Name and section: ____________________

ID number: ____________________

E-mail: ____________________

1. (35 points) Consider the 2-point BVP

$$
\begin{aligned}
&u'' - (4x^2+2)u + 2x(1+2x^2) = 0, \quad x \in (0,1),\\
&u(0) = 1, \quad u(1) = 1+e.
\end{aligned} \tag{1}
$$

   (a) (5 points) Show that $u(x) = x + e^{x^2}$ is the **exact solution**.

   (b) (20 points) Employ a second-order accurate finite difference scheme for Eq. (1) with $h = 0.05$ and set-up the linear system $A\mathbf{u} = \mathbf{f}$. Subsequently, use the three basic iterative schemes, namely, **Jacobi**, **Gauss-Seidel** and **Successive-over-relaxation** (SOR) methods to solve it. As per the SOR use $\omega = 1.6$. Furthermore, use

$$\mathbf{u}_0 = [0\,0\,\ldots\,0]^T$$

   as your initial guess and stop the iterations when

$$||\mathbf{r}_k||_2 \leq \text{tol}\,||\mathbf{f}||_2$$

   holds with tol $= 10^{-6}$. For each method, make a semilog plot of the 2-norm of the residual vector $\mathbf{r}_k$, i.e., $||\mathbf{r}_k||_2$ against k in **the same** figure and state **which is which** by including a legend. Finally, and as per a **sanity check**, plot your numerical solution (e.g., obtained via SOR) and exact solution in **the same** figure and state **which is which** again. Attach all your figures.

   Remark: You do have all the MATLAB codes employing the above basic iterative schemes!

   (c) (10 points) Recall that the iteration matrix for SOR is given by

$$R_{SOR} = (\omega L + D)^{-1}\left[(1-\omega)\,D - \omega U\right].$$

   If $\omega \in [0.5, 2]$, plot the **spectral radius** of R_{SOR}, i.e., $\rho(R_{SOR})$ as a function of ω. What is the optimal value of ω_{opt}?
   Include your code and the figure.

2. (30 points) Assume an $m \times m$ **diagonal** matrix A whose (distinct) eigenvalues λ_i, $i = 1, 2, \ldots, m$, are distributed in the range $[10^{-p}, 1]$ with $p = 1, 2, 3$. Since $\lambda_i > 0$,

$\forall i = 1, 2, \ldots, m$, the matrix A is a **symmetric positive definite** (SPD) matrix. Subsequently, consider the following two distributions for the eigenvalues

$$\lambda_i = 10^{-p} + i\frac{1-10^{-p}}{m-1}, \tag{2}$$

$$\lambda_i = 10^{-p} + \left(1-10^{-p}\right)\cos\left[\frac{i\pi}{2\,(m-1)}\right]. \tag{3}$$

Then, set $m = 100$ and for each p:

- Construct the diagonal matrix A based on the above two distributions.
- Use both the **Steepest Descent** and **Conjugate gradient methods** in order to solve the linear system $A\mathbf{u} = \mathbf{f}$ with `f=rand(m,1)` and initial iterate `u0=rand(m,1)`. Stop the iterations in both methods if

  $$\mathbf{r}_k^T\mathbf{r}_k \leq \mathrm{tol}^2\ \mathbf{f}^T\mathbf{f}$$

  holds with $\mathrm{tol} = 10^{-10}$. Plot $||\mathbf{r}_k||_2$ as a function of k for both methods (in separate figures) and for all cases of p. Which method converges faster? Compare the two cases based on the different distributions of the eigenvalues.

Attach **all** your codes and figures.

Remark: The MATLAB functions `my_std.m` and `my_cg.m` are given!

3. (35 points) Consider again the Poisson problem

$$\nabla^2 u(x,y) = f(x,y), \quad (x,y) \in \Omega = (0,1)^2,$$

where

$$f(x,y) = 2x\,(y-1)\,(y-2x+xy+2)\exp\,(x-y),$$

and exact solution

$$u^{\text{exact}}(x,y) = \exp\,(x-y)x\,(1-x)\,y\,(1-y)\,.$$

Furthermore, consider **Dirichlet** boundary conditions at $\partial\Omega$.

Recall that in HW4 the underlying (pentadiagonal) linear system $A\mathbf{u} = \mathbf{f}$ was solved by using a **direct** method where we considered the five-point Laplacian. In the present case, solve the linear system based again on the five-point Laplacian with $\Delta x = \Delta y \equiv h$ and $m = 49$ by employing the **Conjugate Gradient method** (CG) *without* preconditioning and the **Preconditioned Conjugate Gradient method** (PCG). As far as the **preconditioners** are concerned, consider the Gauss-Seidel and *symmetric* SOR (SSOR) preconditioners:

$$M_{GS} = \left(\mathbb{I} + LD^{-1}\right)(D+U)\,,$$

$$M_{SSOR} = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}D + L\right)D^{-1}\left(\frac{1}{\omega}D + U\right),$$

respectively, and use a value of $\omega = 1.75$ in SSOR. Use 10^{-13} as `tol` in the `my_cg.m` and `my_pcg.m` MATLAB functions provided and consider $\mathbf{u}_0 = [0\,0\,\ldots\,0]^T$ as your initial guess.

Compare/plot (in a semilog scale) the $||\mathbf{r}_k||_2$ of these three runs against k.

Attach your figure and **any** codes/drivers.