UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

DIPARTIMENTO MATEMATICA

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

Laboratorio di Calcolo Numerico LAB 4

Metodo di Bisezione

Docenti: E. Bachini, L. Bruni

Email: elena.bachini@unipd.it Email: bruni@math.unipd.it

27 marzo 2024

Prima di iniziare

Per realizzare quest'esercizio si raccomanda di studiare le lezioni di calcolo numerico sugli *zeri di funzioni non lineari* e il contenuto dei precedenti laboratori di CN di introduzione all'uso di MATLAB.

Outline

Esercizio 1

- Si scriva uno script che rappresenti graficamente una data funzione $f(x)$ in un dato intervallo $[a, b]$. In ogni grafico si rappresentino anche l'asse delle ascisse e delle ordinate come una linea tratteggiata grigia.
- Si considerino le seguenti funzioni e si usi il precedente script per disegnarle modificando opportunamente l'intervallo $[a, b]$ in modo da visualizzare almeno uno zero della funzione.

$$
\begin{aligned}
f(x) &= \sqrt{x+1} - e^{-x} && a = -0.5, \quad b = 1 \\
f(x) &= e^{x} - 5 + x^{2} && a = -2, \quad b = 2 \\
f(x) &= 3x - \cos x && a = -1, \quad b = 1 \\
f(x) &= x + \log x && a = 0.5, \quad b = 1
\end{aligned}
$$

Esercizio 1

- Si scrivano le precedenti funzioni nella forma $x = g(x)$ (ad esempio la prima funzione diventerebbe $x = e^{-2x} - 1$)
- Si modifichi lo script precedente in modo tale che, oltre a disegnare la $y = f(x)$ in $[a, b]$ (figura 1), disegni anche $y = x$ e $y = g(x)$ in $[a, b]$ (figura 2)
- I grafici dovranno avere un titolo, un label sugli assi, la legenda e colori e spessori di linea adeguati (non quelli di default). Nonchè gli estremi degli assi definiti per aiutare la visualizzazione degli zeri

Nota.

Che relazione esiste tra gli zeri di $f(x)$ (x per i quali $f(x) = 0$) e i punti di intersezione di x con $g(x)$?

Esercizio 2

Gli obiettivi di questo esercizio sono:

- la creazione di una funzione `bisezione.m` che implementi l'algoritmo di bisezione per trovare lo zero di una funzione continua
- la creazione di uno script `CN_lab4_es2.m` in cui
  - si definiscono i dati di input della funzione `bisezione.m`
  - si esegue la funzione `bisezione.m`
  - si visualizzano i risultati della stessa

Esercizio 2. Metodo di bisezione

Si supponga sia $f : [a, b] \to \mathbb{R}$ una funzione continua tale che $f(a) \cdot f(b) < 0$.

E' noto che per il teorema degli zeri di funzioni continue, esiste almeno un punto x^* tale che $f(x^*) = 0$.

Per approssimare x^* utilizziamo l'algoritmo di bisezione che genera una successione di intervalli (a_k, b_k) con

- $f(a_k) \cdot f(b_k) < 0$
- $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$
- $|b_k - a_k| = \frac{1}{2}|b_{k-1} - a_{k-1}|$

Fissate la tolleranza ϵ, si arresta l'algoritmo quando

$$|b_k - a_k| \leq \epsilon.$$

Esercizio 2. Intestazione function `bisezione.m`

Si implementi in Matlab, modificando opportunamente il file `bisezione.m` fornito in moodle, il metodo di bisezione

```
function [xv, fxv, n] = bisezione (f, a, b, toll, nmax)
% BISEZIONE Metodo di Bisezione
%
% Uso:
% [xv, fxv, n] = bisezione(f, a, b, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione
%   a:      estremo sinistro
%   b:      estremo destro
%   toll:   tolleranza richiesta sul valore della funzione
%   nmax:   massimo indice dell'iterazione permesso
%
% Dati di uscita:
%   xv:     vettore contenente le iterazioni
%   fxv:    vettore contenente i corrispondenti residui
%   n:      indice dell'iterazione finale calcolata
```

Esercizio 2. `bisezione.m`

Operativamente, dati in input:

- f: la funzione di cui si vogliono calcolare gli zeri
- a e b: gli estremi dell'intervallo con $a \leq b$ e $f(a) \cdot f(b) < 0$
- toll: la tolleranza
- nmax: il numero massimo di iterazioni da compiere

posto $a_k = a$, $b_k = b$, alla k-sima iterazione

- calcola $x_k = (a_k + b_k)/2$
- 1. se $f(a_k) \cdot f(x_k) > 0$ pone "$a_{k+1} = x_k$", "$b_{k+1} = b_k$"
  2. se $f(a_k) \cdot f(x_k) < 0$ pone "$a_{k+1} = a_k$", "$b_{k+1} = x_k$"
  3. se $f(a_k) \cdot f(x_k) = 0$ pone "$a_{k+1} = x_k$", "$b_{k+1} = x_k$"
- termina il processo se le condizioni d'arresto sono verificate, cioè sono state svolte almeno kmax iterazioni o $|f(x_k)| \leq$ toll

Esercizio 2. Pseudo codice `bisezione.m`

```
[xv, fxv,  k] = bisezione (f, a, b, toll, kmax)

   k = 0
   ampiezza = 2*toll
   ak = a;
   bk = b
   fak = f(ak)
   xv = []
   fxv = []

   while (ampiezza > toll) and (k < kmax) do
      k = k + 1

      xk = (ak + bk) * 0.5
      fxk = f(xk)

      xv = [xv xk]
      fxv = [fxv fxk]

      if fak * fxk < 0 then
            bk = xk
      else if fak * fxk > 0 then
            ak = xk
            fak = fxk
      else
            break %(infatti fxk = 0)
      end if

      ampiezza = abs(bk-ak)

   end while
```

Esercizio 2. Script CN_lab4_es2.m

Si scriva lo script CN_lab4_es2.m modificando il template fornito

```
%-------------------------------------------------------------------------
% Script per il Metodo di Bisezione
% Necessita delle Function bisezione
%-------------------------------------------------------------------------
clear all
clc
disp('METODO DI BISEZIONE');

% Ingresso dati
% funzione
f =
%estremi a e b
a =
b =
%-----------------------
%tolleranza
toll =
%numero massimo iterazioni
kmax =
%-----------------------
```

Esercizio 2. Script CN_lab4_es2.m

```
% Controllo sui valori discordi
% se NON discordi,
%     lancia un msg d''errore
% altrimenti

    % Visualizza di controllo dati inseriti
    fprintf('--------DATI--------\n');

    disp('f =  ');
    disp(f);
    fprintf('a = %f \n', a);
    fprintf('b = %f \n', b);
    fprintf('toll = %f \n', toll);
    fprintf('n. max iterazioni = %f \n', kmax);


    % funzione bisezione il metodo di bisezione
    [xv, fxv, k] = bisezione (f, a, b, toll, kmax);

    %se si é raggiunto il numero max di iterazioni, lancia un warning
    %altrimenti visualizza il valore finale dell'ascissa, della funzione
          e
    %il num di iterazioni
```

Esercizio 2. Script CN_lab4_es2.m

```
% Plotta un grafico di convergenza
% ascisse iterazioni, ordinate valore assoluto delle funzioni f(
    xk)
% calcolate

% ATTENZIONE: NON USARE PLOT MA SEMILOGY(X,Y) ---> GUARDA
% L'HELP!!!!
```

Nota. (semilogy)

Il grafico richiesto dovrà essere disegnato con l'ordinata in scala logaritmica (guardate di quanti ordini cambia l'errore!!).Non si userà il comando `plot` *ma* `semilogy`*. Provare a consultare l'help di Matlab.*

Consegna Lab4

- Usate le funzioni proposte nell'esercizio 1 come test (impostando `toll` = $10^{-}8$ e `nmax = 100`) e salvate le immagini dei grafici di convergenza in un formato diverso dal formato di default (.fig) e con nomi significativi (che permettano di capire che grafico si sta aprendo)
- Create un file .zip con i 4 grafici di convergenza ottenuti e opportunamente modificati (cognome_nome_lab4.zip) e caricatelo su moodle nella sezione dedicata

Nota.

Potete usare anche il comando `gtext` *per inserire il vostro nome o altro testo sulle figure. (si veda* `help gtext`*).*