Tutoraggio #4

Federico Pichi

15 maggio 2019

Esercizio 1

Per interpolare la funzione $f(x) = 0.0039 + \frac{0.0058}{1+e^x}$ sull'intervallo $[-5, 5]$ si usi l'interpolazione polinomiale su $(n + 1)$ nodi equispaziati e di Chebyshev-Gauss-Lobatto. Confrontare i risultati e spiegarne l'andamento.

Esercizio 2

Si vuole determinare il volume V occupato da un gas ad una temperatura T e soggetto ad una pressione p. L'equazione di stato è data da

$$[p + a(N/V)^2](V - Nb) = nKT \ .$$

Per l'anidride carbonica (CO2) i coefficienti a e b valgono rispettivamente $a = 0.401$ Pa m^6 e $b = 42.7 \cdot 10^{-6}$m^3 (Pa sta per Pascal). Si trovi il volume occupato da $N = 1000$ molecole di anidride carbonica poste ad una temperatura $T = 300K$ e ad una pressione $p = 3.5 \cdot 10^7 Pa$ utilizzando il metodo di bisezione con una accuratezza pari a $tol = 10^{-12}$ (la costante di Boltzmann è pari a $k = 1.3806503 \cdot 10^{-23}$ Joule K^{-1}).

Esercizio 3

Implementare in Matlab il metodo di Newton come una function che prenda in input la funzione, la sua derivata, un punto iniziale, la tolleranza ed il numero massimo di iterazioni e che restituisca lo zero della funzione, il numero di iterate impiegate, la storia di convergenza ed il vettore dei residui.

Esercizio 4

Si consideri la funzione $f(x) = \cos^2(2x) - x^2$ nell'intervallo $0 \leq x \leq 1.5$. Fissata la tolleranza ϵ sull'errore assoluto pari a 10^{-10}, determinare, per via sperimentale, gli intervalli in cui scegliere il dato iniziale x_0 in modo che il metodo di Newton risulti convergente alla radice $\alpha \approx 0.5149$.

Soluzione 1

Osserviamo che l'errore dell'interpolazione polinomiale su nodi di Chebyshev decresce fino al raggiungimento dell'epsilon di macchina e poi oscilla intorno a tale valore per effetto della propagazione degli errori di arrotondamento ed al malcondizionamento della matrice di Vandermonde. L'errore dell'interpolazione polinomiale su nodi di Chebyshev decresce più velocemente di una qualsiasi potenza di n: in tal caso si dice che la convergenza è di tipo esponenziale.

Codice 1: Soluzione 2

```
f = @(x)0.0039+0.0058./(1+exp(x));
a = -5; b = 5;
x_1 = linspace(xa,xb,1000);
y_1 = f(x_1);
nn = 4:4:70;
m = length(nn);
err_c = [];
err_p = [];
for n=nn
    for i=0:n
    x_cgl = cos(pi*i/n); % Chebyshev-Gauss-Lobatto
    x_cg = cos(pi*(2*i+1)/(2*(n+1))); % Chebyshev-Gauss
    x_c(i+1) = ((a+b)/2)-((b-a)/2)*x_cgl;
    end

    y_c = f(x_c);
  p_che = polyfit(x_c,y_c,n);
  y_che = polyval(p_che,x_1);
  e_che = norm(y_1-y_che,inf);
  err_c = [err_c; e_che];

    x_p = linspace(a,b,n);

    y_p = f(x_p);
    p_p = polyfit(x_p,y_p,n);
  y_p = polyval(p_p,x_1);
  e_p = norm(y_1-y_p,inf);
  err_p = [err_p; e_p];
end
semilogy(nn, err_c, 'r-')
hold on
semilogy(nn, err_p, 'b-')
```

Soluzione 2

Si tratta di calcolare gli zeri della funzione $f(V) = pV + aN^2/V - abN^3/V^2 - pNb - kNT$. Uno studio grafico rivela che questa funzione presenta un solo zero semplice fra 0.01 e 0.06 con $f(0.01) < 0$ e $f(0.06) > 0$. Calcoliamo tale zero come richiesto con il metodo di bisezione tramite le seguenti istruzioni:

Codice 2: Soluzione 1

```
>> f=@(x) 35000000*x+401000./x-17122.7./x.^2-1494500;
>> [zero,res,niter]=bisection(f,0.01,0.06,1.e-12,100)
```

Soluzione 3

Codice 3: Soluzione 3

```
function [zero,iter,xvect,xdif,fx]=newton(fun,dfun,x0,tol,nmax)
% NEWTON metodo di Newton.
% [ZERO,ITER,XVECT,XDIF,FX]=NEWTON(FUN,DFUN,X0,TOL,NMAX) cerca lo
% zero di una funzione continua FUN nell'intervallo [A,B] usando il metodo di
% Newton e partendo da X0.  TOL e NMAX specificano la tolleranza ed il massimo
% numero di iterazioni.  FUN e DFUN ricevono in ingresso lo scalare x e restituiscono
% un valore scalare reale.  DFUN è la funzione derivata di FUN.
% ZERO è l'approssimazione della radice, ITER è il numero di iterate svolte.
% XVECT è il vettore delle iterate, XDIF il vettore delle differenze tra iterate
% successive, FX il vettore dei residui.
err=tol+1; iter=0; xdif=[];
fx0=fun(x0); xvect=x0; fx=fx0;
while iter < nmax && err> tol
    dfx0=dfun(x0);
    if dfx0==0
        fprintf('Arresto causa annullamento derivata\n'); zero=[];return
    end
x=x0-fx0/dfx0;
err=abs(x-x0); x0=x; fx0=fun(x0); iter=iter+1;
xvect=[xvect;x0]; fx=[fx;fx0]; xdif=[xdif; err];
end
zero=xvect(end);
end
```

Soluzione 4

Osserviamo che per $0 < x_0 \leq 0.02$, $0.94 \leq x_0 \leq 1.13$ e $1.476 \leq x_0 \leq 1.5$, il metodo converge alla soluzione $-\alpha$. Per ogni altro valore di x_0 in $[0, 1.5]$ il metodo converge ad α.