HOMEWORK #2

Due: October 29 (Thursday) by midnight

Instructions:

- The assignment consists of *three* questions, worth 4, 3 and 3 points.
- Submit your assignment *electronically* to the Email address `math3q03@math.mcmaster.ca`; hardcopy submissions will not be accepted.
- It is obligatory to use the MATLAB template file available at `http://www.math.mcmaster.ca/bprotas/MATH3Q03/template.m` (see also the link in the "Computer Programs" section of the course website); submissions non compliant with this template will not be accepted.
- All plots should have suitable axis labels and legends.
- Make sure to enter your name and student I.D. number in the appropriate section of the template.
- Late submissions and submissions which do not comply with these guidelines will not be accepted.

---

1. You are given five algebraic systems $\mathbb{A}\mathbf{x}_1 = \mathbf{y}_1$, $\mathbb{A}\mathbf{x}_2 = \mathbf{y}_2$, $\mathbb{A}\mathbf{x}_3 = \mathbf{y}_3$, $\mathbb{A}\mathbf{x}_4 = \mathbf{y}_4$, and $\mathbb{A}\mathbf{x}_5 = \mathbf{y}_5$, with the same matrix

$$\mathbb{A} = \begin{bmatrix} 6 & 1 & 0 & 3 \\ 5 & 8 & 4 & 0 \\ 2 & 2 & 7 & 1 \\ 0 & 1 & 2 & 4 \end{bmatrix} \tag{1}$$

and different right–hand side (RHS) vectors, such that $\mathbf{y}_1 = [4 \;\; 1 \;\; 9 \;\; 2]^T$ and the remaining RHS vectors are defied as $[\mathbf{y}_2]_k = ([\mathbf{y}_1]_k)^2$, $[\mathbf{y}_3]_k = ([\mathbf{y}_1]_k)^3$, $[\mathbf{y}_4]_k = ([\mathbf{y}_1]_k)^{-1}$ and $[\mathbf{y}_5]_k = ([\mathbf{y}_1]_k)^{1/2}$, where $k = 1, \dots, 4$.

   (a) using MATLAB function `lu` perform LU decomposition of the matrix $\mathbb{A}$; print out the resulting matrices $\mathbb{L}$ and $\mathbb{U}$ (note that, due to the specific structure of the matrix $\mathbb{A}$, the matrix $\mathbb{L}$ returned by the function `lu` should indeed be lower triangular),

   (b) Write your own two functions `Lbck` and `Ubck` that solve an algebraic system with, respectively, lower and upper triangular matrix using back substitution; use these functions to solve the above five system in two steps, i.e., first solve $\mathbb{L}\mathbf{z}_i = \mathbf{y}_i$ and then $\mathbb{U}\mathbf{x}_i = \mathbf{z}_i$, $i = 1, \dots, 5$ (do not use the operator "\"!),

   (c) print out in a single row the first elements of the five solutions vectors $\mathbf{x}_1, \dots, \mathbf{x}_5$.

   (4 points)

2. Write a MATLAB code which will perform the following calculations:

   (a) will construct the Hilbert matrix with entries $A_{ij} = \frac{1}{i+j-1}$, $i, j = 1, \dots, N$ for any given dimension N,

   (b) calculate the norms $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$ and $\|A\|_F$, where $\|\cdot\|_F$ denotes the Frobenius "norm", for any given square matrix A (do not use the MATLAB function `norm`, but write your own function instead; your can use the function `norm` to check if your results are correct),

(c) calculate and write out all of the above norms for the Hilbert matrix with $N = \{25, 50, 75, 100, \dots, 500\}$; plot these results on a single graph (using linear coordinates and different line colors for different norms); this plot should appear as `figure(1)`.

(3 points)

3. Consider an accelerated version of the Gauß-Seidel iterative method, known as the *Successive Over-Relaxation (SOR)*. Given a number $\omega \in (0, 2)$ and the algebraic system

$$\mathbb{A}\mathbf{x} = (\mathbb{L} + \mathbb{D} + \mathbb{U})\mathbf{x} = \mathbf{b},$$

where $\mathbb{L}$, $\mathbb{D}$ and $\mathbb{U}$ are, respectively, the strictly lower-triangular, diagonal and upper-triangular matrices, we can rewrite it in the fixed-point form as

$$(\omega\mathbb{D} + \omega\mathbb{L})\mathbf{x} = -\omega\mathbb{U}\mathbf{x} + \omega\mathbf{b}. \tag{2}$$

After simple manipulations, relation (2) can be transformed to the following iterative algorithm

$$\mathbf{x}^{(n+1)} = (\mathbb{D} + \omega\mathbb{L})^{-1}\left\{-\left[\omega\mathbb{U} + (\omega - 1)\mathbb{D}\right]\mathbf{x}^{(n)} + \omega\mathbf{b}\right\}, \quad n = 1, 2, \dots, \tag{3}$$

where n denotes the iteration count. Use approach (3) to solve the algebraic problem (with the "poisson" matrix) defined in the function `compare_it.m` (posted as a part of "Iterative Solvers (I)" on the course website). In the computations use $N = 10$ (so that the system of linear equations has $N^2 = 100$ unknowns) and the tolerance of 10^{-6}. Obtain solutions for $M = 21$ values of the parameter $\omega = 0.0, 0.1, \dots, 2.0$ and plot the number of iterations required to solve the problem with the prescribed tolerance as a function of the parameter ω. This plot should appear as `figure(2)`. Limit the number of iterations to 1000 (i.e., if for a given value of ω more iterations are required to solve the problem, then plot 1000).
(3 points)