Laboratorio di Calcolo Numerico: Algebra Lineare Numerica

Esercizio per casa

Data una matrice $A \in \mathbb{R}^{n \times n}$, non singolare e con gli elementi diagonali diversi da 0 e un vettore $b \in \mathbb{R}^n$, il metodo di eliminazione gaussiana si basa sull'idea di ridurre il sistema $Ax = b$ ad un sistema equivalente della forma $Ux = c$, con $U \in \mathbb{R}^{n \times n}$ triangolare superiore e $c \in \mathbb{R}^n$ un nuovo vettore.
Questo procedimento è possibile moltiplicando a sinistra per delle matrici che eseguono operazioni elementari sia agli elementi a sinistra che destra dell'uguaglianza e, in tale modo, andando, in vari passaggi, a rendere uguali a 0 gli elementi al di sotto della diagonale.
In questo modo, un passaggio alla volta andremo a creare una sequenza di matrici $A^{(k)}$ (e vettori $b^{(k)}$ di consequenza) tali che gli elementi sotto agli elementi da 1 a $k-1$ della diagonale sono uguali a 0.
In particolare, data la matrice $(A^{(k)}) = a_{ij}^{(k)}$, per generare la matrice $A^{(k+1)}$, le operazioni che dovremmo compiere saranno indicare il moltiplicatore

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \dots, n$$

e porre le componenti della nuova matrice $A^{(k+1)}$ e del nuovo vettore $b^{(k+1)}$ uguali a

$$\begin{aligned} a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \qquad i = k+1, \dots, n, j = k, \dots, n \\ b_i^{(k+1)} &= b_i^{(k)} - m_{ik} b_k^{(k)}, \qquad i = k+1, \dots, n \end{aligned}$$

mantenendo fissi gli elementi di $b^{(k)}$ da 1 a k e gli elementi di $A^{(k)}$ al di sopra della diagonale dalle colonne 1 alla colonna k e, invece, rendendo 0 gli elementi sotto la diagonale di $A^{(k+1)}$ dalla colonna 1 alla colonna k.
Quindi si procede con queste iterate fino ad ottenere $A^{(n)} = U$ triangolare superiore e $b^{(n)} = c$ il nuovo termine noto.
Una volta effettuato questo è possibile risolvere il sistema lineare equivalente

$Ux = c$ con il metodo delle sostituzioni all'indietro, ovvero

$$x_n = \frac{c_n}{u_{nn}}$$
$$x_i = \frac{1}{u_{ii}}\left(c_i - \sum_{j=i+1}^{n} u_{ij}x_j\right), i = n-1, \dots, 1.$$

Si creino quindi due function, la prima `equi_system` che, prende in input una matrice A e un vettore b e, dopo avere controllare le condizioni per poter ridurre il sistema nella forma triangolare (dando errore altrimenti), effettua tutti i passaggi indicati sopra, fino a creare la matrice U, restituita poi come output, assieme al vettore c.
In seguito si implementi il metodo di sostituzioni all'indietro creando una function `meg_backward` che prende come input una matrice U e un vettore c, controlla che la matrice sia triangolare superiore e gli elementi della diagonale siano diversi da 0 (si possono usare, ad esempio, le funzioni Matlab `triu` e `diag`) e infine genera il vettore x soluzione del sistema triangolare, andando poi a restituirlo come output.
Infine si utilizzino le funzioni per risolvere il seguente sistema lineare

$$\begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = \frac{13}{12} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = \frac{47}{60} \end{cases}$$

generando uno script con la matrice e il termine noto relativo al sistema. Si stampi a schermo l'elemento $U(2,3)$ in format esponenziale con 1 cifra prima della virgola e due dopo.
Infine si trovi la soluzione x e si stampi a schermo la norma 2 del vettore soluzione in formato decimale con 4 cifre prima della virgola e 4 dopo.