# Workout — Iterative Methods

## Non-mandatory exercises

4. Repeat Exercise 1 for Gauss-Seidel's method.

   Pseudo-code for the Gauss-Seidel method:

   set $u^{(0)}$
   **for** $i = 1$ to *maxiter* **do**
   $u^{(i)}(1) = \frac{1}{2}u^{i-1}(2) - \frac{1}{2}b(1)$
   **for** $n = 2$ to $N-1$ **do**
   $u^{(i)}(n) = \frac{1}{2}\left(u^{(i)}(n-1) + u^{(i-1)}(n+1)\right) - \frac{1}{2}b(n)$
   **end for**
   $u^{(i)}(N) = \frac{1}{2}u^{i}(N-1) - \frac{1}{2}b(N)$
   **end for**

   Iteration 1:

$$
\begin{aligned}
u^{(1)}(1) &= \frac{1}{2}u^{(0)}(2) - \frac{1}{2}b(1) = -\frac{\alpha}{2} \\
u^{(1)}(2) &= \frac{1}{2}\left(u^{(1)}(1) + u^{(0)}(3)\right) - \frac{1}{2}b(2) = -\frac{\alpha}{4} \\
u^{(1)}(3) &= \frac{1}{2}u^{(1)}(3) - \frac{1}{2}b(3) = -\frac{\alpha+4\beta}{8} \\
&\Rightarrow u^{(1)} = \begin{pmatrix} -\frac{\alpha}{2} & -\frac{\alpha}{4} & -\frac{\alpha+4\beta}{8} \end{pmatrix}^T
\end{aligned}
$$

   Iteration 2:

$$
\begin{aligned}
u^{(2)}(1) &= \frac{1}{2}u^{(1)}(2) - \frac{1}{2}b(1) = -\frac{5\alpha}{8} \\
u^{(2)}(2) &= \frac{1}{2}\left(u^{(2)}(1) + u^{(1)}(3)\right) - \frac{1}{2}b(2) = -\frac{3\alpha+2\beta}{8} \\
u^{(2)}(3) &= \frac{1}{2}u^{(2)}(3) - \frac{1}{2}b(3) = -\frac{3\alpha+10\beta}{16} \\
&\Rightarrow u^{(2)} = \begin{pmatrix} -\frac{5\alpha}{8} & -\frac{3\alpha+2\beta}{8} & -\frac{3\alpha+10\beta}{16} \end{pmatrix}^T
\end{aligned}
$$

Iteration 3:

$$
\begin{aligned}
u^{(3)}(1) &= \frac{1}{2}u^{(2)}(2) - \frac{1}{2}b(1) = -\frac{11\alpha + 2\beta}{16} \\
u^{(3)}(2) &= \frac{1}{2}\left(u^{(3)}(1) + u^{(2)}(3)\right) - \frac{1}{2}b(2) = -\frac{7\alpha + 6\beta}{16} \\
u^{(3)}(3) &= \frac{1}{2}u^{(3)}(3) - \frac{1}{2}b(3) = -\frac{7\alpha + 22\beta}{32} \\
&\Rightarrow u^{(3)} = \begin{pmatrix} -\frac{11\alpha+2\beta}{16} & -\frac{7\alpha+6\beta}{16} & -\frac{7\alpha+22\beta}{32} \end{pmatrix}^T
\end{aligned}
$$

5. Does the stationary iterative method $u^{(k+1)} = Gu^{(k)} + c$ converge for all $u^{(0)}$ if the matrix $G$ has eigenvalues $\lambda = \{\cos(\pi j/(N+1))\}_{j=1}^{N}$?

$$
\begin{aligned}
j \in [1, \dots, N] \Rightarrow 0 < j < N + 1 \iff 0 < \frac{j}{N+1} < 1 \\
\Rightarrow -1 < cos\left(\pi \frac{j}{N+1}\right) < 1 \Rightarrow |\lambda| < 1
\end{aligned}
$$

All eigenvalues are strictly less than 1, so the method converges.

6. A matrix $A$ has 5 non-zero diagonals (including the main diagonal). Due to fill-in, the $L$ and $U$ factors of $A$ each have 201 non-zero diagonals. The matrix $A$ is $N \times N$, where $N = 200\,000$. Roughly how many floating point numbers need to be stored when using

   (a) Jacobi's method?

   The LDU factorization in the Jacobi method doesn't cause fill in, so only the original $\sim 5N$ non-zero elements need to be stored, plus the vectors $b$, $u^{(i)}$ and $u^{(i+1)}$ (although with careful coding, most parts of $u^{(i)}$ and $u^{(i+1)}$ don't need to be stored simultaneously). In total, we need to store $8N = 1.6 \times 10^6$ elements.

   (b) LU factorization?

   For each of the LU matrices, we need to store $\sim 201N$ non-zero elements. We also need to store two vectors of length $N$. In total, that makes $203N = 40.6 \times 10^6$ non-zero elements to be stored.

   You may assume that each diagonal has the same length and you can neglect everything except the matrices involved.

7. For the same matrix as in the previous exercise, let's say that you have decided to use the Power Method to compute its largest magnitude eigenvalue. What will the execution time per iteration be if the average time per floating point operation is $10^{-9}$ seconds?

   Operation count:

$$\left.\begin{array}{lr} \text{matrix-vector multiplication:} & \sim 10N \\ \text{computation of } ||zk||_2 : & \sim 2N \\ \text{normalization of } z_k : & N \\ \text{computation of } \mu_k : & 2N \end{array}\right\} \qquad \text{total: } \sim 15N$$

$$15N \times 10^{-9}s = 3 \times 10^{6} \times 10^{-9}s = 3 \times 10^{-3}s$$

   Each iteration will take approximately 3 ms.