Corso di Abilità Informatiche: MATLAB

Docente S. Finzi Vita

Appello del 10 luglio 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 assegnati una matrice quadrata A ($n \times n$), un vettore b ($n \times 1$), una tolleranza $toll$, un numero massimo di iterazioni $Nmax$, generi a partire dal vettore nullo X^0 la successione di vettori colonna X^k ($n \times 1$) tali che
$$LX^{k+1} = b + (L - A)X^k, \quad k = 0, 1, 2, ...$$
dove L denota la matrice triangolare inferiore estratta da A, arrestandosi appena la norma infinito del vettore residuo $b - AX^k$ diviene inferiore alla tolleranza o quando si raggiungono $Nmax$ iterazioni. In uscita sarà restituito il numero delle iterazioni effettuate e l'ultimo vettore X^k calcolato.

   Creare poi uno script che utilizzi la function precedente per risolvere il sistema lineare $Ax = b$ con A matrice 7×7 tridiagonale con valori 4 sulla diagonale principale e -1 sulle due diagonali secondarie, b vettore unitario, $Nmax = 50$ e $toll = 0.0001$, stampando il numero di iterazioni effettuate e, solo se tale numero è inferiore a $Nmax$, la soluzione trovata.

2. Creare uno script che assegnato un intero n genera due vettori di n numeri casuali uniformemente distribuiti, X nell'intervallo $(0, 1)$ e Y nell'intervallo $(-1, 1)$, poi il vettore R ottenuto riordinando le componenti di X in senso crescente, e a partire dai dati (R, Y):
   (a) genera il polinomio di grado 3 che meglio li approssima nel senso dei minimi quadrati (data fitting);
   (b) genera l'interpolata cubica a tratti;
   (c) rappresenta nella stessa finestra grafica $(0, 1) \times (-1, 1)$ i dati (come punti) e le due funzioni approssimanti in base ai valori ottenuti su di una griglia uniforme di 50 punti in (0,1), aggiungendo una legenda opportuna.

3. Scrivere una function in grado di risolvere l'equazione differenziale $y'(t) = f(y(t))$, $y(0) = y_0$ nell'intervallo $(0, T)$ con il metodo di Eulero implicito
$$(*) \quad u_{i+1} = u_i + \Delta t f(u_{i+1}), \quad i = 0, 1, ..., n - 1$$
dove $\Delta t = T/n$. Alla function vengono passate la funzione f, il tempo finale T, il numero di passi equidistanti n e la condizione iniziale y_0, e deve restituire il vettore degli istanti temporali t e quello della soluzione approssimata u su di essi. Per risolvere l'equazione implicita (*) si usi la funzione di Matlab *fzero*.

   Creare poi uno script che assegnato n chiama la funzione precedente per $f(y) = \arctan(y), T = 3, y_0 = 1$, poi approssima la soluzione del problema anche con la funzione *ode45* sul medesimo set di punti t e disegna in uno stesso grafico le due soluzioni approssimate con un'opportuna legenda.