Solution 10 – MATH-250 Advanced Numerical Analysis I

(⋆) Problem 4.

Let $A, P \in \mathbb{R}^{n\times n}$ be symmetric and positive definite matrices. Consider the linear system

$$A\mathbf{x} = \mathbf{b} \tag{1}$$

(a) We denote the Cholesky factorisation $P = LL^T$, where L is a lower triangular matrix.

Derive a relation between the solution x of (1) and the solution $\tilde{x}$ to

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{2}$$

where $\tilde{A} = L^{-1}AL^{-T}$ and $\tilde{\mathbf{b}} = L^{-1}\mathbf{b}$.

(b) Apply the gradient method to the linear system (2) and show that it is equivalent to an iterative method given by the update

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k P^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}). \tag{3}$$

Starting from the expression for α_k given in the lecture notes, derive an expression for α_k that only involves A and P^{-1}. In particular, it should not involve $\tilde{A}$ or the Cholesky factor L.

(c) Define $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ the residual after the k-th iteration.

Show that

$$\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k+1)} \rangle_{P^{-1}} = 0, \quad k \geq 1$$

where $\langle \mathbf{y}, \mathbf{z} \rangle_{P^{-1}} = \mathbf{y}^\top P^{-1}\mathbf{z}$.

(d) The method (3) is called the preconditioned gradient method.

Write a Python function `gradient(A, b, P)` that implements the preconditioned gradient method for a matrix A, a vector $\mathbf{b}$, and a preconditioning matrix P. Stop the iteration once the relative error $\frac{\|\mathbf{r}^{(k)}\|_2}{\|\mathbf{b}\|_2}$ is smaller than 10^{-6}, and return the solution $\mathbf{x}^{(k)}$, the residual norms $\|\mathbf{r}^{(1)}\|_2, \|\mathbf{r}^{(2)}\|_2, \ldots, \|\mathbf{r}^{(k)}\|_2$ and the number of iterations k executed to reach the solution. Ensure that if no preconditioner is given in the function arguments then the unpreconditioned gradient method is run.

(e) Run the gradient method for the system given by

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ -1 \\ 2 \end{pmatrix}$$

without any preconditioning. Clearly print the number of iterations.

(f) On Moodle we provide a matrix $A \in \mathbb{R}^{n\times n}$ in the file `matrix10.npz`. Load this matrix using SciPy `sparse`'s `load_npz` (or `scipy.sparse.load_npz` if you are not using our provided Jupyter notebook), and define the right-hand side $\mathbf{b} = [1, 1, \ldots, 1]^\top$ of appropriate size.

Run the preconditioned gradient method with the preconditioners

- $P_1 = I_{n\times n}$,
- $P_2 = \mathrm{diag}(a_{11}, a_{22}, \dots, a_{nn})$, and
- $P_3 = LU$.

Plot the residual norms $\|\mathbf{r}^{(i)}\|_2$ for increasing numbers of iterations for the preconditioners P_1, P_2, and P_3. Use a single plot for all three preconditioners.

Hint: Use `sps.linalg.spilu` to compute the incomplete LU factorisation of the sparse matrix A (use `scipy.sparse.linalg.spilu` if you do not want to use the notebooks on Moodle). This method returns a `SuperLU` object, meaning you can directly call its member function `solve` on a matrix M to compute $P^{-1}M$.

Solution.

(a) We know that $L^{-1}AL^{-\top}\tilde{\mathbf{x}} = L^{-1}\mathbf{b}$ and by assuming that (1) is solved exactly $A\mathbf{x} = \mathbf{b}$. Multiplying by L from the left we see that $AL^{-\top}\tilde{\mathbf{x}} = \mathbf{b}$ and hence $\mathbf{x} = L^{-\top}\tilde{\mathbf{x}}$.

(b) We begin by applying the gradient method to (2).

$$\tilde{\mathbf{x}}^{(k+1)} = \tilde{\mathbf{x}}^{(k)} - \alpha_k(\tilde{\mathbf{b}} - \tilde{A}\tilde{\mathbf{x}}^{(k)}) \qquad \iff \tag{4}$$
$$L^\top\mathbf{x}^{(k+1)} = L^\top\mathbf{x}^{(k)} + \alpha_k L^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}) \qquad \iff \tag{5}$$
$$P\mathbf{x}^{(k+1)} = P\mathbf{x}^{(k)} + \alpha_k(\mathbf{b} - A\mathbf{x}^{(k)}) \qquad \iff \tag{6}$$
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k P^{-1}(\mathbf{b} - A\mathbf{x}),$$

where in (4) we used the results from (a), in (5) we multiplied the entire system by L from the left and defined $P = LL^\top$, and in (6) we applied P^{-1} from the left. Next, we use

$$\alpha_k = \frac{\langle \tilde{\mathbf{r}}^{(k)}, \tilde{\mathbf{r}}^{(k)}\rangle}{\langle \tilde{A}\tilde{\mathbf{r}}^{(k)}, \tilde{\mathbf{r}}^{(k)}\rangle} \tag{7}$$

from the lecture notes for the application of the gradient method to $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$. We simplify (7) to further see that

$$\alpha_k = \frac{\langle L^{-1}\mathbf{r}^{(k)}, L^{-1}\mathbf{r}^{(k)}, \rangle}{\langle L^{-1}\mathbf{r}^{(k)}, L^{-1}AP^{-1}\mathbf{r}^{(k)}\rangle} = \frac{\langle \mathbf{r}^{(k)}, P^{-1}\mathbf{r}^{(k)}\rangle}{\langle \mathbf{r}^{(k)}, P^{-1}AP^{-1}\mathbf{r}^{(k)}\rangle},$$

where we used that in analogy to (a) it holds that $\tilde{r}^{(k)} = L^{-1}\mathbf{r}^{(k)}$ for $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$.

(c) In the lecture we have seen that $\langle \tilde{\mathbf{r}}^{(k+1)}, \tilde{\mathbf{r}}^{(k)}\rangle = 0$. Applying (b) we then see

$$0 = \langle L^{-1}\mathbf{r}^{(k+1)}, L^{-1}\mathbf{r}^{(k)}\rangle = \langle \mathbf{r}^{(k+1)}, \mathbf{r}^{(k)}\rangle_{P^{-1}}.$$

(d – f) The solutions can be found in the Jupyter notebooks provided on Moodle.