

per avviare Mathematica:

math per avviare il kernel

mathematica per avviare il notebook

una sequenza di comandi di Mathematica:

scritta in un file *filename.m*

salvata in una pagina di notebook *filename.nb*

per caricare un file:

« *filename.m*;

per scrivere in un file:

expr » *filename.res*

Write["*filename.res*", *expr*];

help on-line:

??*comando*

??**partedelnome**

= indica l'assegnazione (es. $a=3$)
==, != stabiliscono la relazione
=== verifica l'uguaglianza; True/False
!= verifica la disuguaglianza; True/False

da una lista

prova = { $a, 3, pippo$ }

il secondo elemento è prova[[2]]

una sostituzione ha la struttura

expr /. { $a \rightarrow b$ }

Operazioni di comune utilizzo

calcolatore scientifico e simbolico

funzioni algebriche e trigonometriche

definite sul campo complesso: $z=x+I y$

definizione di una funzione (prima versione)

esempio: $f[x_] := a x^2 + b x + c$;

derivate di una funzione

comandi: *D, Dt, Derivative*

somme, prodotti e liste

comandi: *Sum, Product, Table*

espansioni in serie

comandi: *Normal, Series*

integrali elementari

comandi: *Integrate*

vettori e matrici

comandi: *Dot, Eigensystem, Inverse, Det*

soluzione di (sistemi di) equazioni

comandi: *Solve*

soluzione di (sistemi di) equazioni differenziali

comandi: *DSolve*

soluzione numerica di integrali

comandi: *NIntegrate*

soluzione numerica di (sistemi di) equazioni differenziali

comandi: *NDSolve*

grafici bi- e tridimensionali

comandi: *Plot*, *Plot3D*, *ContourPlot*

come salvare un grafico in un file

comandi: *Export*

/usr/local/mathematica/AddOns/StandardPackages,

/usr/local/mathematica/AddOns/ExtraPackages/

/usr/local/mathematica/AddOns/Utilities/

contengono materiale addizionale rispetto alle funzioni elementari predefinite

Esempi

$z = 2 + 3 I$; FullForm[z]; Log[z]; N[Log[z]];

D[Sin[Sqrt[x]],{x,2}];

g[x_,y_] := Derivative[1,2][f][x,y];

Integrate[f'' [x], x]; FullForm[f'' [x]];

Sum[$x^n/n!$, {n,3,Infinity}];

Table[a[i],{i,1,5}]]];

Normal[Series[f[x],{x,x₀,5}]]];

Con quanti termini è necessario sviluppare Log[1+x] nell'intorno di x=0, in modo da valutare Log[0.1] con una accuratezza dello 0.01 per cento?

Integrate[$x^{a-1} (1-x)^{b-1}$, {x,0,1}]]];

Una matrice 2x2 si scrive $M = \{ \{ a,b \}, \{ c,d \} \}$;

Il prodotto tra due matrici si indica con un punto:

m1.m2

Verificare le regole di commutazione delle matrici di Pauli.

```
Solve[ a x2 + b x + c == 0, x];
```

```
M = {{ a,b }, { c,d }}; X = { x1,x2 };
```

```
Y1 = { 0,0 }; Y2 = { 3,5 };
```

```
Solve[ M.X==Y1, X]; Solve[ M.X==Y2, X];
```

```
DSolve[ y'' [x] + 4 y[x] == 0, y[x], x];
```

```
2 Integrate[ 1/Sqrt[1-x2],{x,0,1}];
```

```
2 NIntegrate[ 1/Sqrt[1-x2],{x,0,1}];
```

```
ripetere con WorkingPrecision -> 32
```

```
NDSolve[ { y'' [x]+4 y[x] ==0, y[0] ==1, y'[0] == 0 }, y[x], {x,0,5}];
```

chiamare *sol* la soluzione di NDSolve

```
f = y[x] /. sol[[1]]
```

```
Plot[ f,{x,0,5}]
```

```
g3 = Plot3D[ Sin[x2+y2] , {x,-3,3},{y,-3,3} ];
```

```
f[a_,b_,c_] := Show[ g3, ViewPoint-> {a,b,c} ]
```

```
final = f[0,1,0];  
Export[ "final.eps" ,final," EPS"];  
Export[ "final.jpg" ,final," JPEG"];
```

Esercizi

disegnare la funzione Gamma
separatamente parte reale ed immaginaria
per valori:

reali dell'argomento (2D)

complessi dell'argomento (3D)

combinare i quattro grafici in un'unica figura

Esercizio:

Integrazione analitica, numerica, ContourPlot

```
int0 =  
Sin[th] Cos[th]/r3 1/Sqrt[r2 +rp2 -r rp Cos[th] ];
```

```
ir0 = Integrate[ int0, {r,eps,Infinity}];
```

```
it0[Th_,Rp_,epsilon_] :=  
ir0 /. {th->Th, rp->Rp, eps->epsilon};
```

```
cp0 = ContourPlot[
```

```
Rp = Sqrt[x2+z2];
```

```
pippo =
```

```
NIntegrate[ it0[th, Rp, 0.1],{th,0,Pi}];
```

```
pippo,
```

```
{x,-100.01,100}, {z,-100,100.01},
```

```
PlotPoints->15 ];
```


Programmazione in Mathematica

Mathematica può essere utilizzato come:

- calcolatore scientifico/simbolico
- manipolatore di formule e/o oggetti grafici

In Mathematica l'unità elementare è

l' "*espressione*"

Un'*espressione* può essere:

un numero

un simbolo

una stringa

una lista

una regola di sostituzione

un oggetto grafico

un file

uno stream

...

Il tipo di *espressione* può essere ottenuto con il comando *Head*

In Mathematica QUALUNQUE enunciato ha la struttura:

$$\text{Op}[el1,el2,\dots]$$

dove Op indica un operatore, mentre el1, el2,... sono una lista di argomenti su cui l'operatore agisce.

Un elemento può a sua volta avere questa struttura (strutture ad albero).

Ogni espressione descritta in termini dei suoi livelli e del numero delle sue "foglie".

La struttura di un enunciato può essere ottenuta esplicitamente con il comando *FullForm*.

Mathematica è un programma di manipolazione algebrica:

- effettua in modo formale operazioni su oggetti
- è possibile introdurre operatori che godono di proprietà algebriche (associatività, commutatività, distributività,...)

Ogni operazione modifica una o più parti di una espressione.

Questo risultato viene ottenuto con:

- regole di sostituzione,
- applicazione di funzioni.

Le due possibilità si differenziano per i modi e i tempi in cui l'espressione viene modificata.

Pattern

Si chiama *pattern* un modo di rappresentare in maniera generica una o più espressioni di Mathematica .

x_* : una espressione di Mathematica ;

x_{**} : una o più espressioni di Mathematica .

x_{***} : nessuna, una o più d'una espressione.

$\text{MatchQ}[\text{expr}, \text{pattern}] = \text{True/False}$ a seconda che expr soddisfi o meno il pattern assegnato.

es.: $\text{MatchQ}[a+b, x_*] = \text{True}$

$\text{MatchQ}[a+b, \text{Plus}[x_*, y_*, z_*]] = \text{False}$

$\text{MatchQ}[a+b, \text{Plus}[x_*, y_*, z_{**}]] = \text{True}$

x_h, x_{**h}, x_{***h} sono una o più espressioni aventi h come Head

es.: $\text{MatchQ}[5, x_{\text{Integer}}] = \text{True}$

$\text{MatchQ}[5, x_{\text{Real}}] = \text{False}$

Regole di sostituzione: $a \rightarrow b$

`FullForm[a -> b] = Rule[a,b]`

E' comodo accorpare diverse sostituzioni in una lista

`rules = {a1 → b1, a2 → b2, ...}`

`FullForm[expr /. rules] = Replace[expr, rules]`

oppure

`FullForm[expr //. rules] = RepeatedReplace[expr, rules]`

Nel secondo caso le regole vengono riapplicate fin quando il risultato non rimane invariato.

Le regole di sostituzione agiscono solo quando vengono applicate con un `Replace`.

Definizioni immediate e ritardate

Valutazione immediata di una definizione:

```
x=3;  
f[x_] = a x2 + b;  
Print[f[y]];
```

Valutazione ritardata di una definizione:

```
x=3;  
g[x_] := a x2 + b;  
Print[g[y]];
```

L'assegnazione e la sostituzione possono essere immediate o ritardate (=, :=, ->, :>).

La presenza del : indica che il secondo membro deve venire valutato SOLO quando la definizione viene utilizzata.

Nel secondo caso la variabile x=3 e la variabile x della definizione sono per Mathematica due simboli completamente distinti.

Funzioni:

- abbreviazioni di lunghe formule;

$f[x_] := a x^2 + b x + c;$

$sp[x_List, y_List] :=$

$Sum[x[[i]] y[[i]], \{ i, 1, Length[x] \}];$

$s[x_SurfaceGraphics, a_Real, b_Real, c_Real] :=$

$Show[x, ViewPoint -> \{a, b, c\}]$

- procedure da applicare ad una espressione.

per agire sulla Head:

$Apply[f, \{a, b, c\}] = f @@ \{a, b, c\} = f[a, b, c]$

per agire sugli elementi:

$Map[f, \{a, b, c\}] = \{f[a], f[b], f[c]\};$

$Map[f, \{\{a_1, a_2\}, \{b_1, b_2\}, \{c_1, c_2\}\}] =$

$\{f[a_1, a_2], f[b_1, b_2], f[c_1, c_2]\};$

$f = (\#[[2]]) \&;$

$Map[f, \{\{a_1, a_2\}, \{b_1, b_2\}, \{c_1, c_2\}\}] = \{a_2, b_2, c_2\};$