MA398 MATRIX ANALYSIS AND ALGORITHMS: ASSIGNMENT 3

Please submit your solutions to this assignment via Moodle by **noon on Thursday December 10th**. Make sure that your submission is clearly marked with your name, university number, course and year of study.

- The written part of the solutions may be typed in LaTeX, or written on paper and subsequently scanned/ photographed provided the images are clearly legible. The result should be a single multi-page document entitled *MA398_Assignment3_FirstnameLastname.pdf*.
- The Matlab code scripts relevant to each question should be submitted as *MA398_ Assignment3_ ExerciseN.m*, where you should be careful to implement any functions having the same name, input and output format as indicated in the questions below.

⚠Only in an emergency or if the Moodle submission is unavailable because of a general outage, the assignment should in the respective case be submitted by email to radu.cimpeanu@warwick.ac.uk and n.shkeir@warwick.ac.uk.

---

1. (Jacobi and SOR methods). Given a matrix $A \in \mathbb{R}^{n\times n}$, let the matrices L, D and U be such that $A = L + D + U$, with L strictly lower triangular, D diagonal and U strictly upper triangular. Then one step of the Jacobi method is given by
$$x_k = D^{-1}(b - (L+U)x_{k-1}), \tag{1}$$
and one step of the successive-over-relaxation (SOR) method with relaxation parameter $\omega \in \mathbb{R}$ is given by
$$x_k = (L+\omega D)^{-1}(b - (U + (1-\omega)D)x_{k-1}). \tag{2}$$
Note that alternative formulations (based on different prefactors) exist in the literature. These will shift the weights of the various terms (and affect some of the algebra, such as in the hint of point a below), but ultimately they are equivalent if set up correctly. You are welcome to use any of these variants as long as you are consistent. The special case $\omega = 1$ corresponds to the Gauss-Seidel method.

   By Theorem 20.1 in the lecture notes (Lecture 20), the Jacobi method applied to A converges if A satisfies the strong row sum criterion.

   (a) Prove that the Gauss-Seidel method also converges if A satisfies the strong row sum criterion.
   Hint: Show that the equation $\det(R - \lambda I) = 0$ cannot be satisfied for $|\lambda| \geq 1$, where $R = -(L + D)^{-1}U$.
   Remark: It can furthermore be shown that the SOR method applied to A converges if A is symmetric positive definite and $0 < \omega \leq 2$.

   (b) Derive an explicit formula for the iteration step given in equation (2). Your answer must be of the form
   $$x_i^{(k+1)} = \textit{[your formula]} \qquad \text{for } i = 1, \dots, n$$
   with only elementary arithmetic operations ($+, -, \times, \div, \sqrt{\cdot}$, etc.) as well as the entries of $A, b, x^{(k)}$ and ω appearing on the right hand side.

   (c) **(Bonus)** For a fixed $k \in \mathbb{N}$, why would you expect the approximation x_k produced by the Gauss-Seidel method to be more accurate than the approximation x_k produced by the Jacobi method? Assume that both methods converge and use the same starting guess x_0.

   (d) Write a function `sor(A, b, n, w)` in Matlab which computes the iterates for the SOR method up to the `n`th iteration and returns a vector containing the ∞-norm of the residual at each iteration. You may base your code on the following, which computes the solution for the Jacobi method, but can also modify and improve it according to your needs. These helper functions (for both Jacobi and SOR) should form part of your final assignment submission.

```
function r = jacobi(A, b, n)
    D = diag(diag(A));
    L = tril(A)-D;
    U = triu(A)-D;
    r = zeros(n,1);
    x = zeros(size(b));
    for k = 1:n
        x = D\(b - (U+L)*x);
        r(k) = norm(b - A*x, Inf);
    end
end
```

(e) Let $A \in \mathbb{R}^{n\times n}$ be the tridiagonal matrix with diagonal entries equal to the value 4.1 and off-diagonal entries equal to 2.0. Let $b \in \mathbb{R}^n$ have every entry equal to 1.0, and let the starting guess $x_0 \in \mathbb{R}^n$ have every entry equal to 0. Denote by x the solution to $Ax = b$, by $e_k = x - x_k$ the error at the k^{th} step and by $r_k = b - Ax_k$ the residual at the k^{th} step.

i. Show that $\|A\|_\infty$ and $\|b\|_\infty$ are uniformly bounded in n.

ii. Confirm numerically that $\|e_0\|_\infty$ is bounded uniformly in n. More precisely, compute the true solution x through the backslash operator $x = A\backslash b$ for n from 1 to 100 and provide a plot showing that $\|e_0\|_\infty$ is bounded.

iii. Plot the number of iterations required to satisfy $\frac{\|r_k\|_\infty}{\|b\|_\infty} \leq 10^{-9}$ for the Jacobi and Gauss-Seidel methods, for a range of values of n varying from 128 to 4096. What does the behaviour of the required number of iterations suggest about $\|R\|_\infty$, where $R \in \mathbb{R}^{n\times n}$ is such that $e_k = R^k e_0$?

iv. Fix $n = 256$. Find the number of iterations required to satisfy $\frac{\|r_k\|_\infty}{\|b\|_\infty} \leq 10^{-9}$ for the Jacobi and SOR methods with $\omega = 0.2, 0.4, 0.6, ..., 2$. Which methods require the largest and smallest number of iterations?

v. Fix $n = 256$. Find the number of iterations required to satisfy $\frac{\|r_k\|_\infty}{\|b\|_\infty} \leq \varepsilon$ for $\varepsilon = 10^{-1}, 10^{-2}, 10^{-4}, 10^{-8}$ for the Jacobi method and and the SOR method with $\omega = 0.6$. Is the relationship between ε and the number of iterations as you expected?

2. (Nonlinear iterative methods in practice) Let A be the 100×100 tridiagonal symmetric matrix with $1, 2, \ldots, 100$ on the diagonal and constant entries 1 just above and below the diagonal, respectively. Also consider the right hand side to be given by $(1, 1, \ldots, 1)^T$. Write a Matlab script that computes 100 steps of the steepest descent (SD) and conjugate gradient (CG) algorithms to approximately solve $Ax = b$. Analyse your results by adding, to the same plot:

- the computed residual norms $||r^{(k)}||_2$ for the SD method,
- the condition number-based bound on $||e^{(k)}||_A/||e^{(0)}||_A$ for SD highlighted in Theorem 25.2 (Week 9, Lecture 25),
- the computed residual norms $||r^{(k)}||_2$ for the CG algorithm,
- the condition number-based bound on $||e^{(k)}||_A/||e^{(0)}||_A$ for CG highlighted in Theorem 25.4 (Week 9, Lecture 25),

for $k = 1, 2, \ldots, 100$. Briefly comment on your results.

Remarks: you are welcome to use the in-built CG solver in Matlab, but should write your own helper function for the steepest descent method based on Algorithm 10 in the notes (Week 8, Lecture 22).

3. (Tridiagonal Toeplitz matrices and eigenvalues) Continuing on the same theme, in this problem we will consider a particular class of matrices with a diagonal structure, which will turn out to be a property making eigenvalue and eigenvector calculation analytically tractable.

(a) Let $A \in \mathbb{R}^{m\times m}$ have constant entries d along the diagonal, a immediately above and below the diagonal, and 0 elsewhere. Show that the eigenvalues of A are $\{\lambda_k\}_{k=1}^m$ given by

$$\lambda_k = d + 2a\cos\left(\frac{k\pi}{m+1}\right),$$

for $k = 1, \ldots, m$. Furthermore, show that the corresponding eigenvector $x = [x_1, \ldots, x_m]^T$ has the j^{th}-component given by

$$x_j = \sin\left(\frac{jk\pi}{m+1}\right).$$

(b) How do the above results (for both eigenvalues and eigenvectors) extend to the case in which the matrix has different constant values above and below the diagonal? You should identify generalisations of the formulae when considering entries just below the diagonal as being $b \neq a$.

(**Unmarked extra reading**) If you are interested in some of the real-world applications of the mathematics covered in this module, then this article on the 25 000 000 000$ eigenvector (and Google's PageRank algorithm) should be an entertaining read.