

Laboratorio di Calcolo Numerico

Introduzione a Matlab/Octave

Ángeles Martínez Calomardo

[http://www.math.unipd.it/~acalomar/DIDATTICA/
angeles.martinez@unipd.it](http://www.math.unipd.it/~acalomar/DIDATTICA/angeles.martinez@unipd.it)

Laurea in Informatica
A.A. 2018–2019

Matlab

- Prodotto commerciale che fornisce sofisticati strumenti di calcolo.
- È distribuito da The MathWorks (si veda il sito www.mathworks.com).
- La sua principale caratteristica è la manipolazione di matrici, come viene sottolineato dall'acronimo MATLAB che deriva da MATrix LABoratory, vale a dire “laboratorio matriciale”.
- Calcolatrice scientifica evoluta.
- Linguaggio di programmazione ad alto livello.

Per avviare Matlab in ambiente Unix basta digitare il comando `matlab` seguito dal tasto di invio.

Octave

- Anche Octave è un ambiente integrato per il calcolo scientifico e la visualizzazione grafica come Matlab.
- È distribuito gratuitamente dalla GNU (si veda il sito www.octave.org).
- Matlab e Octave presentano delle differenze ma sono sufficientemente compatibili da permettere alla maggior parte di programmi Matlab di essere eseguiti senza modifiche in ambiente Octave e viceversa.

Per avviare Octave in ambiente Unix basta digitare il comando `octave` seguito dal tasto di invio.

**LE INFORMAZIONI CONTENUTE IN QUESTI LUCIDI SONO VALIDE
PER ENTRAMBI I PROGRAMMI**

Interfaccia grafica di Matlab

L'interfaccia grafica di Matlab è costituita da 4 ambienti:

- **Workspace.** Una finestra che mostra il contenuto del workspace (variabili memorizzate e loro valore).
- **Current directory.** Una finestra sulla cartella in cui si sta lavorando, che mostra i files presenti nella cartella stessa.
- **Command history.** Contiene una lista di tutti i comandi digitati.
- **Command window.** Finestra nella quale vengono inseriti i comandi.

L'interfaccia grafica di Octave è costituita degli stessi 4 ambienti con nomi diversi:

- **Variables' list.** Variabili memorizzate e loro valore.
- **Navigator.** Una finestra sulla cartella in cui si sta lavorando, che mostra i files presenti nella cartella stessa.
- **Command List.** Contiene una lista di tutti i comandi digitati.
- **Octave Terminal.** Finestra nella quale vengono inseriti i comandi.

Command window / Octave Terminal

- La **Command window/Octave Terminal** permette di interagire con l'ambiente di calcolo di Matlab/Octave che si presenta come una linea di comando detta *prompt*.
- Permette di eseguire programmi (script e function) presenti in Matlab/Octave, ma anche programmi costruiti dall'utente usando il linguaggio Matlab/Octave e salvati su un file di testo con estensione .m (m-file).
- Per eseguire uno script costruito dall'utente occorre scrivere dopo il prompt il nome del file **senza l'estensione .m**.
- Un programma Matlab/Octave non deve essere compilato: premuto il tasto enter le istruzioni vengono interpretate.
- Per essere eseguiti i programmi devono essere presenti nella cartella di lavoro (current directory).
- Per capire qual è la cartella attuale esiste il comando **pwd**.
- Il comando **what** lista i file .m presenti nella cartella di lavoro, mentre il comando **ls** lista tutti i files presenti nella stessa.

Variabili e assegnazione

- In Matlab/Octave non occorre dichiarare le variabili: l'assegnazione coincide con la dichiarazione.

```
a = 2/3
a =    0.66667
b = 3/2
b =    1.5000
a*b
ans =    1
```

- Matlab/Octave crea le variabili a e b nel momento in cui viene loro assegnato un valore. Se il risultato di un'espressione non viene assegnato a nessuna variabile definita dall'utente, viene assegnato alla variabile di default `ans`.
- I nomi delle variabili possono essere lunghi al massimo 19 caratteri e devono iniziare con un carattere alfabetico (distingue tra maiuscole e minuscole).

Esercizio

Assegnare alla variabile A il valore 1, e scrivere a dopo il prompt. Si osservi che A ed a sono due variabili distinte.

Variabili e assegnazione

- Il comando `who` permette di sapere quali sono le variabili dell'utente attualmente in memoria.
- Il comando `whos` ne mostra anche la dimensione e l'occupazione di memoria (numero di bytes).

`whos`

Variables in the current scope:

Attr	Name	Size	Bytes	Class
	A	1x1	8	double
	a	1x1	8	double
	<code>ans</code>	1x1	8	double
	b	1x1	8	double

Total is 4 elements using 32 bytes

- Le variabili possono essere cancellate utilizzando il comando `clear`.
- Ci sono variabili predefinite come l'unità immaginaria i o il numero π .

Operazioni aritmetiche e funzioni matematiche predefinite

+ addizione
− sottrazione
* prodotto
/ divisione
^ elevamento a potenza

Funzione	function MATLAB
sin	sin
cos	cos
tan	tan
arcsin	asin
arccos	acos
arctan	atan
exp	exp
ln	log
\log_2	log2
\log_{10}	log10
.	abs
$\sqrt{\cdot}$	sqrt

L'istruzione `format`

- Permette di modificare il formato di visualizzazione dei risultati ma **non modifica la precisione con cui i calcoli vengono eseguiti**.
- Tutti i calcoli vengono effettuati in Matlab/Octave utilizzando i numeri in virgola mobile in doppia precisione, secondo lo standard IEEE-754r.
- I principali formati di visualizzazione dei risultati si ottengono digitando `help format`.

Dato il numero $1/7$, alcuni formati comunemente usati sono

<code>format short</code>	produce	0.1429
<code>format short e</code>	produce	1.4286e-01
<code>format short g</code>	produce	0.14286
<code>format long</code>	produce	0.142857142857143
<code>format long e</code>	produce	1.428571428571428e-01
<code>format long g</code>	produce	0.142857142857143

- Gli stessi formati sono disponibili in Octave e forniscono risultati con lievi discrepanze.

Matrici e vettori

- Le variabili per Matlab/Octave hanno una struttura di tipo **matriciale**.
 - ▶ Gli scalari sono considerati matrici 1×1 .
 - ▶ I vettori riga sono matrici $1 \times n$.
 - ▶ I vettori colonna sono matrici $n \times 1$.
- Per definire una **matrice** se ne possono innanzitutto assegnare direttamente gli elementi riga a riga. Ad esempio digitando

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

si produce

```
A =  
    1    2    3  
    4    5    6  
    7    8    9
```

- Notiamo che il punto e virgola separano righe diverse.

Matrici e vettori

- L'elemento in riga i e colonna j di A si accede con $A(i,j)$. Per la matrice A dell'esempio precedente

```
>> A(2,3)
ans = 6
```

Esercizio

Costruire una matrice 2×3 con i primi sei numeri interi come coefficienti. Azzerare gli elementi $A(1,1)$ e $A(2,2)$.

- Soluzione.

```
>> A = [1 2 3; 4 5 6]
A =
     1     2     3
     4     5     6
```

```
>> A(1,1) = 0;
>> A(2,2) = 0;
>> A
```

```
A =
     0     2     3
     4     0     6
```

Estrarre parti di una matrice

Si può fare tramite l'uso del carattere due punti :

Ad esempio, per individuare la seconda riga di A, basta scrivere

```
>> A(2,:) 
```

Per individuare la terza colonna di A si scrive invece

```
>> A(:,3) 
```

- Per estrarre una intera sottomatrice da una matrice assegnata, basta specificare un insieme di righe e colonne. Esempio:

A =

```
1    1    1
-2    3    1
4   -6    7
```

```
>> M=A(2:3,:)
M =
```

```
-2    3    1
4   -6    7
```

Matrici

Matlab allarga una matrice quanto basta per sistemare un elemento dato.

```
>> A=[1 1 1; -2 3 1; 4 -6 7]  
A =
```

```
1 1 1  
-2 3 1  
4 -6 7
```

```
>> A(5,5)=2  
A =
```

```
1 1 1 0 0  
-2 3 1 0 0  
4 -6 7 0 0  
0 0 0 0 0  
0 0 0 0 2
```

```
>> A=A(1:3,1:3)  
A =
```

```
1 1 1  
-2 3 1  
4 -6 7
```

```
>> A(3,6)=9  
A =
```

```
1 1 1 0 0 0  
-2 3 1 0 0 0  
4 -6 7 0 0 9
```

Comandi predefiniti che operano su matrici

- Comandi che generano matrici:

`rand(m,n)` matrice $m \times n$ con coefficienti random.

`eye(n)` matrice identità di ordine n .

`ones(n)` matrice di ordine n con coefficienti tutti uguali ad 1.

`zeros(n)` matrice di ordine n con coefficienti tutti uguali a 0.

- Altri comandi importanti che operano con matrici:

`inv(A)` calcola l'inversa della matrice A ;

`[n,m] = size(A)` restituisce il numero di righe e di colonne di A ;

`det(A)` calcola il determinante di A .

`eig(A)` calcola gli autovalori di A .

Esercizio

Creare una matrice quadrata A di ordine 4 con tutti gli elementi uguali a 1 e calcolarne il determinante.

Che cosa succede se proviamo a calcolare l'inversa di A ?

Comandi predefiniti che operano su matrici

Soluzione dell'esercizio

```
>> A = ones(4)
A =
```

```
1    1    1    1
1    1    1    1
1    1    1    1
1    1    1    1
```

```
>> det(A)
```

```
ans = 0
```

```
>> inv(A)
```

```
warning: inverse: matrix singular to machine precision , rcond = 0
```

```
ans =
```

```
Inf    Inf    Inf    Inf
Inf    Inf    Inf    Inf
Inf    Inf    Inf    Inf
Inf    Inf    Inf    Inf
```

Operazioni tra matrici

- Essendo A, B, C matrici con coefficienti reali ed s uno scalare, si definiscono le operazioni:

$C = s * A$ prodotto di una matrice per uno scalare.

$C = A'$ trasposizione di una matrice.

$C = A + B$ somma di due matrici di dimensione $m \times n$.

$C = A - B$ sottrazione di due matrici $m \times n$.

$C = A * B$ prodotto di A (m righe e n colonne) per B (n righe e p colonne).

$C = A .* B$ $c_{ij} = a_{ij} \cdot b_{ij}$ (prodotto di due matrici componente a componente).

- Esempio:

```
>> A = [2  -1; 3  4; -2  7]
```

```
A =
```

```
    2    -1
    3     4
   -2     7
```

```
>> B = A'
```

```
B =
```

```
    2     3    -2
   -1     4     7
```


Operazioni tra matrici: esempi

Riportiamo un esempio di somma, prodotto, e prodotto componente a componente di due matrici.

```
>> A = [4 -1 0; -1 4 -1; 0 -1 4];  
>> B = [1 2 3; -4 -5 -6; 0 1 2];
```

Il carattere `;` usato alla fine di qualunque istruzione sopprime l'output a video.

```
>> C = A+B
```

```
C =
```

```
    5     1     3  
   -5    -1    -7  
    0     0     6
```

```
>> C = A*B
```

```
C =
```

```
     8     13     18  
   -17    -23    -29  
     4      9     14
```

```
>> C = A.*B
```

```
C =
```

```
     4     -2      0  
     4    -20      6  
     0     -1      8
```

Vettori

- Matlab/Octave tratta i vettori come casi particolari di matrici.
- Per memorizzare il vettore riga $x = [1, 2, 3, 4, 5]$ occorre digitare

```
x = [1 2 3 4 5];
```

- mentre

```
y = [-2; 4 ; 12];
```

produce il vettore colonna

```
y =  
    -2  
     4  
    12
```

- `zeros(1,n)` crea un vettore riga di dimensione n con tutti gli elementi nulli;
`zeros(n,1)` idem per vettori colonna.
- Per creare un vettore riga (colonna) con elementi uguali ad 1 usiamo
`ones(1,n)` (`ones(n,1)`).

Vettori

- La componente i -esima di un vettore si identifica con $x(i)$. Per esempio la terza componente del precedente vettore y sarà:

```
y(3)
ans =
    12
```

- un vettore colonna si trasforma nel corrispondente vettore riga mediante trasposizione:

```
x = [-2; 4; 12] '
x =
    -2     4    12
```

- Si può creare un vettore vuoto (cioè con zero componenti) con il comando $x = []$.
- La norma euclidea di un vettore x si definisce come $\|x\| = \sqrt{x^T x}$. In Matlab/Octave si implementa con il comando $\text{norm}(x)$.

Operazioni tra vettori

- Essendo z, u, v vettori (riga o colonna) ed s uno scalare, si definiscono le operazioni:

$z = s*u$ prodotto di un vettore per uno scalare.

$z = u+v$ somma di due vettori di dimensione n .

$z = u-v$ sottrazione di due vettori n .

$z = u.*v$ $z_i = u_i \cdot v_i$ (prodotto tra due vettori componente a componente).

$z = u./v$ $z_i = u_i / v_i$ (divisione tra due vettori componente a componente).

- Il prodotto scalare tra due vettori colonna x e y di dimensione n :

$s = x^T y = \sum_{i=1}^n x_i y_i$ si esegue in Matlab/Octave come

$s = x'*y$

Esercizio

Si definisca il vettore colonna u di componenti $(1,2)$ e il vettore colonna v di componenti $(3,4)$; si calcoli $u+v$, $u-v$, il prodotto scalare di u per v e il prodotto componente a componente.

Vettori con elementi equispaziati

Notazione “:”

In Matlab/Octave si possono creare vettori riga con elementi equispaziati con la notazione “:” la cui sintassi è

`vettore = [inizio:incremento:fine]`

dove `inizio` è il primo elemento del vettore, e `incremento` è un parametro opzionale che indica la spaziatura tra gli elementi (se omesso `incremento=1`).

Esempio:

```
>> u = [1:2:10]
u =
     1     3     5     7     9
```

```
>> v = [5:-1:1]
v =
     5     4     3     2     1
```

```
>> w = [0:0.2:1]
w =
 0.00000  0.20000  0.40000  0.60000  0.80000  1.00000
```

Crea gli elementi
 $\text{vettore}(i) = \text{inizio} + (i-1) \cdot \text{incremento}$
fino a quando
 $\text{vettore}(i) \leq \text{fine}$

Vettori con elementi equispaziati

Comando `linspace`

- Un'alternativa ai due punti è il comando:
`linspace(inizio,fine,numero di punti)`
- Il comando `linspace` genera un vettore riga con un numero prefissato di punti equispaziati compresi tra `inizio` (primo elemento) e `fine` (ultimo elemento del vettore).
- Se il numero di punti è omesso se ne creano 100.
- Esempi:

```
>> u = linspace(0,8,5)
```

```
u =  
    0    2    4    6    8
```

```
>> v = linspace(-5,5,6)
```

```
v =  
   -5   -3   -1    1    3    5
```

```
>> v = linspace(-5,5,5)
```

```
v =  
 -5.00000  -2.50000   0.00000   2.50000   5.00000
```

Vettori con elementi equispaziati

Esercizio

Creare il vettore u di componenti $u_i = -3 + 0.5 \cdot i$, $i = 0, \dots, 6$, utilizzando sia la notazione `:` che il comando `linspace`.

- Soluzione.

```
>> u = [-3:0.5:0]  
u =
```

```
   -3.00000   -2.50000   -2.00000   -1.50000   -1.00000   -0.50000  
         0.00000
```

```
>> u = linspace(-3,0,7)  
u =
```

```
   -3.00000   -2.50000   -2.00000   -1.50000   -1.00000   -0.50000  
         0.00000
```

- Un'istruzione Matlab/Octave di uso molto comune è `length` che calcola il numero di elementi di un vettore.

```
>> length(u)  
ans = 7
```

Uso vettoriale delle funzioni elementari

- Le funzioni matematiche predefinite in Matlab/Octave sono vettoriali.

Esempi:

```
>> u = [1:1:6]
u =
     1     2     3     4     5     6
>> log(u)
ans =
     0.00000     0.69315     1.09861     1.38629     1.60944     1.79176
>> exp(u)
ans =
     2.7183     7.3891    20.0855    54.5982   148.4132   403.4288
```

- Per le operazioni prodotto e divisione bisogna usare il punto per lavorare vettorialmente.

```
>> u = -1:0.5:1
u =
    -1.0000    -0.5000     0.0000     0.5000     1.0000
>> u.^2
ans =
     1.00000     0.25000     0.00000     0.25000     1.00000
>> 1./u
ans =
    -1     -2   Inf     2     1
```


Uso vettoriale delle funzioni elementari

Esercizio

Calcolare il seno e il coseno dei seguenti angoli: $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$.

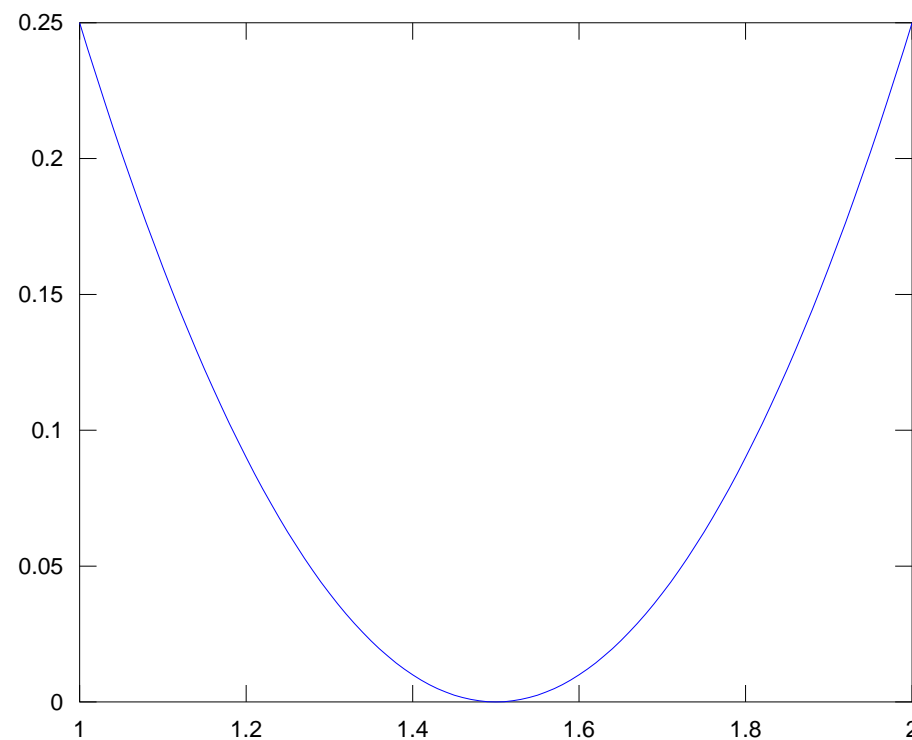
Grafica in Matlab/Octave: comando `plot`

- Il comando `plot(x,y)` traccia il grafico di una serie di dati contenuti in due vettori x (il vettore delle ascisse) e y (il vettore delle ordinate) di lunghezza n .
- I due vettori x e y devono avere **la stessa lunghezza**: il grafico viene disegnato unendo tali punti con dei segmenti.
- Si possono utilizzare tanti comandi opzionali con cui modificare le caratteristiche del grafico:
 - ▶ aggiungere un titolo (`title`),
 - ▶ inserire etichette sugli assi (`xlabel`, `ylabel`)
 - ▶ selezionare tipo di linea, colore e spessore, ecc
- L' `help plot` fornisce tutte le indicazioni a tal proposito.

Grafica in Matlab/Octave: comando `plot`

Esempio 1

```
x = linspace(1,2,101);  
y = (x-1.5).^2;  
plot(x,y)
```



Definizione di funzioni matematiche

Matlab/Octave mette a disposizione le funzioni matematiche predefinite viste in precedenza, ma consente all'utente di definire le proprie funzioni:

- 1 Mediante una stringa di caratteri:

```
f = 'x.^3 - log(x)'
```

- 2 Mediante il comando `inline`:

```
f = inline('x.^3 - log(x)')
```

- 3 Mediante *anonymous function* con l'ausilio del *function handle* `@`:

```
f = @(x) x.^3 - log(x)
```

- 4 Mediante la costruzione di una apposita `function` Matlab salvata in un m-file:

```
function y=fun(x)  
y=x.^3 - log(x);  
end
```

Grafica in Matlab/Octave: comando `plot`

Esercizio

Si disegni nell'intervallo $[-5,5]$ la funzione $f(x) = x^2 - 2e^x/x$ usando il comando `inline` per definire la funzione e il comando `plot` per disegnare la curva per punti.

```
% estremo sinistro
a = -5;
% estremo destro
b = 5;
% definisco la variabile stringa
funs = 'x.^2-2*exp(x)./x';
% definisco la inline function fun
fun = inline(funs);
% 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)
```

Salvare un grafico in Matlab

Comandi `savefig` e `print`

È possibile salvare una figura in un file con estensione `.fig` e volverla a caricare successivamente con i comandi `savefig` e `openfig`, rispettivamente. Il comando

```
savefig( 'grafico' )
```

salva nella cartella di lavoro il file chiamato `grafico.fig` contenente l'ultimo grafico previamente creato. Tale grafico può essere importato nuovamente nell'ambiente di lavoro di Matlab mediante:

```
openfig( 'grafico' )
```

Diversamente, è possibile salvare un grafico in un file esterno, in uno dei formati per immagini, `.pdf` o `.tif` o altri (si consulti la help) mediante il comando `print`. La sintassi per salvare in formato pdf il grafico presente nella finestra grafica numero uno sarebbe la seguente:

```
print( '-f1' , 'nomefile' , '-dpdf' )
```

Grafica in Matlab/Octave: comando `plot`

Esempio 2: uso di `hold on` e `hold off`

```
x=linspace(0,2*pi,100);
```

```
plot(x,sin(x),'r-');
```

```
hold on;
```

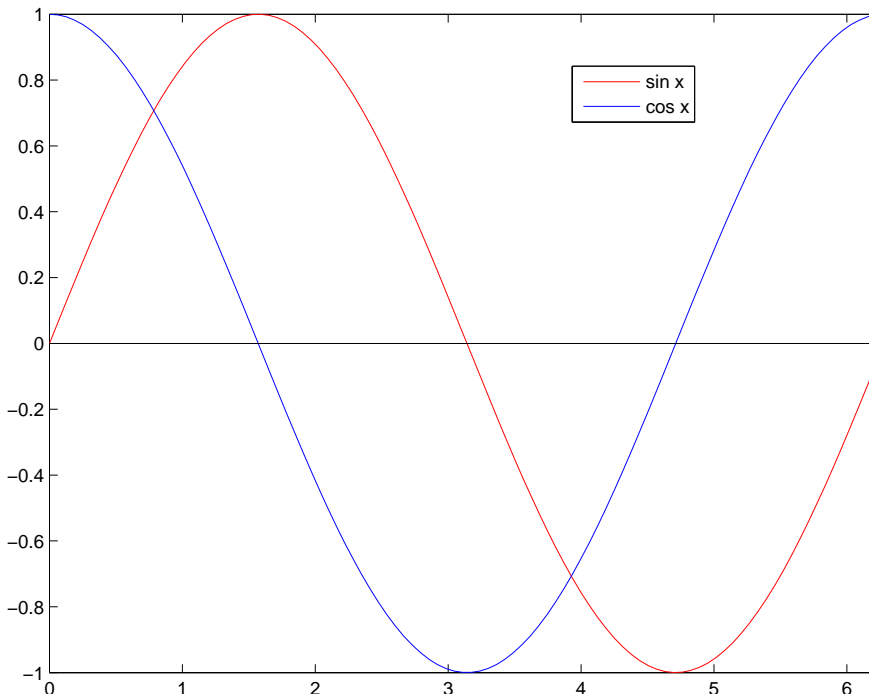
```
plot(x,cos(x),'b-');
```

```
plot([0 2*pi],[0,0],'k-');
```

```
hold off
```

```
legend('sin x','cos x');
```

```
axis([0 2*pi -1 1])
```



Individuazione delle radici di funzioni non lineari

Scrittura $f(x) = 0$ oppure $x = g(x)$

Data l'equazione

$$f(x) = \exp(-x) - \cos(x) + 3x^2 = 0$$

possiamo riscriverla in maniera matematicamente equivalente nella forma $x = g(x)$ in molti modi diversi, ad esempio:

$$x = g_1(x) = -\log(\cos(x) - 3x^2) \quad (1)$$

$$x = g_2(x) = \arccos(\exp(-x) + 3x^2) \quad (2)$$

$$x = g_3(x) = \sqrt{\frac{1}{3}(\cos(x) - \exp(-x))} \quad (3)$$

Esercizio

Disegnare le tre funzioni g_1 , g_2 e g_3 e la bisettrice del primo e terzo quadrante, per individuare le radici della funzione f nell'intervallo $[-0.2, 0.4]$.

Si disegni inoltre, nella stessa finestra grafica, la funzione f e l'asse x e si verifichi che l'intersezioni della f con l'asse x coincidano con i punti fissi evidenziati al punto precedente.