Corso di Abilità Informatiche: MATLAB

Docente S. Finzi Vita

Appello straordinario del 14 novembre 2013

Prima di iniziare create una cartella con il vostro cognome [>> mkdir cognome], *entrate in tale directory* [>> cd cognome] *e lanciate matlab da lì* [>> matlab&].
Alla fine, per consegnare i file con gli esercizi svolti, eseguite sul terminale i comandi :

cd
zip -r cognome.zip cognome

e segnalate al docente di aver terminato.

1. Scrivere una function che presi in input un intero n e una tolleranza ε:
   - costruisce una matrice tridiagonale A di dimensioni $n \times n$ con valore 2 sulla diagonale principale e -1 sulle due diagonali secondarie;
   - calcola tutti i suoi autovalori;
   - risolve il sistema lineare $(n-1)^2 Ax = b$, dove b indica il vettore costante di componenti uguali a 1, mediante il metodo iterativo di Jacobi partendo dal vettore iniziale nullo, arrestandosi appena la norma euclidea del vettore residuo $r_k = Ax_k - b$ risulti minore di ε o si siano superate le 500 iterazioni.

   In output la function deve restituire il vettore V contenente gli autovalori in ordine crescente, la soluzione x_k calcolata dal metodo di Jacobi, e il numero di iterazioni effettuate.

   Creare quindi uno script in grado di chiamare la funzione precedente per $n = 15$ ed $\varepsilon = 1.e-3$, stampando poi tutti i risultati.

2. Scrivere una function che implementa il metodo di Newton per l'approssimazione di una radice di $f(x)$, cioè:

   - a partire da un valore iniziale x_0 calcola la successione $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ per $k = 0, 1, 2, ...$

   - se $k > 100$ oppure $|x_{k+1} - x_k|$ è minore di una tolleranza *toll* allora esce.

   La function deve prendere in input $(x_0, toll, f, f')$ e restituire in output $(x_k, f(x_k), k)$.

   Creare poi uno script che utilizza la function precedente per calcolare le radici di $f(x) = e^x - 2x^2$, con tolleranza 10^{-4}, utilizzando rispettivamente $x_0 = -1, 1, 3$ come dati iniziali, stampando ogni volta i risultati.

3. Creare una function che approssima la soluzione del problema di Cauchy

   $$y'(t) = y(t)(2 - y(t)) \text{ in } (0, T],\ y(0) = c,$$

   in 50 istanti temporali equispaziati, al variare dei valori c e T, restituendo in output i passi e la soluzione approssimata.

   Creare poi uno script che utilizzando la function precedente, tracci un grafico dove mette a confronto, usando colori differenti e inserendo le legende corrispondenti, le soluzioni relative alle condizioni iniziali $c = 0.5, 1, 2, 3$, per $T = 5$.