no lectures on 8 & 10 February (Wednesday & Friday). Instead, please carefully go through the following slides:

`bueler.github.io/nade/assets/slides/iterative.pdf`

This Assignment is based on these slides, and previous material. However, sections 4.1, 4.2, and 2.16 in the textbook complement this material.

P12. **(a)** Write a MATLAB/etc. function for Richardson iteration, with signature

```
function z = richardson(A,b,x0,N,omega)
```

It should return the Nth iterate $\mathbf{x}_N$ as `z`. Confirm that it works by showing you get the same $\mathbf{x}_3$ as on page 4 of the slides.

(b) How many iterations are needed to get 8 digit accuracy for LS1 with $\mathbf{x}_0 = 0$ and using the preferred value of ω? (*Clearly state how you interpret "8 digit accuracy".*) How many iterations for $\omega = 0.1$ and $\omega = 0.5$?

P13. **(a)** Write MATLAB functions which do N iterations of the Jacobi and Gauss-Seidel (GS) methods:

```
function z = jacobi(A,b,x0,N)
  function z = gs(A,b,x0,N)
```

For each one use the entries of A directly. That is, for `jacobi()`, implement formula (5) from the slides, and for `gs()` implement formula (7). (*Do not split $A = D - L - U$ and store those parts; this is a waste of memory and misses the point.*) Your implementation of GS should use less memory than Jacobi; make sure this is clear.

(b) For each method, how many iterations are needed to get 8 digit accuracy for LS1 using $\mathbf{x}_0 = 0$? (*Clearly state how you interpret "8 digit accuracy".*)

(c) Demonstrate that GS fails on LS2. Now compute an explanatory spectral radius.

P14. Show that Jacobi iteration converges if A is strictly diagonally-dominant. (*Hints: Jacobi iteration converges if and only if $\rho(M) < 1$ for $M = -D^{-1}(L + U)$. So suppose $M\mathbf{v} = \lambda\mathbf{v}$ for $\mathbf{v} \neq 0$. Choose the largest-magnitude entry v_i of $\mathbf{v}$, so that $|v_i| \geq |v_j|$ for all j. Show then that $M\mathbf{v} = \lambda\mathbf{v}$, and the assumption of strict diagonal dominance, shows $|\lambda v_i| < |v_i|$ which shows $|\lambda| < 1$.*)

P15. **(a)** Consider this boundary value problem from **P10** on Assignment #2:

$$u''(x) + q\,u(x) = f(x), \quad u(x_L) = \alpha, \quad u(x_R) = \beta.$$

Implement the centered finite difference method for this problem. Your code should have the signature

```
function [x,u] = bvpq(m,xL,xR,q,f,alpha,beta)
```

where input `f` is the function $f(x)$, but the other inputs are integers or real numbers. The outputs are the grid vector `x` and the (approximate) solution vector `u`. In this initial implementation, your code should use MATLAB's backslash command, or similar built-in solver, to solve the linear system. (*Include the codes from parts* ***(a)*** *and* ***(d)*** *in what you turn in.*)

(b) Check correctness of `bvpq` by solving the problem

$$u''(x) - u(x) = f(x), \quad u(0) = 1, \quad u(2) = 0.$$

exactly, using the solution $u_{\text{ex}}(x) = 1 - \sin(\pi x/4)$. That is, start by finding the $f(x)$ for which this $u_{\text{ex}}(x)$ is the exact solution. (*This is using the method of* manufactured solutions.) Show a figure which confirms that you have a verified code.

(c) Your part **(a)** code sets up and solves a linear system $AU = F$. For what q values is A strictly diagonally-dominant (SDD)?

(d) Duplicate the code from part **(a)**, give it a new name `bvpqgs`, and implement Gauss-Seidel (GS) to solve the linear system, instead of calling the built-in linear solver. Let $x_L, x_R, f(x), \alpha, \beta$ all be as in part **(b)**. For each of $m = 5$ and $m = 50$ find nonzero values q where Gauss-Seidel does converge and does not converge. (*That is, find 4 values of q with these properties.*) When convergence happens, how many iterations give 8 digit accuracy?

P16. In calculus you probably learned Newton's method as a memorized formula: $x_{k+1} = x_k - f(x_k)/f'(x_k)$. Rewrite equations Newton's method equations (8), (9) from the slides, in the one-dimensional case ($n = 1$), to derive this memorized formula.

P17. **(a)** Consider these 3 equations, chosen for visualizability:

$$x^2 + y^2 + z^2 = 4, \quad x = \cos(\pi y), \quad z = y^2$$

Sketch each equation individually as a surface in $\mathbb{R}^3$. (*Do this by hand or by computer, the goal being a clear mental image of intersecting surfaces. Accuracy is not important.*) Describe informally why there are two solutions of this system of three equations, that is, two points $(x, y, z) \in \mathbb{R}^3$ at which all three equations are satisfied. Explain why both solutions are inside the box $-1 \le x \le 1, -2 \le y \le 2, 0 \le z \le 2$.

(b) The slides describe Newton's method for nonlinear systems. Implement it in MATLAB/etc. to solve the above nonlinear system. Show your script and generate at least five iterations. Use $\mathbf{x}_0 = (-1, 1, 1)$ as an initial iterate to find one solution, and also find the other solution using a different initial iterate. Note that `format long` is appropriate here.