TW776: Assignment 3 — For Friday, 18 March

Assignment Guidelines

- Write a short report to illustrate your work. Use full sentences and include code snippets where applicable.
- Reports are to be handed in during the lecture on the due date.
- Feel free to discuss the work with your classmates, but write your own computer codes, generate your own figures, devise your own proofs and examples, and write your own report.
- It is acceptable to consult the library or internet on any problem. But if you do, it is important that you rewrite the material in your own words and cite your source.
- Unless told otherwise, use built-in functions in MATLAB or Python as far as possible. Do not write your own. This is not a course in programming.

Cholesky Factorization

Do Problem 9, p. 97, in Ascher & Greif, by using the Cholesky factorization (hand calculation).

Instability in Gaussian Elimination

Let A be Wilkinson's "pathological matrix" of order $n \times n$, as defined in Example 5.9, p. 118, Ascher & Greif. In this exercise we consider the solution of $A\boldsymbol{x} = \boldsymbol{b}$, using Gaussian Elimination with two pivoting strategies. The object of the exercise is to confirm that GE with partial pivoting can be unstable, while GE with complete pivoting is stable. You should not forget, however, that this is a rare occurrence and in practice GEPP is considered stable. (In the next assignment we shall consider an alternative to complete pivoting, namely the QR-factorization, so store the codes you develop for this exercise in a safe place.)

To have an exact solution available, we first choose an $\boldsymbol{x}$, and then compute the corresponding right-hand side. In this case, use $\boldsymbol{x} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n})^T$, and then compute $\boldsymbol{b}$ as $A\boldsymbol{x}$.

(a) Write a code that assembles A for any order n, as well as the right-hand side $\boldsymbol{b}$ as defined above.

(b) Using GEPP (i.e., $PA = LU$) solve the system $A\boldsymbol{x} = \boldsymbol{b}$ and calculate the ℓ_∞-error in the computed values of $\boldsymbol{x}$. Use $n = 10, 20, 30, 40, 50, 60$ and list the errors in a table, as well as in a log-linear plot (error vs n). *Important:* Use the built-in LU decomposition in MATLAB (`lu`) or Python (`scipy.linalg.lu`). Likewise, the forward and back substitutions should be done by built-in codes; in MATLAB use backslash (it knows what to do with triangular systems), and in Python `scipy.linalg.solve_triangular`.

(c) In Example 5.9 (and in class) the growth factor of this matrix was shown to be $g_n(A) = 2^{n-1}$. Add the curve $\epsilon 2^{n-1}$ to your log-linear plot of part (b), where $\epsilon = 2^{-52} \approx 2.2 \times 10^{-16}$ is the machine epsilon of IEEE arithmetic. Discuss what you see.

(d) Repeat part (b) for GECP (i.e., $PAQ = LU$). Draw conclusions regarding the stability of GECP vs GEPP. *Note:* There is no built-in code in MATLAB for LU decomposition with complete pivoting, but a code downloaded from the MATLAB file-exchange will be put on the class web site. Likewise, a Python code that does GECP will be put on the web site.

Re-ordering Strategies

(a) Let A be the $N^2 \times N^2$ matrix defined on p. 183 in Ascher & Greif (i.e., the five point discretization of Laplace equation in the natural ordering). Now generate figures similar to Figures 5.6 and 5.7 on p. 136. Use $N = 8$. In MATLAB, use the built-in functions for computing the RCM and AMD orderings. In Python, AMD ordering is available in `scipy.sparse.linalg.splu`. If you cannot find a Python code for RCM ordering, it is ok to substitute here and below the RCM ordering with red-black ordering; the code for the latter is easy enough to generate yourself.

(b) By increasing the value of N, create a table that lists the number of nonzeros in the Cholesky factors of PAP^T, for (i) the natural ordering, (ii) the RCM ordering and (iii) the AMD ordering. Which ordering seems most efficient?

(c) Create a table similar to the one in part (b), but instead of listing the number of nonzeros, list the actual execution times for solving $A\boldsymbol{x} = \boldsymbol{b}$. Here $\boldsymbol{b}$ is an arbitrary vector of the appropriate dimension. You only need to record the execution times of the forward and back substitution step, as summarized by

$$LL^T\boldsymbol{x} = \boldsymbol{b} \quad \Longrightarrow \quad L\boldsymbol{y} = \boldsymbol{b},\ L^T\boldsymbol{x} = \boldsymbol{y}$$

Use built-in codes for sparse forward and back substitution. In MATLAB, execution times can be obtained by using the `tic; toc` commmands; type `help tic` to learn more. In Python use `time.time()`, or `timeit` if you're using IPython. It is advisable to run the code several times and take the minimum over the execution times.

Ill-conditioning

Consider the infamous Hilbert matrix, the 5×5 version of which looks like

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

If we choose $\boldsymbol{b}$ equal to the last column of H, then the solution of the system $H\boldsymbol{x} = \boldsymbol{b}$ is $\boldsymbol{x} = (0, 0, \ldots, 0, 1)^T$.

In MATLAB there is a built-in function `hilb` for generating the Hilbert matrix. In Python, use the following code snippet:

```
import numpy as np

def hilbert(N):
    x, y = np.ogrid[1.0:N + 1, 1.0:N + 1]
    return 1 / (x + y - 1)
```

(a) On the internet or in the library, search for "Hilbert matrix", and in a single sentence summarize what you have learned about its condition number.

(b) Solve the 5×5 system. You may assume without proof that the matrix is symmetric positive definite, so Cholesky should be the appropriate algorithm. (Use built-in functions!) Display the computed solution $\boldsymbol{x}$ to all digits. How many of these digits are correct?

(c) Repeat for the 10×10 case. You should find that many digits are incorrect. This is due to the extreme ill-conditioning of the Hilbert matrix.

(d) Does pivoting improve the results? Solve the system using both partial pivoting and complete pivoting. Did that work? Discuss.

(e) For the 10×10 case, compute the residual $\boldsymbol{r} = \boldsymbol{b} - A\widetilde{\boldsymbol{x}}$, where $\widetilde{\boldsymbol{x}}$ is the solution of part (c). Does a small residual guarantee a good approximation? Discuss with reference to inequality (5.1), p. 138, in Ascher & Greif; compute numerical values for all factors in that formula and confirm that the inequality is satisfied.