Algoritmo di Thomas

Esercizio

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

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \gamma_1 & \alpha_2 & \beta_2 & & \vdots \\ 0 & \gamma_2 & \alpha_3 & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & 0 & \gamma_{n-1} & \alpha_n \end{bmatrix}$$

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

Scrivere una

function [~~alphaL, betaU, gammaU~~] = lutridiag(alpha, beta, gamma)

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

Algoritmo di Thomas

Esercizio

Scrivere una funzione che, dati in input b e alphaL, betaU, gammaU 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).