First Name: _____   **Last Name:** _____

Homework 7

Exercise 1   100% (MATLAB)

This homework involves developing simple matlab functions for implementing Richardson's iteration for computing the solution of a linear system,

$$Ax = b, \tag{1}$$

where A is a nonsingular $n \times n$ matrix and $x, b \in \mathbb{R}^n$ with b given.

Richardson's method is a fixed point iteration with a parameter $\alpha \in \mathbb{R}$. Note that (1) can be rewritten $0 = b - Ax$. Richardson's method comes from the fixed point formulation,

$$x = x + \alpha(b - Ax)$$

and is given by

$$x^{(j+1)} = x^{(j)} + \alpha(b - Ax^{(j)}).$$

This method requires an initial iterate x_0. Without *a priori* information, $x^{(0)} = 0 \in \mathbb{R}^n$ is often used.

Write a MATLAB m-file function implementing Richardson's iteration with interface:

```
function [x,k]=Richardson(x0,L,A,b,a,tol).
```

Here $x0$, A, b and a stands for $x^{(0)}$, A, b and α. The positive real number *tol* is the desired tolerance and L is the maximum number of iterations. The outputs $[x, k]$ are the final iterate and the corresponding iteration number.

We shall measure the error in a weighted norm, specifically, if $x \in \mathbb{R}^n$,

$$\|x\| = \left(n^{-1} \sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

so that the algorithm would stop when

$$\|x^{(i+1)} - x^{(i)}\| \leqslant tol.$$

MATLAB provides a function for computing the Euclidean norm of a matrix, namely, if M is an $n \times m$ (MATLAB) matrix, the function

```
norm(M)
```

computes

$$norm(M) := \left(\sum_{i=1}^{n} \sum_{j=1}^{m} M(i,j)^2 \right)^{1/2}.$$

Thus, if P is the $n \times 1$ matrix representing the vector $p \in \mathbb{R}^n$, we can compute $\|p\|$ by

`norm(P)/sqrt(n).`

The remainder of the assignment involves running your Richardson's function on various matrices. In all cases, use a right hand side consisting of all ones and an initial iterate consisting of all zeros. There are three possible outcomes of the iteration:

(a) The iterations converge to the unique solution of $Ax = b$, i.e. reached the tolerance $tol = 10^{-5}$ within k iterations (take L large enough).

(b) The iteration diverges (iterates becomes arbitrarily large).

(c) The iteration remains bounded but does not converge.

1. A is a 3×3 diagonal matrix with diagonal entries $1, 2, 4$. Run your code for $\alpha = \{-1, 1, 1/2, 1/4, 2/5\}$ and report whether (a), (b) or (c) holds for each iteration. When (a) holds, report the number of the iteration when the error indicator reaches the tolerance.

2. A is a 3×3 diagonal matrix with diagonal entries $1, 2, -4$. Run your code for $\alpha = \{-2, -1, 1, 2\}$ and report whether (a), (b) or (c) holds for each iteration.

3. For
$$A = \begin{pmatrix} 2 & 10 \\ 0 & 4 \end{pmatrix},$$

experiment with different values of α until you have found two which lead to divergence and two which lead to convergence. In the convergent cases, report the iteration number when the error indicator reaches the tolerance.

Hand in the results for the four problems and a copy of the matlab code implementing the iteration.