

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi

malcangi@di.unimi.it

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Interfaccia grafica

L'interfaccia grafica di Matlab consiste principalmente di tre finestre:

Workspace

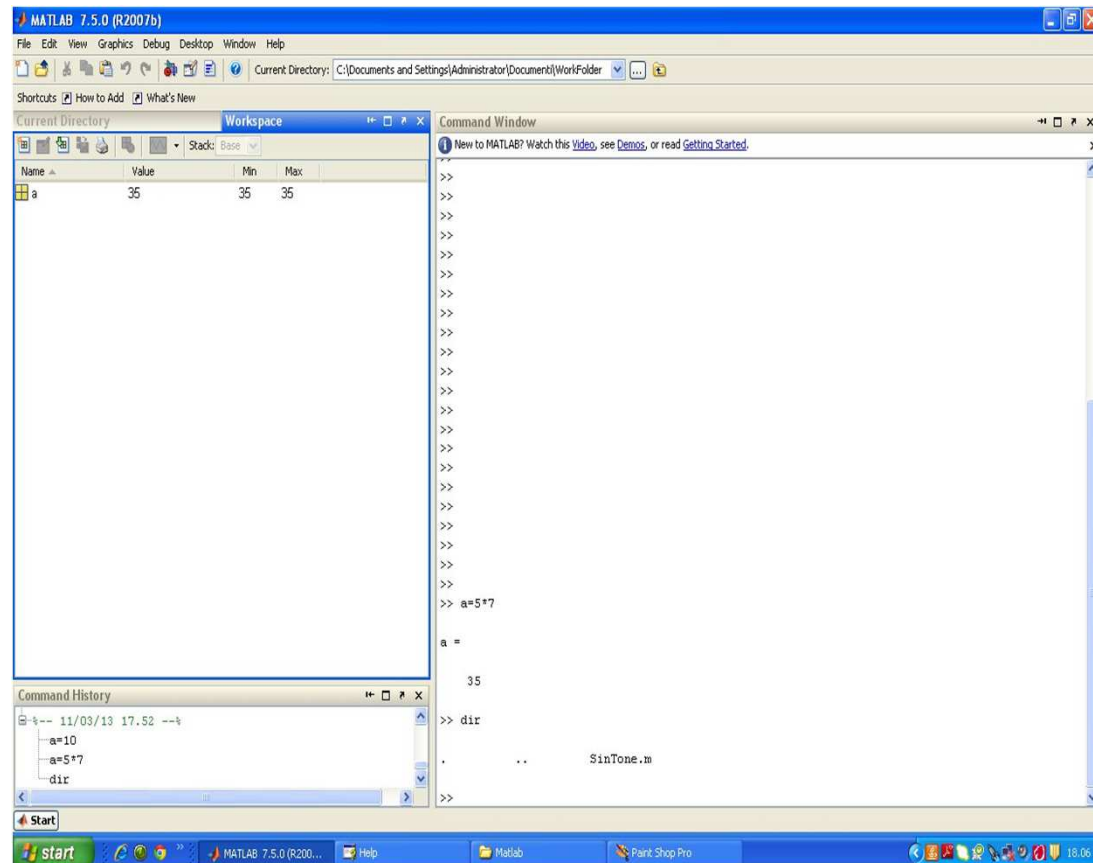
Command History

Command Window

Il **Workspace** visualizza tutto lo stato di allocazione dei dati. Questi compaiono in questa finestra appena creati, sono visualizzabili e modificabili relativamente al contenuto.

La **Command Window** consente di inserire comandi (sia di sistema, sia di linguaggio) eseguibili da Matlab. Questa consente anche di visualizzare messaggi di sistema e risultati dell'esecuzione delle istruzioni del linguaggio (è l'equivalente del I/O standard).

La **Command History** visualizza la sequenza di comandi immessi nella **Command Window** (log).



Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Come funziona

- ✓ Un programma Matlab (detto anche Script), è una sequenza mista di istruzioni di programmazione e di comandi di sistema.
- ✓ Le istruzioni di programmazione sono simil-C. E' bene tenere presente comunque che Matlab è un linguaggio di programmazione matriciale e non scalare, quindi la sua variabile tipica è la matrice. Lo scalare è considerato una matrice 1x1.
- ✓ Un' altra particolarità di Matlab è che gli array sono indicizzati a partire da 1 (e non da 0 come nel linguaggio C). Questo, soprattutto per il trattamento dei segnali non è naturale e porta a possibili errori di indicizzazione (per esempio, al tempo $t=0$ corrisponde un indice $n=1$).
- ✓ I commenti in Matlab sono preceduti dal simbolo %, quindi, qualsiasi linea di testo preceduta da tale simbolo viene ignorata durante l'esecuzione.
- ✓ Matlab è un linguaggio interpretato e valuta le istruzioni al momento dell'esecuzione.

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Editing dei programmi

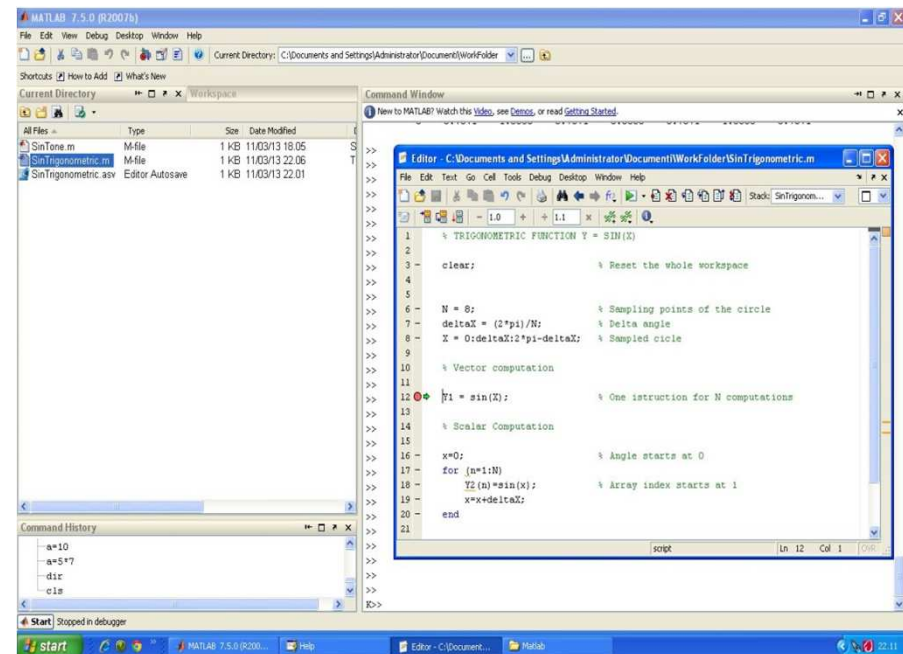
Matlab fa riferimento per default alla Current Directory che considera directory di lavoro. In questa directory devono essere presenti tutti i files cui fa riferimento il programma. I programmi eseguibili di Matlab hanno stensione .m

Anche se la linea di comando della Command Window consente di eseguire qualsiasi istruzione del linguaggio di modellazione Matlab, la sequenza di programmazione va editata in un file di testo .m

I file di testo .m permettono di scrivere la sequenza di programmazione in modo che questa possa essere eseguita in free-running mode oppure in debug mode.

In debug mode i principali strumenti di controllo dell'esecuzione sono i breakpoint (attivabili o disattivabili a fianco della linea di istruzione) e gli step. La visualizzazione del valore delle variabili è possibile o tramite il workspace o posizionando il cursore sulla variabile.

- ✓ Il programma editato e salvato con un nome (nome.m) può contenere sia istruzioni di calcolo e controllo del flusso di programma, sia script, cioè sequenze di comandi dell'ambiente MATLAB.
- ✓ Il nome del file serve a referenziare il programma per la sua esecuzione tramite la command window, oppure all'interno di altri programmi



Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Particolarità del linguaggio

- ✓ Matlab è un linguaggio di programmazione simile al linguaggio «C» con alcune differenze formali ed alcune sostanziali.
- ✓ Utilizza un linguaggio di programmazione interpretato
- ✓ Le variabili sono fondamentalmente di tipo vettoriale multidimensionale (matrici)
- ✓ Gli indici delle matrici hanno come minimo valore «1»
- ✓ Le variabili scalari sono trattate come caso particolare di matrici con un solo elemento
- ✓ I calcoli e le variabili sono in doppia precisione
- ✓ Le variabili vengono create al momento in cui sono referenziate nel programma e il relativo spazio di memoria è statico (fino alla cancellazione fisica con il comando «clear»)
- ✓ E' un linguaggio di programmazione vettoriale, quindi consente di trattare dati vettoriali con operazioni scalari
- ✓ Utilizza operatori utili al trattamento di dati vettoriali, per esempio «:».

```
>> A = [10, 20, 30; 100, 200, 300]
```

```
A =
```

```
    10    20    30  
   100   200   300
```

```
>> size(A)
```

```
ans =
```

```
     2     3
```

```
>>
```

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Principali comandi

- ✓ «**help**» – consente di avere in linea le spiegazioni su una istruzione/comando
- ✓ «**clear**» - consente rimuovere le variabili allocate in memoria
- ✓ «**;**» - chiude una linea di istruzione/comando e inibisce l'output sulla command window
- ✓ «**:**» - definisce una gamma di valori da .. a.
- ✓ «**%**» - commenta l'intera riga
- ✓ «**plot**» – visualizza graficamente i dati
- ✓ «**play**» - consente di riprodurre in forma audio i dati
- ✓ «**size**» – fornisce le dimensioni di un dato multidimensionale (matrice)
- ✓ «**length**» – fornisce le dimensioni di un dato monodimensionale (vettore)

```
>> B = 1:10

ans =

     1     2     3     4     5     6     7     8     9    10

>> length(B) % numero degli elementi di B

ans =

    10

>> size(B) % dimensioni di B

ans =

     1    10

>> B=B+3;

>>
```

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Variabili e funzioni predefinite

Alcune variabili e funzioni ricorrenti sono predefinite come elementi del linguaggio:

- ✓ «**pi**» - costante pigreco: 3.14.... radianti
- ✓ «**Inf**» qualsiasi numero diviso 0
- ✓ «**NaN**» - not a number: qualsiasi numero indeterminato
- ✓ «**sin, cos, tan asin, acos**» - funzioni trigonometriche
- ✓ «**sinh, cosh, asinh, acosh**» – funzioni iperboliche
- ✓ «**sqrt, log, log10, exp**» - funzioni non lineari
- ✓ «**mod**» – resto divisione intera (modulo)

```
>> pi  
  
ans =  
  
    3.1416  
  
>> sin(pi/4)  
  
ans =  
  
    0.7071  
  
>> rem(7, 4)  
  
ans =  
  
     3  
  
>>
```


Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Controllo del flusso di programma (cicli iterativi)

I principali cicli iterativi sono:

✓ Ciclo «**for**»

- ✓ Step default : (Es: n=1:10
incrementa di +1)
- ✓ Step qualsiasi (Es.n=1:2:10
incrementa di 2)

✓ Ciclo «**while**»

- ✓ Step default : (Es: n=1:10
incrementa di +1)
- ✓ Step qualsiasi (Es.n=1:2:10
incrementa di 2)

```
a=0;  
for n=1:10  
    a=a+B(n);  
end
```

```
n=1;  
a=0;  
while (x <11)  
    a=a+B(n);  
end
```

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Controllo del flusso di programma (istruzioni condizionali)

I principali cicli iterativi sono:

✓ Condizione semplice «**if**»

Operatori: < > <= >= == ~=

~ «NOT» && «AND» || «OR»

✓ Condizione multipla «**switch**»

```
if x == 0
    x=Inf;
elseif
    x=y/x;
end


---


switch x
    case 1
        disp ('massimo');
    case 0
        disp ('neutro');
    case -1
        disp ('neutro');
end
```

Informatica Applicata al Suono

Tutorial Matlab

Prof. Mario Malcangi – malcangi@di.unimi.it

Visualizzazione e riproduzione audio

✓ Output grafico «**plot**»

✓ Output audio «**wavplay**»

```
Fs = 44100;
```

```
Ts=1/Fs;
```

```
for n=0:Fs
```

```
    x(n+1) = sin(2*pi*440*n*Ts);
```

```
end
```

```
plot (x);
```

```
wavplay(x, Fs);
```