Tutoraggio #5

Federico Pichi

23 maggio 2019

Esercizio 1

Data la funzione $f(x) = cosh(x) + cos(x) - \gamma$, per $\gamma = 1, 2, 3$ si individui un intervallo contenente uno zero di f e lo si calcoli con il metodo di bisezione con una accuratezza pari a 10^{-10}. Si calcoli a priori il numero di iterate minime impiegate tramite la stima

$$k_{min} > \log_2\left(\frac{b-a}{\epsilon}\right) - 1 \ .$$

Esercizio 2

Si applichi il metodo di Newton per il calcolo dello zero di $f(x) = x^3 - 3x^2 2^{-x} + 3x4^{-x} - 8^{-x}$ in $[0, 1]$ ponendo $x_0 = 0$ e $tol = \epsilon_M$, si analizzi sperimentalmente l'ordine di convergenza e si commenti il risultato.

Esercizio 3

Implementare in Matlab il metodo di punto fisso come una function che prenda in input la funzione, la funzione di iterazione, 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

Nello studio della dinamica delle popolazioni (di batteri, ad esempio) l'equazione $y = \phi(x) = xR(x)$ stabilisce un legame fra il numero x di individui di una generazione ed il numero y di individui della generazione successiva. La funzione $R(x)$ modella il tasso di variazione della popolazione in esame e può essere scelta in vari modi:

- il modello di crescita in presenza di risorse limitate, o modello discreto di Verhulst

$$R(x) = R_V(x) = \frac{r}{1 + xK}, \quad r > 0,\ K > 0$$

- il modello predatore/preda con saturazione

$$R(x) = R_P(x) = \frac{rx}{1 + (x/K)^2}, \quad K > 0 \ .$$

Studiare graficamente la presenza di punti fissi per le due funzioni di iterazioni ϕ_V e ϕ_P scegliendo $r = 3$ e $K = 1$ ed analizzare le proprietà di convergenza ad essi.

Soluzione 1

Utilizzando il comando fplot studiamo l'andamento della funzione data al variare di γ. Per $\gamma = 1$ essa non ammette zeri reali. Per $\gamma = 2$, si ha il solo zero $\alpha = 0$ con molteplicità 4 (cioè tale che $f(\alpha) = f'(\alpha) = f''(\alpha) = f'''(\alpha) = 0$ ma $f^{(4)}(\alpha) \neq 0$). Per $\gamma = 3$, f presenta due zeri semplici, uno nell'intervallo $[-3, -1]$, uno in $[1, 3]$. Nel caso $\gamma = 2$ il metodo di bisezione non può essere impiegato non essendo possibile trovare un intervallo per il quale $f(a)f(b) < 0$. Per $\gamma = 3$ a partire da $[a, b] = [-3, -1]$, il metodo di bisezione converge in 34 iterazioni allo zero $\alpha = -1.8579$ (con $f(\alpha) \approx -3.6 \cdot 10 - 12$). Analogamente, scegliendo $[a, b] = [1, 3]$, per $\gamma = 3$ si ha convergenza in 34 iterazioni allo zero $\alpha = 1.8579$ (con $f(\alpha) \approx -3.68 \cdot 10 - 12$).

Codice 1: Soluzione 1

```
gamma = [1,2,3];
x = linspace(-3,3);

for g = gamma
    f = @(x) cosh(x)+cos(x)-g;
    plot(x, f(x), 'color', rand(1,3))
    hold on
    grid
end

a = -3;
b = -1;
tol = 1.e-10;
nmax = 200;
[zero_1,res_1,niter_1] = bisezione(f,a,b,tol,nmax);
k_min_1 = log2((b-a)/tol)-1;
a = 1;
b = 3;
[zero_2,res_2,niter_2] = bisezione(f,a,b,tol,nmax);
k_min_2 = log2((b-a)/tol)-1;
```

Soluzione 2

Si osserva che il metodo di Newton converge in 43 iterazioni al valore $\alpha = 0.641182985886554$. Sappiamo che la differenza tra due iterate successive $|x_{k+1} - x_k|$ è una buona stima dell'errore $|\alpha - x_k|$ per k grande. Rappresentiamo pertanto in scala semilogaritmica il vettore xdif fornito in output dal metodo di newton al variare dell'iterazione k. Si nota che l'errore decresce linearmente per $k \leq 30$, poi comincia ad oscillare attorno a valori dell'ordine di 10^{-6}. Il comportamento dell'errore per $k \leq 30$ è chiaramente lineare, sintomo che la radice non è semplice, ma multipla. Analizzando meglio l'espressione di f osserviamo infatti che $f(x) = (x - 2^{-x})^3$, ovvero la radice ha molteplicità 3. Per recuperare il secondo ordine di convergenza potremmo ricorrere al metodo di Newton modificato. La ragione del comportamento oscillante dell'errore per $30 \leq k \leq 43$ è da imputare agli errori di cancellazione che si generano dalla somma degli addendi di f e della sua derivata in prossimità della radice. Infatti, risolvendo nuovamente l'equazione non lineare con il metodo di Newton, ora con $f(x) = (x - 2^{-x})^3$ anzich è nella forma originaria: otteniamo convergenza in 85 iterazioni al valore $\alpha = 0.6411857445049863$, vediamo che le oscillazioni nel vettore xdif spariscono e l'errore decade (sempre linearmente perch è la radice è multipla) fino alla tolleranza imposta per il test d'arresto, ovvero l'epsilon di macchina.

Codice 2: Soluzione 2

```
f=@(x) x.^3-3.*x.^2.*2.^(-x)+3.*x.*4.^(-x)-8.^(-x);
df=@(x) 3.*x.^2-6.*x.*2.^(-x)+3.*x.^2.*2.^(-x).*log(2)...
    +3.*4.^(-x)-3.*x.*4.^(-x).*log(4)+8.^(-x).*log(8);
[zero,iter,xvect,xdif,fx]=newton(f,df,0,eps,100);
semilogy(xdif,'r','Linewidth',2);
grid on, hold on
f=@(x)(x-2^(-x)).^3;
df=@(x)(1+2^(-x)*log(2))*3.*(x-2^(-x)).^2;
[zero1,iter1,xvect1,xdif1,fx1]=newton(f,df,1,eps,100);
semilogy(xdif1,'b','Linewidth',2);
```

Soluzione 3

Codice 3: Soluzione 3

```
function [zero,iter,xvect,xdif,fx]=puntofisso(fun,phi,x0,tol,nmax)
% FIXPOINT iterazione di punto fisso.
% [ZERO,ITER,XVECT,XDIF,FX]=FIXPOINT(FUN,PHI,X0,TOL,NMAX) cerca lo
% zero di una funzione continua FUN usando il metodo di punto fisso X=PHI(X),
% partendo dal dato iniziale X0.  FUN riceve in ingresso lo scalare x e restituisce un
% valore scalare reale.  TOL specifica la tolleranza del metodo.  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;
fx0=fun(x0); xdif=[]; xvect=x0; fx=fx0;
while iter < nmax && err > tol
    x=phi(x0);
    err=abs(x-x0); xdif=[xdif; err]; xvect=[xvect;x];
    x0=x; fx0=fun(x0); fx=[fx;fx0];
    iter=iter+1;
end
zero=xvect(end);
end
```

Soluzione 4

Applichiamo le iterazioni di punto fisso alla funzione $\phi_V(x) = rx/(1 + xK)$ del modello discreto di Verhulst ed alla funzione $\phi_P(x) = rx^2/(1+(x/K)^2)$ del modello predatore/preda. Se partiamo dal dato iniziale $x_0 = 1$ troviamo il punto fisso $\alpha = 2$ nel primo caso e $\alpha = 2.6180$ nel secondo. Il punto fisso $\alpha = 0$ comune a ϕ_V e ϕ_P può essere calcolato solo come punto fisso di ϕ_P, ma non di ϕ_V. Infatti $\phi_P'(\alpha) = 0$, mentre $|\phi_V'(\alpha)| = r > 1$. Analogamente il punto fisso $\alpha = 0.3820$ di ϕ_P non può essere calcolato in quanto $|\phi_P'(\alpha)| > 1$.

Codice 4: Soluzione 4

```
r = 3;
K = 1;
tol = 1.e-10;
nmax = 100;
id = @(x) x;
phi_v = @(x) r*x./(1 + K*x);
f_v = @(x) r./(1 + K*x);
phi_p = @(x) r*x.^2./(1 + (x/K).^2);
f_p = @(x) r*x./(1 + (x/K).^2);
fplot(id, [-10,10], 'LineWidth', 2)
hold on
fplot(phi_p, [-10,10])
fplot(phi_v, [-10,10])

x_0 = 1;
a = - 0.5;
b = 4;
[zero_v,iter_v,xvect_v,xdif_v,fx_v]=puntofisso(f_v,phi_v,x_0,tol,nmax);
[zero_p,iter_p,xvect_p,xdif_p,fx_p]=puntofisso(f_p,phi_p,x_0,tol,nmax);
```