- Scrivere una function xnew = gaussseidel_step(A, b, xold) che esegue *un* passo del metodo di Gauss–Seidel; non creando le matrici M ed N ma utilizzando la formula vista a lezione

$$x^{(new)} = \frac{b_i - \sum_{j<i} A_{ij} x_j^{(new)} - \sum_{j>i} A_{ij} x_j^{(old)}}{A_{ii}}.$$

- Costruendo le matrici A, M, N con Matlab, verificare che per $A = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix}, x^{(old)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ si ha $x^{(new)} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$. Usare questi valori per controllare il buon funzionamento. Verificare, sempre con Matlab, che vale la condizione di convergenza $\rho(H) < 1$.
- Scrivere una function x = gaussseidel(A, b, x0, m) che (chiamando gaussseidel_step) esegue m passi del metodo di Gauss–Seidel. Usarla per verificare numericamente la convergenza del metodo.
- Modificarla in function [x, v] = gaussseidel_errore(A, b, x0, m, z), in modo che restituisca anche gli errori (in norma) rispetto a $z \in \mathbb{R}^n$ dato

$$v = [v_1, v_2, \ldots, v_m] = [\|x^{(1)} - z\|_\infty, \|x^{(2)} - z\|_\infty, \ldots, \|x^{(m)} - z\|_\infty] \in \mathbb{R}^{1\times m}.$$

- Plottare v (con $z = A^{-1}b$ la soluzione esatta) in scala logaritmica, e verificare che $\frac{v_{k+1}}{v_k} \to \rho(H)$, come previsto dalla teoria.