Metodo di bisezione e metodo di Newton

Laboratorio di Calcolo Numerico

Federico Piazzon

3 Maggio 2022

Outline

Attenzione!!

Lezione altamente interattiva!

Per seguire questa lezione è necessario fare i propri programmi.

- Preparare una cartella di lavoro odierna
- Avviare matlab e settate tale cartella come current folder
- Scaricare i programmi forniti dal docente (ultime versioni) nella cartella di lavoro.

Metodo di bisezione

Richiamo teorico

- **Ipotesi:** $[a, b] \subset \mathbb{R}$, $f \in \mathcal{C}^0([a, b])$, $f(a) \cdot f(b) < 0$
- **Algoritmo** (con test dello scarto):

```
INPUT f a,b, toll
n=0
WHILE (b − a)>toll DO
        x=(a+b)/2
        IF f(a)f(x)<0 DO
                b=x
        ELSE DO
                a=x
        END IF
        n=n+1
END WHILE
```

- **Convergenza:** garantita sempre dalla *stima a priori* $|\xi - x_n| \leq (b - a)2^{-(n+1)}$
- **Test alternativo:** residuo pesato approssimato

Implementazione

Esercizio 4.1

Creare una **function** `mybisezione.m` che implementi l'algoritmo sopra descritto.

Creare poi uno **script** `testbisezione.m` che testi la function con i seguenti dati $a = -\pi/6$ $b = \pi/4$ $f(x) = \sin(x)$, `toll=1e-9` [illegible]

Obiettivo

Es 4.2 (svolto)

Scrivere una function `Bisezione.m` che implementi il metodo di bisezione con **scelta del test di arresto e controllo sulla** possibile **convergenza in numero finito di passi**.

NB:

- nel test dello scarto possiamo stimare a priori le iterazioni necessarie: $n^*(toll) := \lceil \log_2(b-a) - \log_2(toll) \rceil - 1$ (in matlab si usa `ceil`).
- vogliamo poter scegliere se usare scarto, residuo pesato approssimato o il minimo dei 2
- vogliamo effettuare un controllo sui dati in input: $f(a)f(b) < 0$?
- vogliamo controllare ad ogni passo se $f(x_n) = 0$

help Bisezione

Editor - /home/federico/Bisezione.m

radicesecgrado.m | pigreco.m | untitled2* | succricorrente.m | Bisezione.m

```
function[zero,res,wres,iterates,flag]=Bisezione(f,a,b,toll,method)
%% METODO DI BISEZIONE
%
% ----------INPUT-----------------------------------------------------------
% f             function handle di una funzione continua da [a,b] in R
% a             double [1 x 1] estr inf intervallo
% b             double [1 x 1] sup intervallo
% toll          double [1 x 1] tolleranza per criterio di arresto
% method        char [1 x 1] test di arresto:
%                       method = 's' test dello scarto
%                       method = 'r' test del residuo pesato approssimato
%                       method = 'm' minimo delle due stime < toll
%
% ----------OUTPUT----------------------------------------------------------
% zero          double [1 x 1] ultima approssimazione della radice
% res           double [1 x 1] modulo del residuo
% wres          double [1 x 1] modulo del residuo pesato approssimato
% iterates      double [3 x N] iterate del metodo di bisezione:
%                       iterates(1,:)= x_0,x_1,...
%                       iterates(2,:)= a_0,a_1,...
%                       iterates(3,:)= b_0,b_1,...
% flag          char [1 x 1] stato:
%                       flag = 's' uscita per test dello scarto
%                       flag = 'r' uscita per test dell residuo pesato approssimato
%                       flag = 'b' uscita causata da entrambi i test
%                       flag = 'f' residuo 0 in numero finito di iterazioni
%--------------------------------------------------------------------------
```

Scarichiamo il file da moodle e vediamo lo su matlab

Alcuni commenti ed osservazioni

- `iterates` viene inizializzato (per efficienza) calcolando i massimo numero di componenti che potrà avere grazie alla stima a priori
- Per calcolare il residuo pesato approssimato serve avere già iterato il metodo almeno una volta: in caso contrario `wres` viene inizializzato a `NaN`. Si noti l'iterazione prima del ciclo `while` senza controllo su `wres` per non generare errori.

Esercizio 4.3 (calcolo di $\sqrt{2}$)

Tempo: 10-15 min.

Si scriva uno script che

- definisca l'anonymous function $f(x) := x^2 - 2$ e la plotti (`figure(1)`) in $[1, 2]$ assieme alla retta $y = 0$
- Calcoli lo zero di f chiamando `Bisezione.m` con il metodo `'m'` e `toll`$= 10^{-12}$
- Stampi a video i risultati salienti tra cui il criterio per il quale si è arrestato l'algoritmo.
- Faccia un grafico semilogaritmico (`figure(2)`) dell'errore delle iterate e del modulo dei residui.
- Stampi il rapporto errori vs moduli dei residui e la retta che vale costantemente $1/f'(\sqrt{2})$.

Si ripeta l'esperimento con $f = (x^2 - 2)^3$ e `toll`$= 10^{-4}$. Si motivino i risultati.

Pro e contro di Bisezione

Pro

- Convergenza globale garantita (sotto semplici ipotesi)
- Non necessita di conoscenza della derivata

Contro

- Radici doppie di funz che non cambiano segno: inapplicabilità.
- Relativa lentezza della successione di iterate (in che senso?)

Ordine di convergenza

Si ricorda che una successione $\{x_k\}_{k\in\mathbb{N}}$ convergente a $x^*\in\mathbb{R}$ ha ordine di convergenza:

- $p=1$ se $\exists L\in(0,1)$: $\lim_k \frac{|x_{k+1}-x^*|}{|x_k-x^*|}=L$
- $p>1$ se $\exists L\in(0,+\infty)$: $\lim_k \frac{|x_{k+1}-x^*|}{|x_k-x^*|^p}=L$

Una tale L è detta costante asintotica del metodo.
Si noti che per $k>>1$ e per successioni di ordine $p\geq 1$ abbiamo

$$e_k := |x_k-x^*| \preceq L|x_{k-1}-x^*|^p =: e_{k-1} \preceq \cdots \preceq L^{2k-1}e_0^{p^k}.$$

Osservazione

Il metodo di bisezione non ha ordine di convergenza, ma la sua stima a priori ha ordine 1

Stima della costante asintotica

Il rapporto tra gli errori non è calcolabile senza conoscere la soluzione. Vale però il seguente fatto:

Stima con lo scarto

Se $x_k \to x^*$ con ordine $p \geq 1$ e costante L allora

$$\lim \frac{|s_{k+1}|}{|s_k|^p} := \lim \frac{|x_{k+1} - x_k|}{|x_k - x_{k-1}|^p} = L.$$

- Se $L_{k,p} := \frac{|s_{k+1}|}{|s_k|^p}$ si mantiene limitato per $k >> 1$ possiamo presupporre che l'ordine sia p e
- in tal caso consideriamo l'approssimazione $L \sim L_{k,p}$ per $k >> 1$.

Metodo di Newton

Richiamo teorico

Convergenza locale di Newton

Supponiamo $f \in \mathcal{C}^2([a,b])$, $\exists x^* \in [a,b] : f(x^*) = 0$ con $f'(x) \neq 0$ in un intorno forato di x^*, allora esiste $\epsilon > 0$ tale che, per ogni scelta di $x_0 \in (x_0 - \epsilon, x_0 + \epsilon)$, la successione

$$x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)} =: x_k - s_k$$

converge a x^* con ordine almeno 1.
Se inoltre $f'(x^*) \neq 0$ allora (al più diminuendo ϵ) l'ordine di convergenza è $p = 2$

- Se $f'(x^*) \neq 0$ (**radice semplice**) allora $p = 2$ e $L = \frac{f''(x^*)}{2f'(x^*)}$
- Se $f(x^*) = f'(x^*) = \cdots = f^{(m-1)}(x^*) = 0$ e $f^{(m)}(x^*) \neq 0$ (**radice di ordine m**) allora $p = 1$ e $L = \frac{m-1}{m}$

Algoritmo

Un implementazione elementare può essere:

```
INPUT f,df,x0, toll,itmax
x=x0; n=0; s=toll+1
WHILE abs(s)>toll AND n<itmax DO
        IF df(x)=0 DO
                ERROR
        ELSE DO
                s=f(x)/df(x)
                x=x-s
        END IF
        n=n+1
END WHILE
```

Arrestare Newton?

Test di arresto

Il test dello scarto è lo standard per il metodo di Newton perchè:

- **Radici Semplici:** in caso di convergenza almeno quadratica $|e_k| \leq |s_k|$ per $k >> 1$
- **Radici di ordine m:** in caso di convergenza lineare $|e_k| \sim m|s_k|$ per $k \to +\infty$

Implementazione di base Newton

Esercizio 4.4

Creare una **function** `mynewton.m` con l'implementazione dell'algoritmo sopra descritto in pseudocodice.
Modificare lo script `testbisezione.m` per ottenere uno script `testnewton.` che risolva la stessa equazione ma con il metodo di Newton appena implementato.

help Newton.m

Editor - /home/federico/Newton.m

Bisezione.m | provabisez.m | Newton.m | provanewton.m

```matlab
function[zero,res,iterates,flag]=Newton(f,df,x0,toll,itmax,method)
%% METODO DI NEWTON CON SCELTA DEL CRITERIO DI ARRESTO
%
%                                               Versione 04-19-2021
%                                               Federico Piazzon
%
% ----------INPUT---------------------------------------------------------
% f             function handle di una funzione continua da [a,b] in R
% df
% x0            double [1 x 1] punto di partenza
% toll          double [1 x 1] tolleranza per criterio di arresto
% itmax         double [1 X 1] massimo numero di iterazioni
% method        char [1 x 1] test di arresto:
%                       method = 's' test dello scarto
%                       method = 'r' test del residuo pesato approssimato
%                       method = 'm' minimo delle due stime < toll
%
% ----------OUTPUT--------------------------------------------------------
% zero          double [1 x 1] ultima approssimazione della radice
% res           double [1 x 1] modulo del residuo
% iterates      double [1 x N] iterate del metodo di bisezione:
% flag          char [1 x 1] stato:
%                       flag = 's' uscita per test dello scarto
%                       flag = 'r' uscita per test dell residuo
%                       flag = 'a' uscita per entrambi i test
%                       flag = 'e' raggiunto il massimo numero di
%                                  iterazioni
%                       flag = 'f' residuo 0 in numero finito di iterazioni
%----------FUNCTION BODY--------------------------------------------------
```

Scarichiamo la function e vediamola su nell'editor

Esercizio 4.5 (importantissimo)

Tempo stimato 30 min.

Si crei uno script che:

1. definisca le funzioni $f_1(x) := x^2 - 2$, $f_3(x) = (x^2 - 2)^3$ ed $f_5(x) := (x^2 - 2)^5$ e le loro derivate prime tramite anonymous functions. Plotti funzioni e derivate in $[1, 2]$ in tre grafici con legenda e titolo. Stampi a video iterazioni errore finale motivo dello stop.
2. Approssimi la soluzione $x^* = \sqrt{2}$ di $f_m(x) = 0$ (nei tre casi $m = 1, 3, 5$) chiamando il metodo di Newton con il criterio dello scarto con $x_0 = 2$, tolleranza 10^{-8} e al più 100 iterazioni.
3. Crei un grafico semilogaritmico (per i 3 valori di m) con modulo del residuo, modulo dello scarto ed errore ad ogni passo.
4. Crei un grafico semilogaritmico (per i 3 valori di m) con $|s_{k+1}|/|s_k|^{p_m}$ dove p_m va opportunamente scelto al variare di m.
5. Crei un grafico semilogaritmico (per i 3 valori di m) con rapporto $e_k/|s_k|^{p_m}$.

Quali sono i limite teorici delle successioni plottate ai punti 3,4,5?

Esercizio 4.6

Tempo stimato 15 min

Si ripeta sostanzialmente l'esercizio precedente (saltando il punto 4), ma con due metodi: Newton e Bisezione, il primo con il criterio dello scarto e il secondo con il residuo pesato approssimato.
Si consideri a tal fine la funzione $f(x) := \exp(1 - 1/x) - e + 0.01$ definita su $x > 0$, si richieda una tolleranza di 10^{-12} per entrambi i metodi.