

Laboratorio Calcolo Numerico

Esercizio 1 (comando `plot`, definizione di anonymous function con stringa e comando `str2func`).
Si desidera visualizzare una funzione data in un certo intervallo $[a, b]$. L'algoritmo è molto semplice:

```
assegna a estremo sinistro
assegna b estremo destro
assegna f funzione
assegna n numero di punti
definisci  $x_i, i = 1 : n$  ascisse
definisci  $y_i = f(x_i), i = 1 : n$  ordinate
disegna la curva per punti  $(x_i, y_i), i = 1 : n$ 
```

Ad esempio, supponiamo di voler disegnare nell'intervallo $[-5, 5]$ la funzione $f(x) = x^2 - 2e^x/x$. La funzione in Matlab viene scritta in ingresso come una stringa (sequenza di caratteri). Pertanto deve essere inserita entro due apici. Alcuni degli operatori da usare nella descrizione sono gli operatori "con punto" (a meno che uno degli operandi non sia una costante). Ovvero:

```
+ Addition
- Subtraction
.* Multiplication
./ Division
.^ Power
```

Quindi la traduzione dell'algoritmo in Matlab è

```
% estremo sinistro
a = -5;
% estremo destro
b = 5;
% definiamo la anonymous function
fun = @(x) x.^2-2*exp(x)./x;
% numero punti
n = 101;
% vettore delle ascisse
x = linspace(a, b, n);
% vettore delle ordinate
y = fun(x);
%
% Disegna la curva nell'intervallo [a,b]
plot(x,y)
```

Per la definizione della funzione abbiamo utilizzato la definizione esplicita della anonymous function. Ma se vogliamo modificare la funzione siamo obbligati a modificare lo script ogni volta. Pertanto vediamo un modo alternativo di definirla anche se può sembrare complicato, ma che ci permette di scrivere uno script maggiormente utile. Invece di definire esplicitamente la anonymous function possiamo usare il comando Matlab `str2func` dando come argomento la stringa di definizione:

```
% definisco la variabile stringa
fs = 'x.^2-2*exp(x)./x';
funs = ['@(x)',fs];
% uso str2func per definire la anonymous function fun
fun = str2func(funs);
```

Si noti che la funzione scelta ha un asintoto verticale in $x = 0$ e si guardi la figura. Ovviamente non essendo uno studio di funzione nelle vicinanze di $x = 0$ la curva non viene rappresentata bene, perchè

quello che Matlab effettua è unire i punti con una spezzata. Ma perchè la parte centrale non è connessa? (Si guardi quanto vale $x(51)$ e $y(51)$).

Se si prova ad aumentare il valore di n ponendolo uguale a 1001 l'aspetto risulta migliore ma la parte centrale risulta ancora non connessa.

Si inserisca ora il valore $n = 100$. Cosa cambia nella figura? Perchè?

Esercizio 2 (commenti, comandi `input`, `disp`, `hold on` e `hold off`, creare ed eseguire uno script).

Vogliamo rendere il nostro script più generale in modo da non doverlo cambiare quando vogliamo considerare un intervallo diverso oppure una funzione diversa. Inoltre vogliamo anche inserire l'asse delle ascisse sulla stessa figura visto che le intersezioni con l'asse x della curva che rappresenta la funzione sono le soluzioni dell'equazione non lineare $f(x) = 0$ presenti nell'intervallo $[a, b]$. Disegnare l'asse delle ascisse corrisponde a tracciare una retta ed, in Matlab, basta unire due punti della retta stessa ottenendo il segmento che unisce tali due punti.

Si consideri pertanto il seguente script che serve a visualizzare una funzione data in un certo intervallo, e lo si trascriva in un file esterno, di nome `disegna.m`:

```
% DISEGNA Script per disegnare una funzione in un intervallo [a,b]
%
% Parte di ingresso dati
a = input(' Dammi l''estremo sinistro: ');
b = input(' Dammi l''estremo destro: ');
fs = input(' Dammi la stringa funzione: ');
n = input(' Dammi il numero di punti: ');
%
disp(' Estremo sinistro'); disp(a)
disp(' Estremo destro'); disp(b)
disp(' Numero di punti'); disp(n)
% uso str2func per definire la anonymous function fun
funs = ['@(x)',fs];
fun = str2func(funs);
disp(' Funzione'); disp(fs)
%
% Definisce il vettore delle ascisse
x = linspace(a, b, n);
%
% Definisce il vettore delle ordinate
y = fun(x);
%
% Disegna la curva nell'intervallo [a,b]
plot(x,y)
% Disegna l'asse x nell'intervallo [a,b]
hold on
plot([a,b],[0,0],'k-')
hold off
```

Poi mettiamoci nella finestra comandi e diamo semplicemente

```
>> disegna
```

oppure usiamo il tasto Run, e forniamo i dati richiesti. Quindi, nel caso del nostro esempio, nella finestra comandi avremo:

```
>> disegna
Dammi l'estremo sinistro: -5
Dammi l'estremo destro: 5
Dammi la stringa funzione: 'x.^2-2*exp(x)./x'
Dammi il numero di punti: 101
Estremo sinistro
-5
```

Estremo destro

5

Numero di punti

101

Funzione

$x.^2-2*\exp(x)./x$

>>

Per visualizzare la funzione abbiamo scritto

```
disp(' Funzione'); disp(fs)
```

Avremmo potuto scrivere la seguente istruzione? Cosa sarebbe cambiato?

```
disp(' Funzione'); disp(fun)
```

Esercizio 3 (salvare una figura in formato `.fig` e `.pdf`).

Si studino graficamente le seguenti funzioni identificando per ognuna di esse tutte le soluzioni (od alcune di esse) dell'equazione $f(x) = 0$ e gli intervalli che ne contengono una ed una sola. Per far pratica in preparazione del test finale, si salvi qualche figura in formato `.fig` (formato del Matlab) e `.pdf`.

$$f(x) = 1 - x - e^{-2x}$$

$$f(x) = 2xe^x - 1$$

$$f(x) = e^{x^2} - 1$$

$$f(x) = x + \log x$$

$$f(x) = x^3 - x - 2$$

$$f(x) = x^2 - 2 - \sin x$$

$$f(x) = 3x - \cos x$$

$$f(x) = \sin x - x^2/2$$

$$f(x) = e^x - 5 + x^2$$

$$f(x) = x^2 - 2x - e^{-x+1}$$

$$f(x) = x^3 - 4x^2 + \log x$$

$$f(x) = 2 + \log(1 + x^2) - x$$

$$f(x) = x^2 + 3 - \frac{\sin x}{x^2}$$

$$f(x) = \log(3x^2 - x + 1) - 4$$

$$f(x) = 6 - (1 + x) \frac{(1 + x)^5 - 1}{x}$$

$$f(x) = e^x - 4x^2$$

$$f(x) = \sqrt{x + 2} - e^{-x}$$

$$f(x) = x^2 + 3 - \tan(x)$$

$$f(x) = (x - 2)(x - 1) \log(x)$$

$$f(x) = 3 + \log(2 + x^2) - x$$

$$f(x) = e^x - 2x^2$$

$$f(x) = \sqrt{x + 1} - e^{-x}$$