Esercizio dalla volta scorsa

Sia T_n la matrice $n \times n$ tale che

$$(T_n)_{ij} = \begin{cases} 1 & i = j, \\ -2 & i = j - 1, \\ 0 & \text{altrimenti}, \end{cases}$$

ad esempio

$$T_4 = \begin{bmatrix} 1 & -2 & & \\ & 1 & -2 & \\ & & 1 & -2 \\ & & & 1 \end{bmatrix}$$

(gli elementi non visualizzati sono 0).

Esercizio

Scrivete una function x = sup_solve_T(b) che risolve il sistema triangolare $T_n x = b$. Riuscite a scrivere questa funzione utilizzando $O(n)$ operazioni aritmetiche, senza memorizzare esplicitamente la matrice T_n ma vedendo come sono fatte le formule per questo caso particolare?

Esercizio dalla volta scorsa

Utilizzando il comando triu (ones(n)), potete generare una matrice triangolare superiore con tutte le entrate nel triangolo uguali a 1:

$$U_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Esercizio

Scrivete una function x = risolvi_sistema (b) che, dato in input un vettore $b \in \mathbb{R}^n$, calcola la soluzione del sistema lineare $U_n x = b$. Sapreste scrivere questa funzione in modo che abbia costo computazionale $\mathcal{O}(n)$?

Algoritmo di Thomas

Esercizio

Trovare la fattorizzazione LU (senza pivoting, tramite matrici elementari di Gauss) della matrice *tridiagonale*

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \alpha_2 & \beta_2 & & \\ & \gamma_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_{n-1} & \alpha_n \end{bmatrix}$$

(al solito, gli elementi non scritti sono 0).

Scrivere una

function [alphahat, beta, gammahat] = lutridiag(alpha, beta, gamma)

che, dati in input i tre vettori α, β, γ, restituisce tre vettori che contengono gli elementi non-nulli della sua fattorizzazione LU.

Algoritmo di Thomas

Esercizio

Scrivere una funzione che, dati in input b e alphahat, beta, gammahat dalla routine precedente, calcola la soluzione del un sistema lineare $Tx = b$ per sostituzione.
Testare la funzione per la soluzione del sistema $Tx = b$ con $n = 50$,
alpha = 1e−8 ∗ ones(n, 1),
beta = ones(n, 1); gamma = beta; b = (1:n)'.
Confrontare l'output della funzione con quello ottenuto usando T \ b in Matlab.

Come sono fatte le matrici L, U restituite da [L, U] = lu(T) di Matlab? Dove hanno elementi non-zero? Controllare con spy(L), spy(U).