

```

a = pi;
a

a =

    3.1416 //il programma mostra solo 4 cifre decimali

format long //posso modificare il tipo di a per far mostrare + cifre
a

a =

    3.141592653589793

format short //di default usa questo formato, usa un arrotondamento. Modificano
solo i formati di visualizzazione, non di memorizzazione!!!
a

a =

    3.1416

format shorte //short exponential
a

a =

    3.1416e+00

format longe //long exponential
a

a =

    3.141592653589793e+00

```

```

whos
  Name      Size      Bytes  Class    Attributes

  a         1x1         8    double //double è il formato di default, 8
byte

```

//INIZIO HELP

help round //round si utilizza specificando che cio che stiamo andando ad arrotondare ha un numero precisato di cifre significative

round rounds towards nearest decimal or integer

round(X) rounds each element of X to the nearest integer.

//in questo caso se non gli diamo il numero di cifre significative, arrotonda al numero intero + vicino

round(X, N), for positive integers N, rounds to N digits to the right

of the decimal point. If N is zero, X is rounded to the nearest integer.

If N is less than zero, X is rounded to the left of the decimal point.

N must be a scalar integer.

//in questo caso invece arrotonda basandosi sul numero di cifre che gli diamo, senza normalizzarlo

round(X, N, 'significant') rounds each element to its N

most significant

digits, counting from the most-significant or left side of the number.

N must be a positive integer scalar.

//noi usiamo questo comando, perché normalizza e arrotonda

`round(X, N, 'decimals')` is equivalent to `round(X, N)`.

For complex X, the imaginary and real parts are rounded independently.

Examples

% Round pi to the nearest hundredth

```
>> round(pi, 2)
    3.14
```

% Round the equatorial radius of the Earth, 6378137 meters,
% to the nearest kilometer.

```
round(6378137, -3)
    6378000
```

% Round to 3 significant digits

```
format shortg;
round([pi, 6378137], 3, 'significant')
    3.14      6.38e+06
```

If you only need to display a rounded version of X,
consider using `fprintf` or `num2str`:

```
fprintf('%.3f\n', 12.3456)
    12.346
fprintf('%.3e\n', 12.3456)
    1.235e+01
```

See also [floor](matlab:help floor), [ceil](matlab:help ceil), [fprintf](matlab:help fprintf).

[Documentation for round](matlab:doc round)
<a

`round`)>Other functions named `round`
//FINE HELP

fla = round(a,4,'significant')//arrotondo il nostro a con 4 cifre significative

fla =

3.1420000000000000e+00

*//quindi, questa operazione di round è il nostro float, che dobbiamo fare dopo
ogni operazione, per ottenere il risultato arrotondato correttamente*

*//il risultato poi viene visualizzato nella maniera indicata, short, long,
shorte,longe*

eps //l'epsilon di macchina è una variabile predichiarata su matlab

*//è il + piccolo numero che devo aggiungere ad 1 in modo che il troncamento di 1
+ eps sia maggiore di 1*

ans =

2.220446049250313e-16

format short

eps

```

ans =

    2.2204e-16
//inoltre, l'epsilon di macchina è = a 1-t, e t, in caso della precisione double
è 53

2^(1-53)

ans =

    2.2204e-16 //identico a eps

eps/100 //dimostriamo che eps non è il numero + piccolo rappresentabile!

ans =

    2.2204e-18

2^(-1022-53) //per avere l'errore di underflow devo scrivere un numero + piccolo
del numero + piccolo memorizzabile: 2^(1-t) (perché beta è 2)

ans =

    0 //risultato 0==underflow

2^(-1021-53)

ans =

    4.9407e-324

clear

-----SCRIPT: machine_eps - CALCOLO EPSILON DI MACCHINA-----
//Usiamo la definizione di epsilon di macchina: il + piccolo numero che aggiunto
ad 1 e troncato, mi dà un numero maggiore di 1---la distanza tra il numero 1 e
il successivo numero di macchina

//devo usare un ciclo while

ep = 1;
x = 1 + ep; //devo far decrescere ep finchè x non sia uguale a 1
k=0; //inizializzo un contatore per contare quanti passi faccio
while (x > 1) //fino a quando x>1, faccio decrescere ep
    ep = ep/2; //far decrescere ep significa farlo diminuire di un ordine di grandezza
//diminuire di un ordine di grandezza in base 10 significa dividere per 10, in base 2
significa dividere per 2
    x = 1+ep; //aggiorno il valore di x
    k = k+1; //incremento il contatore
end

ep = ep*2; //l'ep risultante non rispecchia le caratteristiche che vogliamo, a noi
serve il valore precedente all'uscita del while. Moltiplichiamo per 2 per ottenerlo.

//k in uscita è t, il numero di cifre significative, l'ultima rappresentazione della
mantissa. Dopo di questo il numero è troppo piccolo e vado in underflow
-----

machine_eps
ep

```

```

ep =

    2.2204e-16 //identico ad eps

eps

ans =

    2.2204e-16

k

k =

    53 //stesso valore del nostro t

clear

-----SCRIPT: num_macchina - SIMULAZIONE CALCOLO CON NUM MACCHINA-----

//Chiediamo i valori in ingresso all'utente
a = input('Inserisci il valore di a:');
b = input('Inserisci il valore di b:');
c = input('Inserisci il valore di c:');

//effettuiamo un float ad a,b,c con 4 cifre significative
fla = round(a,4,'significant');
flb = round(b,4,'significant');
flc = round(c,4,'significant');

%calcolo (a+b)-c
d = fla+flb; //sommo a + b flottati
fld = round(d,4,'significant'); //faccio float della somma
t1 = fld - flc; //faccio la differenza tra fl(fla+alb) e flc
flt1 = round(t1,4,'significant'); //flotto il risultato finale
%calcolo a+(b-c)
e = flb - flc; //salvo in una var la differenza tra b e c flottati
fle = round(e,4,'significant'); //flotto il risultato
t2 = fla + fle; //faccio la somma tra fla e fl(flb-flc)
flt2 = round(t2,4,'significant'); //flotto il risultato finale

x = a+b-c; //mi serve il valore vero per calcolare l'errore relativo
%errori relativi rispetto a x valore vero (val assoluto risultato flottato-valore
vero)/(val assoluto del valore vero)
rho1 = abs(flt1-x)/abs(x); //abs fa il valore assoluto
rho2 = abs(flt2-x)/abs(x);

-----

num_macchina
Inserisci il valore di a:374.567
Inserisci il valore di b:982.7123
Inserisci il valore di c:67.987788
flt1

flt1 =

    1289

```

```

flt2

flt2 =

    1289
//flt1 e flt2 hanno lo stesso valore, ma non è sempre così!

//calcoliamo gli errori rho1 e rho2
rho1

rho1 =

    2.2610e-04

rho2

rho2 =

    2.2610e-04

X //calcoliamo il valore vero

x =

    1.2893e+03

format short
x

x =

    1.2893e+03

clear

-----SCRIPT: limite - ERRORI DI CANCELLAZIONE-----

//limiti si possono fare in 2 modi in matlab:
    -ciclo for: si fa avanzare con un numero fissato
    -ciclo while: si fa avanzare fino ad un certo punto preciso
//noi useremo il ciclo for perché ci serve far avanzare con tot passi decisi da noi
//la variabile x parte da 10^-1 e possiamo portarla fino a 10^-20 che è molto sotto l'epsilon di macchina
Ricordiamo eps=2.2204e-16<<10^-20

for i = -1:-1:-20 //posso far variare solo l'esponente, e non tutto il numero x.
    //Inizializzo una variabile i che parte da -1 a -20
    //for i=val iniziale : passo : val finale
    x = 10^i; //mi calcolo la x
    f1 = (1-cos(x))/x^2 //inserisco la formula del limite1
    f2 = 1/2*((sin(x/2))/(x/2))^2 //inserisco la formula del limite2
    pause //serve per vedere tutti i passi che andranno stampati
//un'alternativa è mettere degli fprintf
end
-----

```

```
limite
```

```
f1 =
```

```
0.4996
```

```
f2 =
```

```
0.4996
```

```
//se ci sono degli 0 dopo il 5 significa che dopo ci sono dei valori diversi da 0
```

```
f1 =
```

```
0.5000
```

```
f2 =
```

```
0.5000
```

```
f1 =
```

```
0.5000
```

```
f2 =
```

```
0.5000
```

```
f1 =
```

```
0.5000
```

```
f2 =
```

```
0.5000
```

```
f1 =
```

```
0.5000
```

```
f2 =
```

```
0.5000
```

```
f1 =
```

```
0.5000
```

```
f2 =
```

```
0.5000
```

```
//al passo successivo vediamo come f1 cambia
```

f1 =

0.4996

f2 =

0.5000

//f1 è quello con la sottrazione, mentre f2 è quello senza. Il condizionamento tende ad infinito

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0

f2 =

0.5000

f1 =

0


```
f2 =
```

```
0.5000
```

```
f1 =
```

```
0
```

```
f2 =
```

```
0.5000
```

```
//lo script viene stoppato con ctrl+c
```

```
diary off
```