

Homework Lab 4 - Calcolo Numerico: Esercizi di base per creare delle semplici funzioni

Si propongono alcuni semplici esercizi (alcuni veramente banali) che prevedono l'utilizzo di vettori e matrici.

Si scriva prima l'algoritmo relativo alla funzione richiesta.

Poi, dopo aver provato la sua correttezza semantica, lo si traduca in una function m-file Matlab (completa di intestazione e commenti). Per provare la function Matlab, si scriva anche un semplice script che ne chieda l'utilizzo. A tal fine si propongono alcuni dati per provare il corretto funzionamento.

Se ne svolgano almeno 4 a scelta.

1. `K = krylov(A,v,n)`; assegnata una matrice $A \in \mathbb{R}^{N \times N}$, un vettore colonna $v \in \mathbb{R}^N$ e un intero $n > 0$, produce una matrice K le cui colonne sono i vettori di Krylov $v, Av, A^2v, \dots, A^{n-1}v$. La function deve controllare che la matrice sia quadrata, che il vettore v sia di tipo colonna con dimensione consistente con quella della matrice A , e che n sia intero positivo (in tutti i casi contrari si dia un errore). Si testi con

```
1 A = [1 2 3 4; 5 6 7 8; 9 10 11 12; ...
2     13 14 15 16] % (4 x 4)
3 v = [1;1;1;1]; % (4 x 1)
4 K = krylov(A,v,4)
5 % test errori (lasciarne non commentata una alla volta)
6 %K = krylov(A(1:3,:),v,4)
7 %K = krylov(A,v',4)
8 %K = krylov(A(1:3,1:3),v,4)
9 %K = krylov(A,v,-2)
```

2. `B = ridotta(A,j,k)`; assegnata una matrice A e una coppia di interi positivi j, k , costruisce la matrice ridotta B ottenuta da A eliminando la j -esima riga e la k -esima colonna. La function dovrà prevedere dei controlli sulla consistenza tra gli interi i e j e le dimensioni della matrice A .

Se j è maggiore al numero di righe di A si considerano tutte le righe di A . Stesso comportamento per k e le colonne di A .

Si testi con i seguenti dati:

```
1 A = [1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15; ...
2     16 17 18 19 20]
3 B = ridotta(A,2,2)
4 B = ridotta(A,4,6)
5 B = ridotta(A,6,1)
6 B = ridotta(A,1,7)
7 B = ridotta(A,10,10)
8 B = ridotta(A,5,6)
```

3. `B = newsized(A,m,n)`; assegnata una matrice A e una coppia di interi positivi m, n , estrae da A la sottomatrice formata dalle prime m righe ed n colonne. Se m è maggiore del numero di righe di A , oppure n è maggiore del numero di colonne di A , oppure valgono entrambe queste

condizioni, allora, anzichè estrarre si devono aggiungere degli zeri in corrispondenza delle posizioni (i, j) in cui A non è definita. In tutti i casi, pertanto, in uscita si avrà $B \in \mathbb{R}^{m \times n}$. Si testi la funzione utilizzando gli stessi dati dell'esercizio precedente (cambiando il nome `ridotta` con il nome `newsize`)

4. `B = tridiag(A)`; assegnata una matrice A (quadrata), ne estrae la sua parte tridiagonale (diagonali di indice $-1, 0, 1$) memorizzandola nella matrice B . La function deve testare se la matrice è quadrata (in caso contrario dare un errore). Si testi la function con

```
1 A = [1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15; ...
2     16 17 18 19 20; 21 22 23 24 25] % 5 x 5
3 B = tridiag(A)
4 % test per errore
5 B = tridiag(A(:,1:4))
```

5. `w = disp0(v)`; assegnato un vettore v , produce in output un vettore colonna w tale che $w_j = v_j$ se j è pari, e $w_j = 0$ se j è dispari. La funzione deve conservare il tipo del vettore di ingresso (riga o colonna).

Si testi con

```
1 vr = [1 2 3 4 5 6 7 8 9 10] % vettore riga
2 vc = vr' % vettore colonna
3 wr = disp0(vr)
4 wc = disp0(vc)
```

6. `w = revord(v)`; assegnato un vettore v inverte l'ordine delle sue componenti (l'ultima deve diventare la prima, la penultima deve diventare la seconda, ...). La funzione deve conservare il tipo del vettore di ingresso (riga o colonna).

Si testi con gli stessi vettori dell'esercizio precedente, (cambiando il nome `disp0` con il nome `revord`)

7. `s = sommeparz(a)`; assegnato un vettore a , restituisce il vettore s contenente le somme parziali. Ogni componente s_k di s sarà quindi

$$s_k = \sum_{j=1}^k a_j$$

La funzione deve conservare il tipo del vettore di ingresso (riga o colonna).

Si testi con gli stessi vettori dell'esercizio precedente, (cambiando il nome in `sommeparz`)

8. `[win, wout] = findab(v, a, b)`; assegnato un vettore v (riga o colonna), e due numeri reali a e b tali che $a < b$, restituisce un vettore colonna `win` contenente le sole componenti di v appartenenti all'intervallo $[a, b]$ ed in `wout` (vettore colonna) tutte le altre. La function deve controllare che $a \leq b$ (in caso contrario deve dare un errore). Si testi con

```
1 vr = [1 9 3 10 5 6 2 8 -1 11] % vettore riga
2 [win, wout] = findab(vr, 0, 6)
3 % test per errore
4 [win, wout] = findab(vr, 6, 3)
```