Corso di Laurea in Ingegneria Aerospaziale/ Meccanica
Laboratorio di Analisi Numerica
A.A. 2015/2016 – II Ciclo

Esercitazione 3

Creare una cartella <cognome> in C: dove verranno salvati i file creati nella sessione di lavoro.
Appena entrati in MATLAB posizionarsi in <cognome>.
Risolvere in ambiente MATLAB i seguenti esercizi.

RISOLUZIONE EQUAZIONI/SISTEMI NON LINEARI

L'ambiente MATLAB offre una built-in function per il calcolo degli zeri di una funzione *funz* in una singola variabile vicino ad x0:

root=fzero(@funz, x0)

Mentre la funzione built-in

root=roots(c)

calcola le radici del polinomio i cui coefficienti sono le componenti del vettore c.
Se c ha n+1 componenti allora il polinomio è dato da

c(1)*X^n + ... + c(n)*X + c(n+1).

1. Creare due function ***fa.m e fb.m*** per realizzare le seguenti funzioni (a) e (b) e mediante **plot()** disegnare per punti le funzioni.
   a) $f(x)= (x^3-3x+2)\, e^x$; [a,b]=[-3,1/2]
   b) $f(x)= (\ln x)^2/x-1$; [a,b]=[0.2,1]

   Nello script ***ex1.m***, disegnare per punti le due funzioni. Localizzare graficamente gli zeri delle due funzioni per determinare gli iterati iniziali con cui innescare il metodo per determinare le radici. Determinare gli zeri delle funzioni test (a) e (b), utilizzando la funzione **fzero()** di MATLAB.

2. Si consideri il seguente polinomio cubico:
   $$p(x) = 30x^4 - 28x^3 - 110x^2 - 31x + 11$$
   a) Nello script ***ex2.m*** determinare le radici di p(x) mediante la funzione roots() di MATLAB.
   b) Visualizzare *p(x)* nell'intervallo [(xmin-0.1), (xmax+0.1)] dove xmin e xmax sono il minimo e massimo tra le radici trovate, disegnare inoltre un marker nella posizione delle radici.

3. Il programma ***newton.m*** realizza il metodo di Newton per la risoluzione di sistemi non lineari. Si determinino le soluzioni del seguente sistema non lineare:
   $2x_1-\cos(x_2)=0$,
   $2x_2+\sin(x_1)=0$

   Si memorizzi il sistema nella funzione ***funz3.m*** e la matrice Jacobiana, determinata in forma analitica, nella funzione ***jacf3.m*** . (Esistono almeno 4 soluzioni reali)

4. Scrivere una funzione ***bisection.m*** che realizzi il metodo di Bisezione.
   In uno script ***ex4.m*** svolgere i punti seguenti:
   (a) Calcolare e stampare gli iterati x_k nella risoluzione dell'equazione: $f(x) = 2 - e^x$

in [0.5, 1] ottenuti con i metodi Newton con $x_0=0$ (***newton.m***), e con quello di bisezione.

(b) Confrontare i due metodi in termini di numero di iterazioni per raggiungere una certa accuratezza: 1e-2 e 1e-14 (radice esatta x*=ln 2=0.6931471806).

5. E' stato suggerito di mettere le scorie nucleari in speciali cilindri, che saranno buttati nell'oceano. Fintanto che i cilindri restano intatti, la contaminazione del mare circostante è minima. Risolvendo le equazioni del moto dei cilindri dal momento in cui cadono in acqua, si ottiene la seguente relazione tra la velocità di impatto v e la profondità dell'acqua D:

$$D = \frac{1}{k^2 g}[W(W-B)\ln(1+\frac{kv}{W-B}) - Wkv] \qquad (*)$$

dove W è il peso dei cilindri, B è la spinta idrostatica dei cilindri, g è la costante gravitazionale e k è il coefficiente di resistenza dei cilindri. Esperimenti mostrano che i cilindri si rompono quando avviene un impatto con il fondo marino a una velocità (verso il basso) di 40 piedi al secondo. La spinta idrostatica dei cilindri è di 570 libbre, k=0.08 libbre per piede al secondo, g=32 piedi al secondo quadro. Se i contenitori pesano 527 libbre, risolvere il problema nel file ***ex5.m*** usando il metodo di regula falsi (***regulafalsi.m***) per determinare se possiamo lanciare con sicurezza i cilindri in 300 piedi di acqua. (Scrivere la funzione (*) nella forma f(v)=0, nella function ***fun_cil.m***) Fare il grafico della funzione (*) per v in [10,80] e mostrare con un marker la radice.