Petunjuk:

Pada Tugas Kelompok ini, disediakan tiga pilihan topik. Di setiap kelas, setiap topik hanya boleh dikerjakan paling banyak 7 kelompok. Pilih salah satu topik saja, daftarkan di SceLE, dan kerjakan sebaik-baiknya.

Untuk mendapatkan nilai ≥ 8.5, Anda harus mengeksplor lebih jauh topik yang kelompok Anda pilih (tidak hanya menjawab pertanyaan yang diberikan di deskripsi tugas) dan melaporkan hasil eksplorasi kelompok Anda. Silakan berkonsultasi dengan asisten dosen unuk kemungkinan hal yang dieksplorasi. Lihat file Informasi Tugas Kelompok untuk melihat komponen penilaian.

TOPIK 1

Matriks Hilbert H berukuran $n \times n$ memiliki entri $h_{ij} = 1/(i + j - 1)$, sehingga bentuknya adalah

$$\begin{pmatrix} 1/1 & 1/2 & 1/3 & \cdots \\ 1/2 & 1/3 & 1/4 & \cdots \\ 1/3 & 1/4 & 1/5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Untuk $n = 2,3,4, \ldots$ buatlah matriks Hilbert dengan ukuran $n \times n$ dan juga buat vektor berukuran $n \times 1$ yang Anda sebut $b = Hx$ di mana $x = [1\ 1\ 1 \ldots 1]^{\mathrm{T}}$. Gunakan Gaussian Elimination atau faktorisasi Cholesky (periksa terlebih dahulu apakah matriks Hilbert merupakan matriks positif definit dan simetris) untuk menyelesaikan sistem persamaan linear $Hx = b$, menghasilkan solusi aproksimasi $\hat{x}$. Hitung norm-∞ dari residu $r = b - H\hat{x}$ dan juga error $\Delta x = \hat{x} - x$, di mana $x = [1\ 1\ 1 \ldots 1]^{\mathrm{T}}$. Berapakah n terbesar sebelum erornya 100% (tidak ada angka penting lagi dalam komputasi Anda)? Selanjutnya, aproksimasilah $\mathrm{cond}(H)$ untuk setiap nilai n dan coba tuliskan sebagai fungsi dalam n. Untuk n yang bervariasi, bagaimana hubungan Anda condition number matriks tersebut dengan banyaknya digit yang akurat dalam komponen-komponen solusi komputasi?

TOPIK 2

Implementasi LU Factorization tanpa pivoting, dengan *partial pivoting*, dan dengan *complete pivoting*.

Buatlah beberapa sistem persamaan linear dengan matriks-matriks yang *random* (gunakan *random number generator* untuk mendapatkan entri matriks) dan buat sisi kanannya agar solusi eksaknya Anda ketahui. Gunakan program hasil implementasi Anda di atas untuk menyelesaikan sistem-sistem ini dan bandingkan masing-masing program dalam hal akurasi, residual, dan juga performa dari ketiga implementasi.

Apakah Anda dapat merancang matriks (yang tidak *random*) agar hasil *complete pivoting* lebih akurat secara signifikan dibandingkan *partial pivoting*?

TOPIK 3

Buatlah fungsi dan / atau script MATLAB / Octave untuk menyelesaikan sistem persamaan linear tridiagonal berdasarkan algoritma yang diberikan di bawah dan uji dengan menggunakan beberapa sistem persamaan linear yang diacak. Jelaskan bagaimana implementasi Anda berubah jika Anda juga ingin mengimplementasikan partial pivoting. Jelaskan bagaimana implementasi Anda akan berubah jika sistem persamaan linearnya juga positif definit dan Anda menghitung faktorisasi Cholesky daripada faktorisasi LU.

Algorithm 2.8 Tridiagonal LU Factorization

$d_1 = b_1$
for $i = 2$ **to** n { loop over columns }
  $m_i = a_i / d_{i-1}$ { compute multiplier }
  $d_i = b_i - m_i c_{i-1}$ { apply transformation }
end