

# Introduzione all'uso di Mathematica

Marco Bramanti

Questi appunti sono scritti per chi vuole imparare velocemente a usare *Mathematica* per fare le più comuni operazioni che hanno a che fare con il contenuto dei corsi di Analisi Matematica 1 e 2. Scorrendo il Sommario qui sotto saprete subito cosa potrete imparare e cosa no, leggendo questi appunti.

Facendo le operazioni che sono spiegate qui, vi capiterà di chiedervi come potreste farne molte altre, che qui non sono spiegate. Le risposte a queste domande si trovano nei manuali ufficiali di *Mathematica*, che sono i vari libri di Stephen Wolfram, il creatore del programma: c'è un libro per ogni versione di *Mathematica* (attualmente, ottobre 2000, sono in circolazione le versioni 2, 3 e 4). Ciascuno di questi libri ha dalle 900 pagine in su, il che potrebbe scoraggiare qualcuno a consultarli: in realtà sono libri ben scritti, in cui non è difficile orientarsi. Comunque, questi appunti dovrebbero servire proprio per evitare di sfogliare i manuali, a chi desideri imparare in fretta l'essenziale.

## Sommario

### 1. Nozioni di base su Mathematica

(Conviene leggere integralmente questa prima parte)

- 1.1. Primo impatto
- 1.2. Calcoli numerici
- 1.3. Funzioni elementari
- 1.4. Costanti notevoli
- 1.5. Definire una funzione o un numero
- 1.6. Matematica discreta
  - 1.6.1. Successioni, tabulazioni
  - 1.6.2. Sommatorie

### 2. Argomenti specifici di Analisi Matematica

(Sono abbastanza indipendenti gli uni dagli altri, si può leggere solo quello che interessa; richiedono però le Nozioni di base)

- 2.1. Serie numeriche
- 2.2. Numeri complessi
  - 2.2.1. Radici n-esime in C
- 2.3. Risoluzione di equazioni, esatta o approssimata
- 2.4. Grafici di funzioni
- 2.5. Funzioni definite a tratti
- 2.6. Uso dei comandi Sum e Table con funzioni
- 2.7. Calcolo infinitesimale per funzioni di una variabile
  - 2.7.1. Calcolo dei limiti
  - 2.7.2. Calcolo delle derivate

- 2.7.3. Formula di Taylor
- 2.7.4. Calcolo delle primitive
- 2.7.5. Calcolo degli integrali definiti, esatto o approssimato
- 2.7.6. Integrali generalizzati
- 2.7.7. Funzioni integrali
- 2.7.8. Equazioni differenziali
- 2.7.9. Serie di Fourier
- 2.8. Funzioni di più variabili (questa parte non è ancora stata scritta)
  - 2.8.1. Grafici di funzioni reali di due variabili
  - 2.8.2. Grafici di linee in forma parametrica, nel piano o nello spazio
  - 2.8.3. Grafici di superfici in forma parametrica
  - 2.8.4. Calcolo di integrali doppi, esatto o approssimato

## 1. Nozioni di base su Mathematica

### ■ 1.1. Primo impatto

All'avvio di Mathematica compare una finestra principale bianca (Untitled), più eventualmente altre finestre con raccolte di simboli e comandi. Non ostante l'aspetto grafico sia quello di un comune programma windows (con menù ecc.), per fare qualcosa di significativo non basta cliccare sui vari menù con il mouse: occorre usare la tastiera e digitare istruzioni secondo una certa sintassi. Quanta sintassi sia necessario imparare dipende quindi da quale e quanta matematica si vuole fare con Mathematica. Più precisamente, per fare qualsiasi operazione, si digita dalla tastiera una opportuna stringa di testo (input) nella finestra principale, e quindi si preme **shift+invio**: *Mathematica* a questo punto interpreta come input quello che si è appena scritto (più precisamente, quello che è scritto nel blocco di righe in cui sta lampeggiando il cursore), esegue quanto richiesto e fornisce il risultato (output): un numero, un'espressione, un grafico..., a seconda di cosa si è chiesto. L'input va digitato secondo una precisa sintassi: in particolare, sono importanti le maiuscole/minuscole, gli spazi, l'uso corretto delle parentesi graffe, quadre e tonde.

Ricordiamo che sulla tastiera italiana:

le parentesi quadre [ ] si trovano sulla tastiera, ma sono il *terzo* carattere di un tasto, perciò si ottengono premendo quel tasto insieme ad AltGr;

le parentesi graffe { } non ci sono sulla tastiera, ma si ottengono digitando i numeri 123 e 125 col tastierino numerico, mentre si tiene premuto il tasto Alt.

Nella versione 2 di Mathematica, gli input si digitano esclusivamente da tastiera. Nelle versioni 3 o 4 invece, ci sono varie barre di strumenti cliccando sui quali si può immettere almeno parte dell'input, usando un po' meno sintassi. In questi appunti si insegna a usare solo la tastiera; l'uso del mouse e dei toolbar, del resto, è un'alternativa intuitiva che non richiede molte spiegazioni.

Nota tecnica: alcune delle operazioni che può capitare di fare con *Mathematica* sono piuttosto impegnative per il PC; se usate una vecchia macchina (con processore "lento", tipo 486 o poca RAM, tipo 8Mb), potrebbe bloccarsi o metterci parecchio. Nel caso si abbia questo dubbio, meglio salvare *prima* di premere shift+invio.

### ■ 1.2. Calcoli numerici

Esempio. Se scriviamo:

$2+3*4^2$

e quindi premiamo shift+invio, succede questo:

```
In[1]:= 2 + 3 * 4 ^ 2
```

```
Out[1]= 50
```

Come si vede, Mathematica scrive In[n]= davanti all'input (n è il numero progressivo degli input digitati nella sessione), e fornisce un output.

(Quando un file *Mathematica* viene salvato, chiuso e riaperto, le scritte In[n]= e Out[n]= *non ricompaiono*, finché eventualmente l'input non viene valutato di nuovo; questo è il motivo per cui in questo file non tutti gli input e output sono numerati).

Regole di base: le 4 operazioni si indicano con

+ - \* /

Il prodotto si può indicare anche con uno spazio, ad esempio:

2 x

è lo stesso che

2 \* x

ma è diverso da 2x (senza spazio tra 2 e x), che denoterebbe una nuova variabile di nome "2x", e non 2 volte la variabile x. L'elevamento a potenza "a alla b" si scrive così:

a ^ b

Si possono usare tutte le parentesi che si vuole, *sempre tonde*, le une dentro le altre. Anzi, nel dubbio su quale sia la precedenza con cui Mathematica interpreta le operazioni, meglio abbondare, ad esempio:

```
In[2]:= (2 ^ (3 + x)) / (5 x ^ 2 + 1)
```

```
Out[2]=  $\frac{2^{3+x}}{1 + 5 x^2}$ 
```

Le parentesi quadre e graffe hanno un significato sintattico particolare che si vedrà in seguito, e non vanno usate quindi come parentesi "normali".

Esempio.

```
In[3]:= (2 / 45) ^ 36
```

```
Out[3]=  $\frac{68719476736}{327831360408733226531187361612285036244429647922515869140625}$ 
```

Se il risultato esatto di un calcolo è un numero razionale, Mathematica lo scrive come frazione esatta! Se vogliamo un valore numerico approssimato, dobbiamo scrivere:

**N[operazione che voglio fare]**

Ad esempio:

```
In[4]:= N[(2 / 45) ^ 36]
```

```
Out[4]= 2.09618 × 10-49
```

**Trucco.** Col simbolo % si può indicare, in un input, l'ultimo output precedente. Ad esempio:

```
In[5]:= % * 1000
```

```
Out[5]= 2.09618 × 10-46
```

Ad esempio, per ottenere un valore numerico approssimato dell'ultimo output, basta scrivere N[%]. Invece, per indicare l'output numero n si scrive %n. Ad esempio,

```
In[6]:= 1 / %3
```

```
Out[6]=  $\frac{327831360408733226531187361612285036244429647922515869140625}{68719476736}$ 
```

### ■ 1.3. Funzioni elementari:

In generale una funzione elementare si scrive F[x], cioè ha un nome che comincia sempre per maiuscola, e il suo argomento è scritto tra parentesi quadre. Funzioni comuni:

**Exp[x]**

(esponenziale di base e)

**Log[x]**

(logaritmo in base e)

**Log[a, x]**

(logaritmo in base a)

**Sin[x], Cos[x], Tan[x], ArcSin[x], ArcCos[x], ArcTan[x]**

(funzioni trigonometriche e loro inverse)

**Sinh[x], Cosh[x], Tanh[x]**

(funzioni iperboliche)

**Sqrt[x]**

(radice quadrata di x; ricordare che si può comunque scrivere x<sup>a</sup> per avere potenze a esponente reale qualunque; Mathematica, però, richiede sempre che la base sia positiva. Ad esempio, se si chiede a Mathematica di disegnare il grafico di x<sup>1/3</sup>, lo disegna solo per x>0. Per averlo tutto occorre usare un trucco che si spiegherà poi).

**Abs[x]**

(valore assoluto di x)

## ■ 1.4. Costanti notevoli:

pi greco	si scrive	Pi
il numero e di Nepero	si scrive	E
l'unità immaginaria	si scrive	I
più infinito	si scrive	Infinity
meno infinito	si scrive	-Infinity

## ■ 1.5. Definire una funzione o un numero

Se si vogliono fare più operazioni su una funzione non elementare, conviene darle un nome, in modo da poterla poi richiamare con quel nome. Ad esempio:

```
In[7]:= F[x_] = (Sin[2 x] + 2 Cos[3 x]) / (1 + x^2)
```

```
Out[7]= 
$$\frac{2 \cos[3 x] + \sin[2 x]}{1 + x^2}$$

```

Se ora scriviamo, ad esempio:

```
F[2 Pi / 3]
```

$$\frac{2 - \frac{\sqrt{3}}{2}}{1 + \frac{4\pi^2}{9}}$$

Mathematica calcola il valore di F in  $x=2\pi/3$ . La sintassi dunque è:

```
Nome[x_] = Qualcosa che contiene[x]
```

Il nome della funzione può essere una lettera (maiuscola) o una parola (con l'iniziale maiuscola). Alcune lettere sono proibite perché hanno già un altro significato. Ad esempio, *si possono usare*:

```
F[x_], G[x_], H[x_], M[x_]
```

ma *non si possono usare*:

```
N[x_], O[x_], D[x_].
```

(Motivo: N indica la valutazione numerica, O il simbolo di "o grande", D la derivazione). Se si prova a farlo, Mathematica ci avverte che non è lecito.

Naturalmente il nome della variabile indipendente è arbitrario. Ad esempio potremmo digitare:

```
In[8]:= F[t_] = (Sin[2 t] + 2 Cos[3 t]) / (1 + t^2)
```

```
Out[8]= 
$$\frac{2 \cos[3 t] + \sin[2 t]}{1 + t^2}$$

```

e sarebbe stata la stessa funzione dell'esempio precedente (In[7]) (t o x sono "variabili mute").

Analogamente, ma più semplicemente, si può *dare un nome a un numero* che in seguito comparirà in parecchie operazioni. Si pone ad esempio:

```
a = 2 Pi Sqrt[3]
```

$$2\sqrt{3}\pi$$

A questo punto possiamo ad esempio calcolare:

```
Exp[a]
```

$$e^{2\sqrt{3}\pi}$$

```
N[%]
```

```
53252.3
```

**Avvertenza.** Quando si definisce una funzione o un numero dandogli dei nomi, per tutta la sessione di lavoro Mathematica darà a quel nome quel significato. Dopo un po' questo potrebbe diventare fastidioso (non si sa più che lettere usare perché sono già tutte "compromesse"). Per dire a Mathematica "ora dimentica che x vuol dire la tal cosa" il comando è:

```
Clear[x]
```

Lo stesso per i nomi di funzione: con

```
Clear[F]
```

Mathematica fa sì che F non abbia più il significato di funzione che gli era stato dato.

## ■ 1.6. Matematica discreta

### 1.6.1. Successioni, tabulazioni

Definire una successione è esattamente analogo a definire una funzione:

```
A[k_] = 1 / (k ^ 2)
```

$$\frac{1}{k^2}$$

Fin qui, anzi, Mathematica non sa se noi intendiamo che k sia intero o reale. Potremmo chiedergli di calcolare:

```
A[Pi]
```

$$\frac{1}{\pi^2}$$

**Fattoriale e coefficiente binomiale** si indicano, naturalmente, con n! e con Binomial[n,k]

```
Binomial[10, 4]
```

```
210
```

```
10!
```

```
3628800
```

Se vogliamo tabulare i primi (o un po' di) valori della successione, il comando è:

```
Table[A[k], {k, n1, n2}]
```

che produce i valori di A[k] quando k è intero tra n1 e n2. Questo comando, quindi, forza Mathematica a interpretare k come intero.

```
Table[A[k], {k, 1, 10}]
```

```
{1, 1/4, 1/9, 1/16, 1/25, 1/36, 1/49, 1/64, 1/81, 1/100}
```

Al solito, se volessimo valori numerici in forma decimale, dovremmo chiedere:

```
Table[N[A[k]], {k, 1, 10}]
```

```
{1., 0.25, 0.111111, 0.0625, 0.04, 0.0277778, 0.0204082, 0.015625, 0.0123457, 0.01}
```

Ossia: "tabula i valori numerici approssimati di A[k] per k da 1 a 10"

Se vogliamo avere un'idea dell'andamento di A[k] per k grande, può essere utile tabulare i valori di A[k] fino a un valore grande, ma incrementando k ogni volta di un passo maggiore di 1 (per non visualizzare troppi numeri). Ad esempio:

```
Table[N[A[k]], {k, 1, 100, 10}]
```

```
{1., 0.00826446, 0.00226757, 0.00104058, 0.000594884,  
0.000384468, 0.000268745, 0.000198373, 0.000152416, 0.000120758}
```

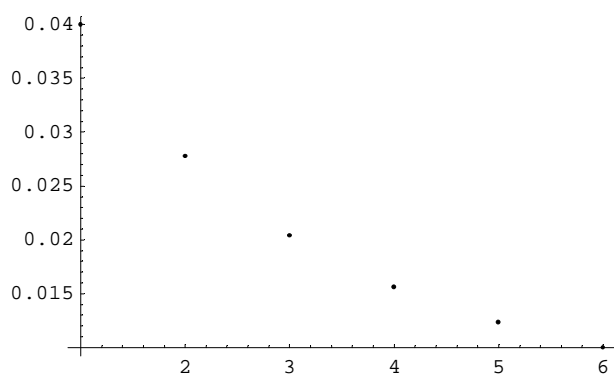
produce i valori di A[k] per k che si incrementa di 10 alla volta, da 1 a 100.

Per rappresentare graficamente i valori di (un po' di termini di) una successione, il comando è:

```
ListPlot[Table[A[k], {k, n1, n2}]]
```

che produce il grafico dei punti di coordinate (k,A[k]) per k da n1 a n2:

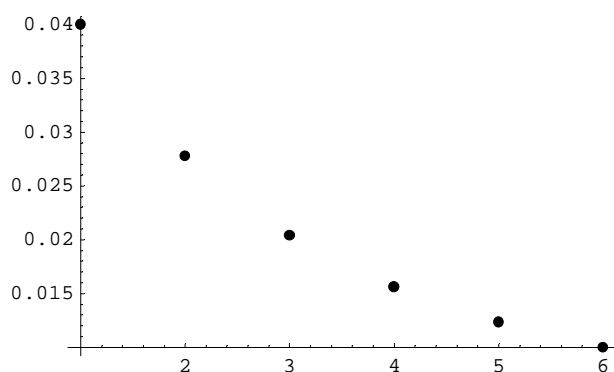
```
ListPlot[Table[A[k], {k, 5, 10}]]
```



- Graphics -

Se volete punti più grossi, fate così (e non chiedetemi perché):

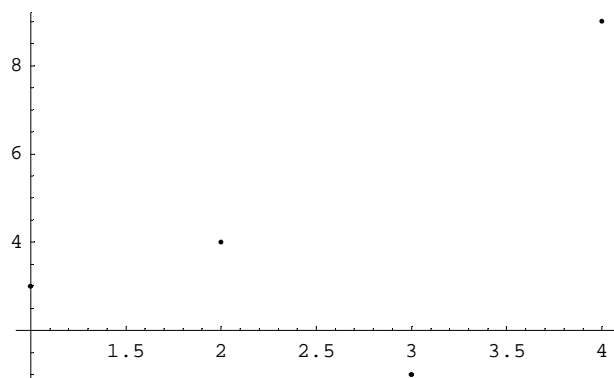
```
ListPlot[Table[A[k], {k, 5, 10}], Prolog -> AbsolutePointSize[5]]
```



- Graphics -

Più in generale, ListPlot[{lista di numeri}], produce un grafico dei punti di ascisse 1,2,3,... e ordinate questi numeri

```
ListPlot[{3, 4, 1, 9}]
```



- Graphics -

## 1.6.2. Sommatorie

L'operazione di sommatoria per k da n1 a n2 di A[k] si scrive naturalmente così:

```
Sum[A[k], {k, n1, n2}]
```

```
Sum[1 / k, {k, 1, 20}]
```

```
55835135  
15519504
```

```
N[%]
```

```
3.59774
```

L'ultimo numero è il valore numerico approssimato del precedente. Come il comando Table, così il comando Sum forza *Mathematica* a interpretare la variabile k (o come l'abbiamo chiamata) come intero.

Con ciò abbiamo introdotto la sintassi principale di *Mathematica*, che si utilizza in qualsiasi contesto. Ora passiamo agli argomenti specifici di calcolo infinitesimale.



## 2. Argomenti specifici di Analisi Matematica

### ■ 2.1. Serie numeriche

Si può definire una serie con lo stesso operatore Sum di sommatoria, facendo variare l'indice fino a infinito; se ne è capace, Mathematica fornisce la somma corretta della serie...

```
Sum[1 / (k^2), {k, 1, Infinity}]
```

$$\frac{\pi^2}{6}$$

```
Sum[1 / k, {k, 1, Infinity}]
```

```
Sum::div : Sum does not converge.
```

$$\sum_{k=1}^{\infty} \frac{1}{k}$$

...oppure dice che la serie non converge!

```
Sum[1 / (k^2 + 2^k), {k, 1, Infinity}]
```

$$\sum_{k=1}^{\infty} \frac{1}{k^2 + 2^k}$$

...oppure ricopia l'espressione perché non ne sa calcolare la somma, ma non ha neanche la prova che la serie diverga.

Per indagare numericamente il comportamento di una serie può essere utile definire la successione delle somme parziali, e tabularne e/o rappresentarne graficamente i valori. Consideriamo ad esempio la serie il cui termine generale è:

```
In[9]:= B[k_] = (2^k) / (k + k!)
```

$$\text{Out}[9]= \frac{2^k}{k + k!}$$

```
In[10]:= S[n_] = Sum[B[k], {k, 1, n}]
```

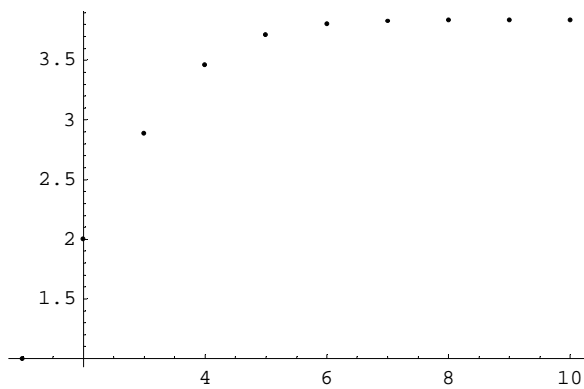
$$\text{Out}[10]= \sum_{k=1}^n B[k]$$

```
In[11]:= Table[N[S[k]], {k, 1, 10}]
```

```
Out[11]= {1., 2., 2.88889, 3.46032, 3.71632, 3.80447, 3.82983, 3.83618, 3.83759, 3.83787}
```

...dopo i primi 10 termini appaiono già stabilizzate le prime 3 cifre decimali della somma...

```
In[12]:= ListPlot[%11]
```



```
Out[12]= - Graphics -
```

## ■ 2.2. Numeri complessi

I numeri complessi si scrivono  $a+I b$ . Si possono eseguire le operazioni algebriche:

$$(3 + 2 I) (2 - I)^3$$

$$28 - 29 i$$

Inoltre, si possono calcolare parte reale, parte immaginaria, coniugato, modulo, argomento di un numero. I comandi corrispondenti sono, rispettivamente:

$$\text{Re}[z], \text{Im}[z], \text{Conjugate}[z], \text{Abs}[z], \text{Arg}[z]$$

Esempi:

$$\text{Re}[(1 + 3 I)^5]$$

$$316$$

$$\text{Arg}[1 + I]$$

$$\frac{\pi}{4}$$

$$\text{Arg}[1 - 2 I]$$

$$-\text{ArcTan}[2]$$

$$\text{Abs}[1 + I]$$

$$\sqrt{2}$$

$$\text{Conjugate}[(3 + 2 I)^2]$$

$$5 - 12 i$$

Esempio di calcolo della somma di una serie nel campo complesso (v. § "Serie numeriche"):

$$\text{Sum}[1 / (1 + I)^k, \{k, 0, \text{Infinity}\}]$$

$$1 - i$$

### 2.2.1. Radici ennesime in C.

Se si calcola la radice quadrata o ennesima di un numero complesso direttamente, Mathematica ne calcola una sola (anziché  $n$ ). Questo perché il simbolo `Sqrt[ ]` oppure `( )^n` denota per *Mathematica* una funzione (univoca). Esempi:

```
Sqrt[2 I]
```

```
1 + i
```

```
I ^ (1 / 3)
```

```
(-1) ^ 1/6
```

Il risultato non è soddisfacente. Per trovare davvero le radici ennesime di  $z$ , bisogna chiedere esplicitamente a *Mathematica* di applicare la formula di De Moivre, cioè dare una sequenza di istruzioni del tipo:

```
x = 1 + I
```

```
1 + i
```

```
n = 6
```

```
6
```

```
r = Abs[x]
```

```
 $\sqrt{2}$ 
```

```
t = Arg[x]
```

```
 $\frac{\pi}{4}$ 
```

```
z[k_] = (r ^ (1 / n)) (Cos[(t + 2 k Pi) / n] + I Sin[(t + 2 k Pi) / n])
```

```
 $2^{1/12} \left( \cos\left[\frac{1}{6} \left(\frac{\pi}{4} + 2 k \pi\right)\right] + i \sin\left[\frac{1}{6} \left(\frac{\pi}{4} + 2 k \pi\right)\right] \right)$ 
```

```
Table[N[z[k]], {k, 0, n - 1}]
```

```
{1.0504 + 0.138288 i, 0.405439 + 0.978816 i, -0.64496 + 0.840529 i,  
-1.0504 - 0.138288 i, -0.405439 - 0.978816 i, 0.64496 - 0.840529 i}
```

Plottare gli  $n$  punti nel campo complesso richiede invece un'istruzione un po'... complessa, si rimanda al Notebook specifico su questo.

## ■ 2.3. Risoluzione di equazioni, esatta o approssimata

La risoluzione di un'equazione può essere esatta o approssimata.

1. Risolvere un'equazione esattamente significa, con passaggi algebrici opportuni, trovare il valore esatto di tutte le soluzioni (come quando risolviamo un'equazione di secondo grado applicando la formula relativa).

2. Risolvere un'equazione con approssimazione significa determinare, coi metodi dell'Analisi Numerica (ossia con qualche tipo di algoritmo iterativo) una soluzione approssimata (o tutte le soluzioni, approssimate) dell'equazione stessa.

Siamo noi che dobbiamo dire a *Mathematica* quale delle 2 cose fare: se gli chiediamo di risolvere esattamente un'equazione quando ciò non è possibile, non farà nulla (in particolare, *non ci darà* soluzioni approssimate). Se chiediamo soluzioni approssimate, queste saranno "solo" approssimate, e non esatte, anche qualora fosse possibile trovarle esatte.

**Il comando per trovare le soluzioni esatte è `Solve`; il comando per trovare le soluzioni approssimate è `FindRoot`.**

**Sintassi di `Solve`:**

```
Solve[A[x] == B[x], x]
```

Esempio:

```
Solve[x^2 + 3 x + 1 == 0, x]
```

$$\left\{ \left\{ x \rightarrow \frac{1}{2} (-3 - \sqrt{5}) \right\}, \left\{ x \rightarrow \frac{1}{2} (-3 + \sqrt{5}) \right\} \right\}$$

Con questo comando *Mathematica* trova le radici *reali o complesse* dell'equazione. Esempi:

```
Solve[x^3 + x^2 + 1 == 0, x]
```

$$\left\{ \left\{ x \rightarrow -\frac{1}{3} - \frac{1}{3} \left( \frac{2}{29 - 3\sqrt{93}} \right)^{1/3} - \frac{1}{3} \left( \frac{1}{2} (29 - 3\sqrt{93}) \right)^{1/3} \right\}, \right.$$

$$\left. \left\{ x \rightarrow -\frac{1}{3} + \frac{1}{6} (1 + i\sqrt{3}) \left( \frac{1}{2} (29 - 3\sqrt{93}) \right)^{1/3} + \frac{1 - i\sqrt{3}}{3 \cdot 2^{2/3} (29 - 3\sqrt{93})^{1/3}} \right\}, \right.$$

$$\left. \left\{ x \rightarrow -\frac{1}{3} + \frac{1}{6} (1 - i\sqrt{3}) \left( \frac{1}{2} (29 - 3\sqrt{93}) \right)^{1/3} + \frac{1 + i\sqrt{3}}{3 \cdot 2^{2/3} (29 - 3\sqrt{93})^{1/3}} \right\} \right\}$$

L'equazione di terzo grado precedente ha una radice reale e due radici complesse coniugate.

```
Solve[Exp[2 x] - 3 Exp[x] + 1 == 0, x]
```

Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found.

$$\left\{ \left\{ x \rightarrow \text{Log} \left[ \frac{1}{2} (3 - \sqrt{5}) \right] \right\}, \left\{ x \rightarrow \text{Log} \left[ \frac{1}{2} (3 + \sqrt{5}) \right] \right\} \right\}$$

L'equazione precedente è un'equazione di secondo grado in  $\text{Exp}[x]$ , perciò può essere risolta esattamente.

Si possono risolvere anche **sistemi, lineari e** (talvolta!) **non lineari**:

```
Solve[{x + 2 y == 1, 3 x - y == 0}, {x, y}]
```

$$\left\{ \left\{ x \rightarrow \frac{1}{7}, y \rightarrow \frac{3}{7} \right\} \right\}$$

```
Solve[{x (y + 1) == 0, 3 x - y == 0}, {x, y}]
```

$$\left\{ \left\{ x \rightarrow -\frac{1}{3}, y \rightarrow -1 \right\}, \left\{ x \rightarrow 0, y \rightarrow 0 \right\} \right\}$$

**Sintassi di `FindRoot`:**

```
FindRoot[A[x] == 0, {x, x0}]
```

Questo comando trova una soluzione approssimata dell'equazione in un intorno del punto  $x_0$  (più precisamente,  $x_0$  viene assunto come primo valore dell'algoritmo iterativo che definisce la soluzione approssimata)

Ad esempio, l'equazione  $\text{Tan}[x] = x$  ha infinite soluzioni (una è zero). Possiamo trovarne qualcuna dando valori diversi a  $x_0$ :

```
FindRoot[Tan[x] == x, {x, 2}]
```

```
{x -> 4.49341}
```

```
FindRoot[Tan[x] == x, {x, 3}]
```

```
{x -> 7.72525}
```

```
FindRoot[Tan[x] == x, {x, 8}]
```

```
{x -> 10.9041}
```

Per cercare le soluzioni con criterio, anziché andando a casaccio, può essere utile prima tracciare con Mathematica il grafico di  $F[x]=\text{Tan}[x]-x$ , e vedere approssimativamente dove cadono i suoi zeri. Vedremo nel prossimo paragrafo come si fa.

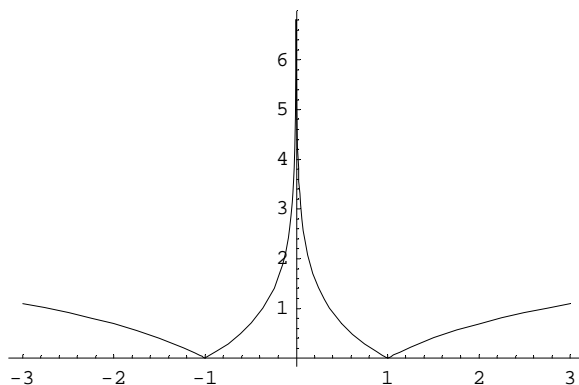
## ■ 2.4. Grafici di funzioni

La sintassi di base per tracciare il grafico di una funzione di una variabile è:

```
Plot[F[x], {x, a, b}]
```

Così facendo, Mathematica traccia il grafico di  $F[x]$  per  $x$  nell'intervallo  $[a,b]$ . Se in quell'intervallo la funzione non è sempre definita, Mathematica protesta, ma fa il possibile.

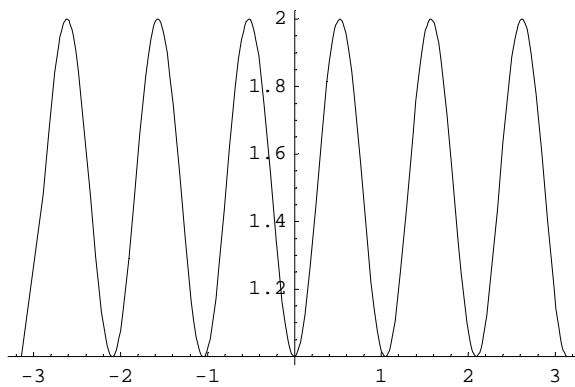
```
Plot[Abs[Log[Abs[x]]], {x, -3, 3}]
```



- Graphics -

Notare che noi abbiamo detto ben poco a Mathematica su come tracciare il grafico, e lui ha fatto una serie di scelte (quale scala mettere su ciascun asse, ad esempio). Talvolta non saremo d'accordo con le sue scelte, e vorremo chiedere qualcosa di più preciso. Questo si fa con vari tipi di *opzione*. Esempi:

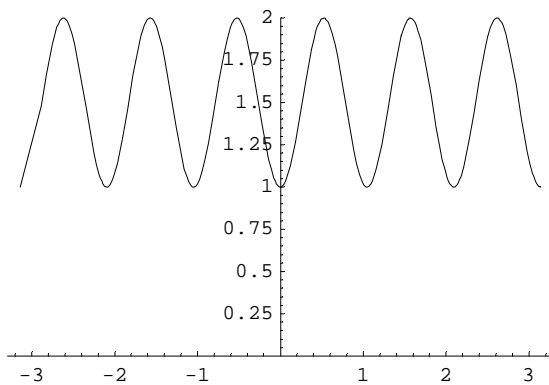
```
Plot[1 + Sin[3 x]^2, {x, - Pi, Pi}]
```



- Graphics -

Mathematica ha tracciato il grafico scegliendo come asse orizzontale la retta  $y=1$  (guardare i numeri scritti sull'asse!). Questo può essere fuorviante. Se vogliamo che il grafico mostri le  $y$  tra 0 e 2, ad esempio (in particolare, che l'asse orizzontale sia la retta  $y=0$ ), possiamo usare l'opzione **PlotRange**->**{a,b}** dove a,b sono i valori estremi delle  $y$  che vogliamo compaiano nel grafico:

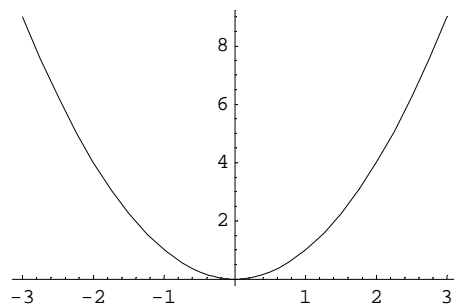
```
Plot[1 + Sin[3 x]^2, {x, - Pi, Pi}, PlotRange -> {0, 2}]
```



- Graphics -

Altro problema:

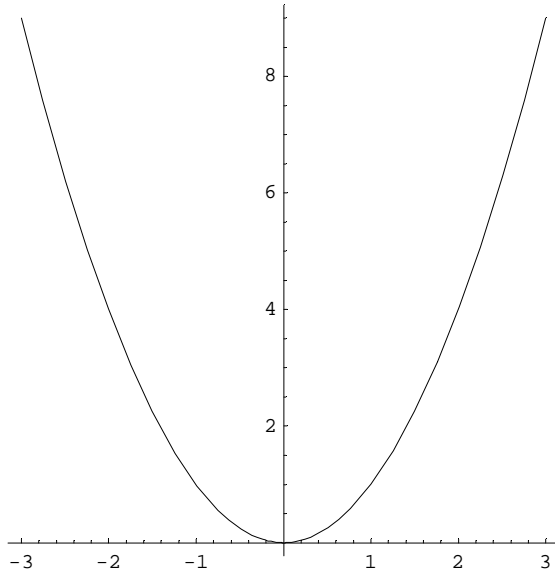
```
Plot[x^2, {x, -3, 3}]
```



- Graphics -

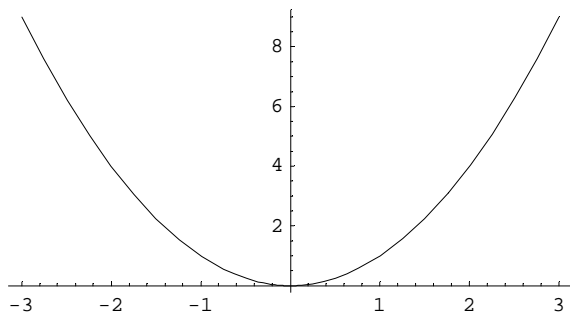
Nel grafico precedente la scala non è la stessa sugli assi. L'opzione **AspectRatio** **a** (a numero positivo, maggiore o minore di 1) permette di dilatare un asse rispetto all'altro. Per tentativi, si può trovare la scala che si vuole:

```
Plot[x^2, {x, -3, 3}, AspectRatio -> 1]
```



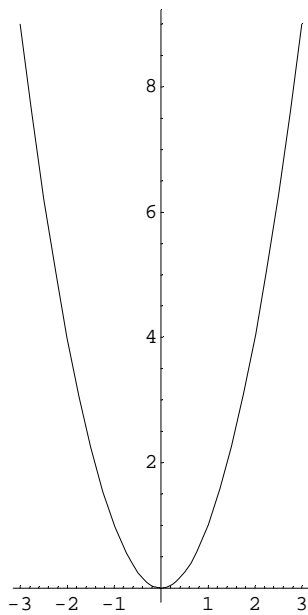
- Graphics -

```
Plot[x^2, {x, -3, 3}, AspectRatio -> 0.5]
```



- Graphics -

```
Plot[x^2, {x, -3, 3}, AspectRatio -> 2]
```

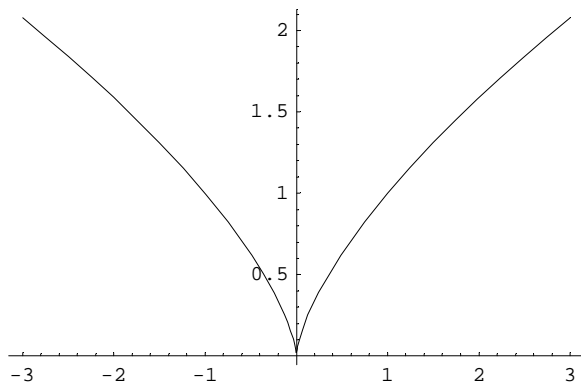


- Graphics -

Vi sono molte altre opzioni possibili per il comando Plot, ma non ve le spiego per non riscrivere qui il manuale di *Mathematica*...

Si osservi il prossimo trucco per tracciare il grafico di  $x^{2/3}$  anche per  $x < 0$ :

```
Plot[(x^2)^(1/3), {x, -3, 3}]
```



- Graphics -

## ■ 2.5. Funzioni definite a tratti

Si possono ottenere con un comando del tipo "se allora". Ad esempio la funzione che vale  $x$  se  $x < 0$  e  $x^2$  se  $x > 0$  si può ottenere così:

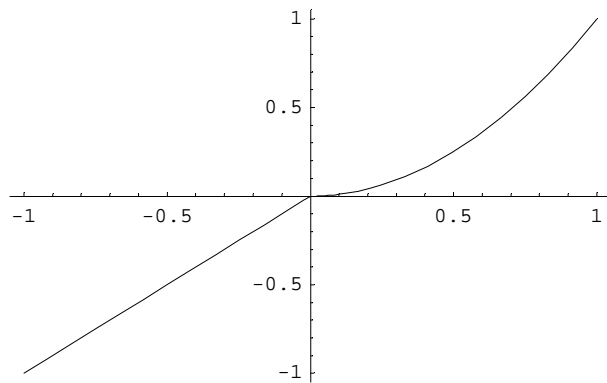
```
In[13]:= If[x < 0, x, x^2]
```

```
Out[13]= If[x < 0, x, x^2]
```

La sintassi è: If[Condizione, Allora così, Altrimenti così]



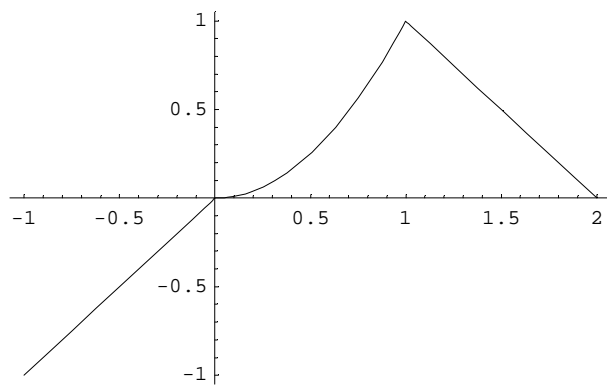
```
In[14]:= Plot[If[x < 0, x, x^2], {x, -1, 1}]
```



```
Out[14]= - Graphics -
```

Si possono annidare tanti If[...] uno dentro l'altro, e ottenere funzioni definite in 3 o più pezzi diversi, ad esempio:

```
In[15]:= Plot[If[x < 0, x, If[x < 1, x^2, 2 - x]], {x, -1, 2}]
```

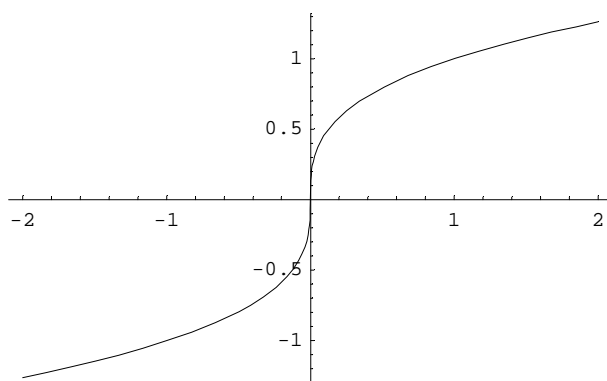


```
Out[15]= - Graphics -
```

L'ultimo input si può leggere così: "Disegna, per  $x$  in  $[-1,2]$ , la funzione che se  $x < 0$  vale  $x$ , altrimenti (se  $x < 1$  vale  $x^2$ , altrimenti vale  $2-x$ )".

Ad esempio, per ottenere il grafico della radice cubica di  $x$  anche per  $x < 0$  si può fare così:

```
In[16]:= Plot[If[x > 0, x^(1/3), -(-x)^(1/3)], {x, -2, 2}]
```



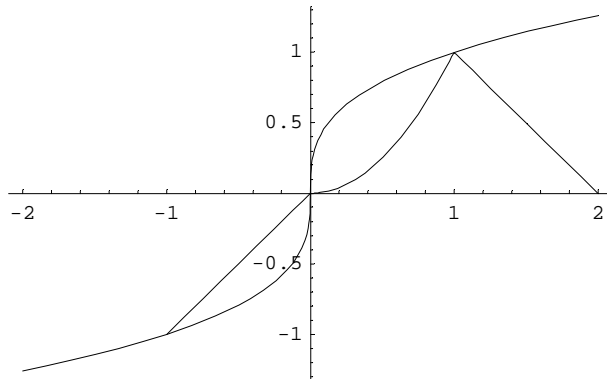
```
Out[16]= - Graphics -
```

Si possono tracciare i grafici di più funzioni su un solo grafico. Questo si può fare in due modi:

```
Plot[{F[x], G[x], H[x]}, {x, a, b}]
```

disegna sullo stesso grafico le funzioni F, G, H. Se invece abbiamo già prodotto separatamente i grafici di F, G e H, e ci piacerebbe ora vederli tutti insieme, usiamo il comando `Show[%n,%m,%r]`, dove n, m, r sono i numeri progressivi degli input in cui abbiamo fatto i grafici di F, G, H. Ad esempio:

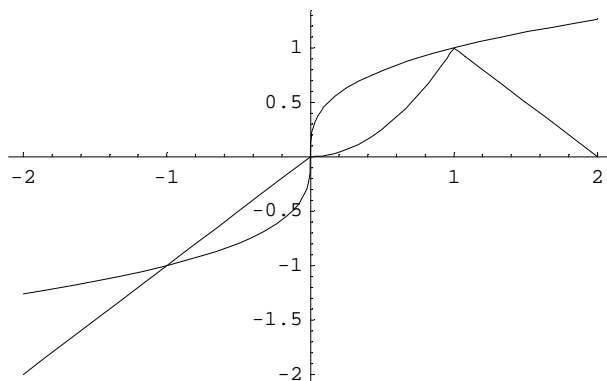
```
In[17]:= Show[%15, %16]
```



```
Out[17]= - Graphics -
```

Si noti che, col comando `Show`, *Mathematica* non esegue nuovi calcoli, ma si limita a mostrare sullo stesso sistema d'assi due o più grafici già ottenuti; se le due funzioni erano state disegnate su intervalli diversi, col comando `show` vedremo solo i tratti di grafico che erano già stati disegnati (nell'ultimo esempio, si noti che la retta  $x$  "non esiste" per  $x < -1$ , perché così era nel grafico originale). Per ovviare a questo problema, dovrei usare il primo metodo, col quale *Mathematica* ricalcola i due grafici:

```
In[18]:= Plot[{If[x < 0, x, If[x < 1, x^2, 2 - x]],  
If[x > 0, x^(1/3), -(-x)^(1/3)]}, {x, -2, 2}]
```



```
Out[18]= - Graphics -
```

## ■ 2.6. Uso dei comandi `Sum` e `Table` con funzioni

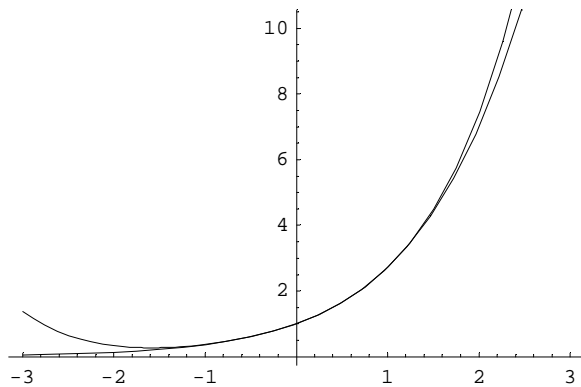
Col comando `Sum` si possono definire anche funzioni che sono somme parziali di serie di funzioni, o serie di funzioni:

```
F[x_, n_] = Sum[(x^k) / k!, {k, 0, n}]
```

$$\frac{e^x (1 + n) \text{Gamma}[1 + n, x]}{\text{Gamma}[2 + n]}$$

Quella ottenuta così è una funzione di 2 variabili:  $x$  e  $n$ . Fissando un valore di  $n$ , è una funzione di  $x$ :

```
Plot[{Exp[x], F[x, 4]}, {x, -3, 3}]
```

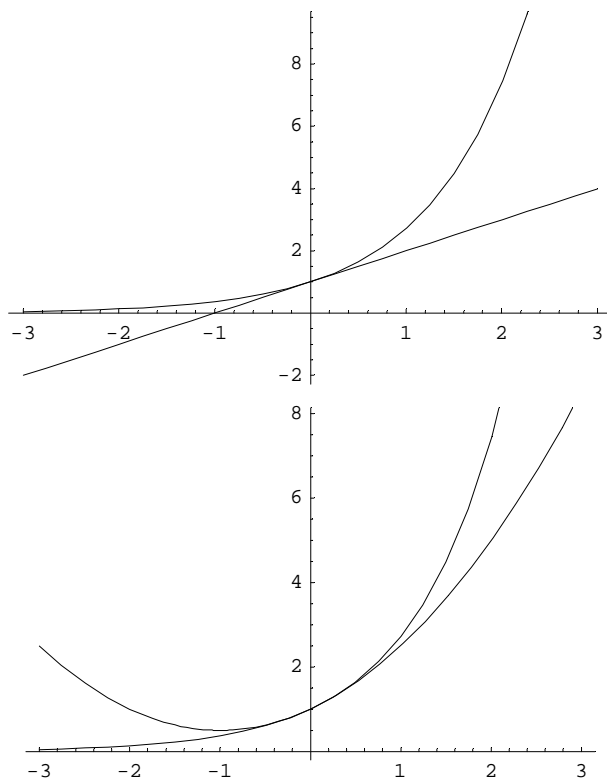


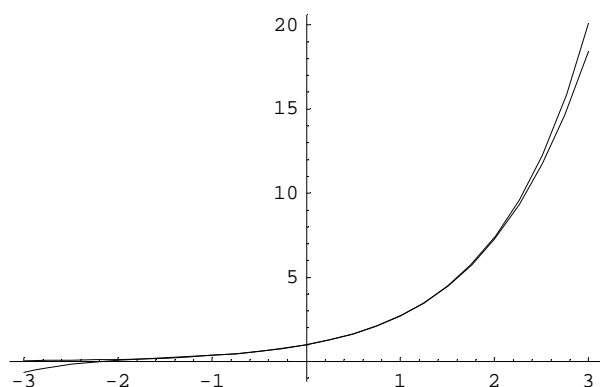
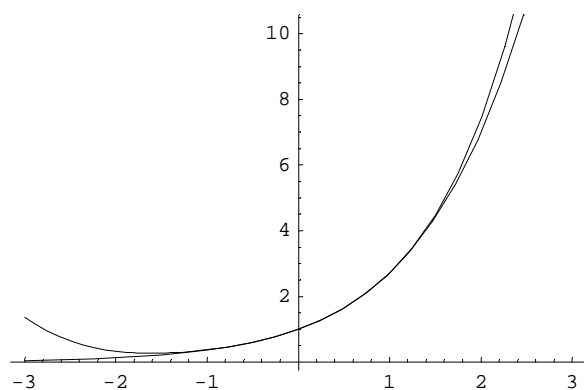
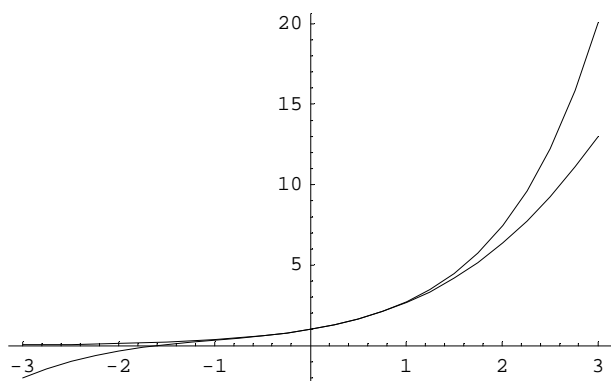
- Graphics -

Il grafico precedente mostra  $\text{Exp}[x]$  insieme al suo "polinomio di Taylor di grado 4": vicino a  $x=0$  c'è buona approssimazione tra i due.

Si possono produrre con una sola istruzione gruppi di grafici col comando Table. Ad esempio, vogliamo confrontare  $\text{Exp}[x]$  e il suo polinomio di Taylor  $F[x,n]$  sull'intervallo  $[-3,3]$  per valori crescenti di  $n$ :

```
Table[Plot[{Exp[x], F[x, n]}, {x, -3, 3}], {n, 1, 5}]
```





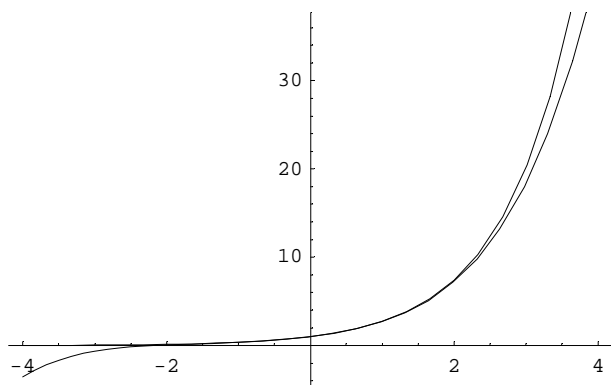
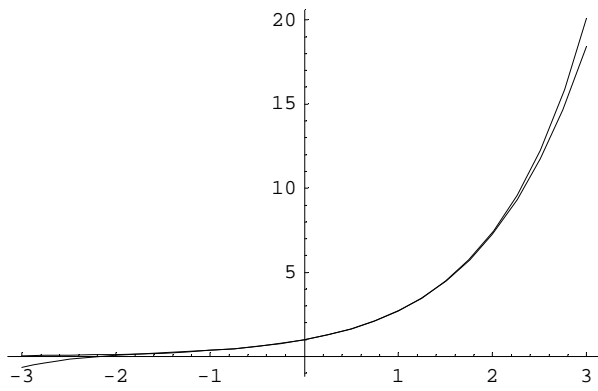
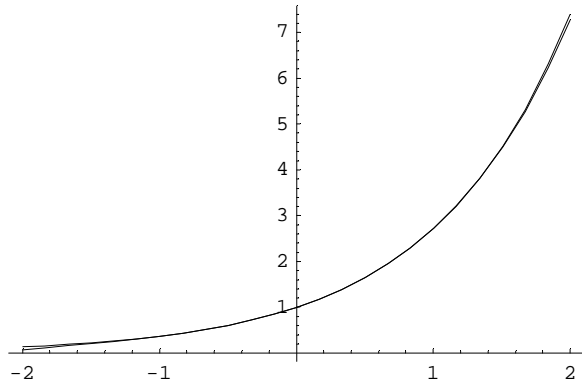
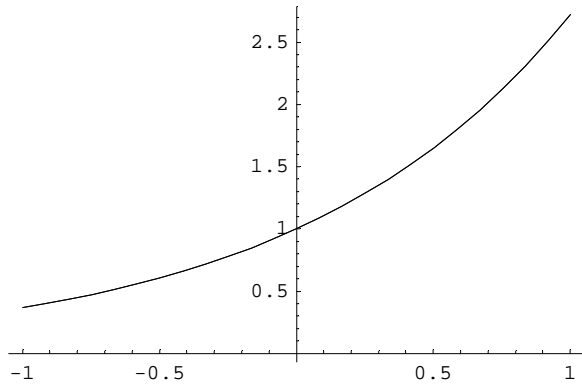
```
{ - Graphics - , - Graphics - , - Graphics - , - Graphics - , - Graphics - }
```

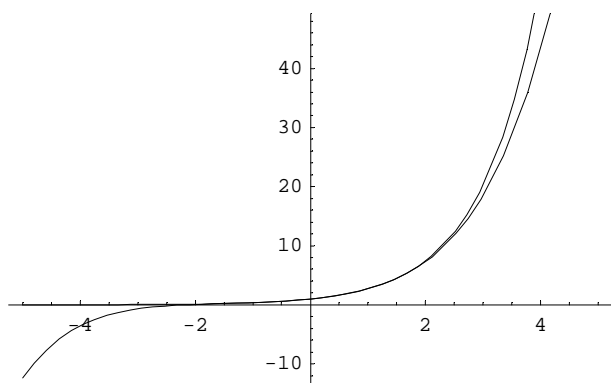
(Un solo input ha prodotto 5 grafici).

Al crescere di  $n$  l'approssimazione è sempre migliore, su un intervallo fissato.

Oppure, vogliamo confrontare  $\text{Exp}[x]$  e il suo polinomio di Taylor  $F[x,5]$  sull'intervallo  $[-n,n]$  per valori crescenti di  $n$ :

```
Table[Plot[{Exp[x], F[x, 5]}, {x, -n, n}], {n, 1, 5}]
```





```
{ - Graphics - , - Graphics - , - Graphics - , - Graphics - , - Graphics - }
```

Aumentando l'ampiezza dell'intervallo, l'approssimazione è sempre peggiore.

## ■ 2.7. Calcolo infinitesimale per funzioni di una variabile

### 2.7.1. Calcolo dei limiti

I prossimi esempi mostrano la sintassi:

```
Limit[Sin[x] / x, x → 0]
```

```
1
```

```
Limit[(x^2 + 2^x) / (1 + Log[x]^3), x → Infinity]
```

```
∞
```

```
Limit[(x^2 + 2^x) / (1 + x^3), x → -Infinity]
```

```
0
```

(La freccia si ottiene digitando - seguito da >). Se il limite destro e sinistro sono diversi, *Mathematica per default calcola il limite destro* (questa è una caratteristica non molto bella del programma! Se si trova un risultato non si sa se il limite c'è o c'è solo da destra):

```
Limit[Exp[1 / x], x → 0]
```

```
∞
```

Per chiedere esplicitamente a Mathematica di calcolare un limite destro o sinistro usiamo l'opzione `Direction` `1` per il limite sinistro, `Direction` `-1` per il limite destro:

```
Limit[x / Abs[x], x → 0, Direction → 1]
```

```
-1
```

```
Limit[x / Abs[x], x → 0, Direction → -1]
```

```
1
```

```
Limit[Exp[1 / x], x → 0, Direction → 1]
```

```
0
```

Se si prova a calcolare un limite che non esiste, cosa succede?

```
Limit[Sin[1 / x], x → 0]
```

```
Interval[{-1, 1}]
```

L'output significa che il limite non esiste e la "classe limite" è l'intervallo [-1,1].

```
Limit[x Sin[x], x → Infinity]
```

```
Interval[{-∞, ∞}]
```

L'output significa che il limite non esiste e la "classe limite" è l'intervallo (-Infinito,Infinito).

### 2.7.2. Calcolo delle derivate

La sintassi per il calcolo della derivata prima di una funzione è:

```
D[F[x], x]
```

Esempio:

```
D[Sin[2 x], x]
```

```
2 Cos[2 x]
```

A questo punto per calcolare la derivata seconda, cioè la derivata prima di quanto appena trovato, si può fare così:

```
D[%, x]
```

```
-4 Sin[2 x]
```

Oppure:

```
D[D[Sin[2 x], x], x]
```

```
-4 Sin[2 x]
```

### 2.7.3. Formula di Taylor

Per scrivere la formula di Taylor di F[x] all'ordine n, centrata in x0, il comando è:

```
Series[F[x], {x, x0, n}]
```

Notare che, non ostante il comando si chiami Series, serve a scrivere la *formula* di Taylor, non la *serie* di Taylor.

Inoltre, notare che il resto (secondo Peano) è scritto in modo insolito: al posto del simbolo di o piccolo viene usato il simbolo di O grande. In pratica, ciò significa che se Mathematica scrive  $O[x]^4$  bisogna interpretare  $o[x]^3$ , e così via.

```
Series[Sin[x], {x, 0, 9}]
```

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} + O[x]^{10}$$

Qui  $O[x]^{10} = o[x]^9$

```
Series[Exp[x^2] - x Sin[x], {x, 0, 5}]
```

$$1 + \frac{2x^4}{3} + O[x]^6$$

Se si vuole solo il *polinomio* di Taylor, anziché la *formula* di Taylor, il comando è

```
Normal[Series[F[x], {x, x0, n}]]
```

```
Normal[Series[Exp[x], {x, 1, 4}]]
```

$$e + e(-1+x) + \frac{1}{2}e(-1+x)^2 + \frac{1}{6}e(-1+x)^3 + \frac{1}{24}e(-1+x)^4$$

## 2.7.4. Calcolo delle primitive

La sintassi per il calcolo di una primitiva di una funzione è:

```
Integrate[F[x], x]
```

```
Integrate[(1+x)/(1+x^2+x), x]
```

$$\frac{\text{ArcTan}\left[\frac{1+2x}{\sqrt{3}}\right]}{\sqrt{3}} + \frac{1}{2} \text{Log}[1+x+x^2]$$

(Notare che *Mathematica* non aggiunge la costante arbitraria). Naturalmente se la primitiva non è una funzione elementare, *Mathematica* non trova nulla:

```
Integrate[1/ArcTan[x], x]
```

$$\int \frac{1}{\text{ArcTan}[x]} dx$$

che è come dire che *Mathematica* non ha fatto nulla.

Talvolta la primitiva di una funzione è una funzione non elementare che però in Analisi ha un nome ed è studiata; se *Mathematica* la conosce, fornisce questo risultato. Ad esempio:

```
Integrate[1/Log[x], x]
```

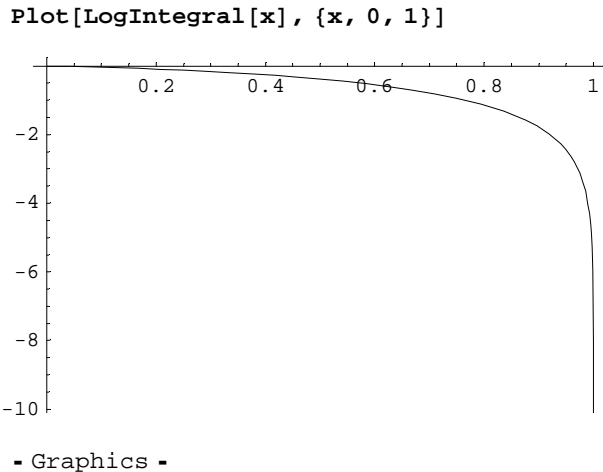
```
LogIntegral[x]
```

La funzione  $\text{LogIntegral}[x]$  è definita proprio come primitiva di  $\frac{1}{\text{Log}[x]}$ . *Mathematica* conosce questa funzione ed è in grado di calcolarne valori. Ad esempio:

```
N[LogIntegral[2]]
```

```
1.04516
```





### 2.7.5. Calcolo degli integrali definiti, esatto ed approssimato

La sintassi per calcolare l'integrale definito di F sull'intervallo [a,b] è:

`Integrate[F[x], {x, a, b}]`

`Integrate[1 / (1 + x^2), {x, 0, 1}]`

$$\frac{\pi}{4}$$

`Integrate[(1 + x) / (1 + x^2 + x), {x, 0, 1}]`

$$\frac{1}{6} \left( 3 \operatorname{Log}[3] + i \sqrt{3} \operatorname{Log}[3 (1 - i \sqrt{3})] - i \sqrt{3} \operatorname{Log}[3 - i \sqrt{3}] - i \sqrt{3} \operatorname{Log}[3 (1 + i \sqrt{3})] + i \sqrt{3} \operatorname{Log}[3 + i \sqrt{3}] \right)$$

Talvolta, come nell'ultimo caso, l'output è infelice: qui ad esempio è espresso attraverso numeri complessi (anche se in realtà è reale). Comunque possiamo sempre chiedergli quanto fa:

`N[%]`

0.851606 + 0. i

cioè è un numero reale, circa 0.851...

Il comando `Integrate` cerca di calcolare un integrale definito in modo *esatto*, passando attraverso la primitiva (o con altri metodi esatti): se si prova a calcolare l'integrale definito di una funzione che non ha primitiva conosciuta, *Matematica* non calcola nulla:

`Integrate[1 / (1 + x + Exp[x]), {x, 0, 1}]`

$$\int_0^1 \frac{1}{1 + e^x + x} dx$$

C'è una strada diversa: calcolare un valore approssimato dell'integrale definito, coi metodi dell'analisi numerica. Questo si fa con un comando diverso:

`NIntegrate[F[x], {x, a, b}]`

```
NIntegrate[1 / (1 + x + Exp[x]), {x, 0, 1}]
```

```
0.329971
```

Si confronti il risultato trovato con `Integrate` e `NIntegrate` in un caso in cui entrambi i metodi sono applicabili:

```
Integrate[1 / (1 + x^2), {x, 0, 1}]
```

```
 $\frac{\pi}{4}$ 
```

```
NIntegrate[1 / (1 + x^2), {x, 0, 1}]
```

```
0.785398
```

Si osservi che:

```
N[Pi / 4]
```

```
0.785398
```

cioè il valore "approssimato" è approssimato davvero bene.

## 2.7.6. Integrali generalizzati

Si possono calcolare integrali definiti generalizzati:

```
Integrate[1 / Sqrt[x], {x, 0, 1}]
```

```
2
```

L'integrale precedente converge ed è calcolabile esattamente.

```
Integrate[1 / x, {x, 0, 1}]
```

```
Integrate::idiv : Integral of  $\frac{1}{x}$  does not converge on {0, 1}.
```

$$\int_0^1 \frac{1}{x} dx$$

L'integrale precedente diverge.

```
Integrate[Exp[-x^2], {x, -Infinity, Infinity}]
```

```
 $\sqrt{\pi}$ 
```

L'integrale precedente converge ed è calcolabile esattamente.

```
Integrate[1 / Sqrt[Sin[x]], {x, 0, 1}]
```

```
2 Hypergeometric2F1[ $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{5}{4}$ , Sin[1]^2] Sqrt[Sin[1]]
```

Questa è una tipica risposta "da Mathematica": ha calcolato il valore esatto dell'integrale, esprimendolo con complicate funzioni. Quanto vale?

**N[%]**

2.03481

**Integrate[1 / Sqrt[x + Sin[x]], {x, 0, 1}]**

$$\int_0^1 \frac{1}{\sqrt{x + \sin[x]}} dx$$

Questa volta non ce l'ha fatta. Se però gli chiediamo di calcolare l'integrale numericamente:

**NIntegrate[1 / Sqrt[x + Sin[x]], {x, 0, 1}]**

1.42608

si capisce che l'integrale converge. Se si provasse a calcolare numericamente un integrale divergente:

**NIntegrate[1 / (x + Sin[x]), {x, 0, 1}]**

NIntegrate::slwcon :

Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration being 0, oscillatory integrand, or insufficient WorkingPrecision. If your integrand is oscillatory try using the option Method->Oscillatory in NIntegrate.

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 7 recursive bisections in x near x = 4.369993747903707`\*\*^-57.

11976.6

Il messaggio di Mathematica mette in guardia sul fatto che la convergenza dell'algoritmo di approssimazione è lenta, il che deve far sospettare che l'integrale possa divergere. L'output è comunque un valore numerico (in questo caso inattendibile).

## 2.7.7. Funzioni integrali

Una funzione integrale di F[x] è una funzione del tipo:

**G[x\_] = Integrate[F[t], {t, 0, x}]**

(Attenzione all'uso corretto della variabile muta t, di integrazione).

Proviamo a definire, ad esempio:

**G[x\_] = Integrate[1 / (1 + t^2), {t, 0, x}]**

ArcTan[x]

La primitiva qui è calcolabile esattamente e la funzione integrale è calcolata esplicitamente.

**H[x\_] = Integrate[1 / (Exp[t] + t^2), {t, 0, x}]**

$$\int_0^x \frac{1}{e^t + t^2} dt$$

Ovviamente, non riesce a fare nulla in questo caso. Se proviamo a valutare:

**H[10]**

$$\int_0^{10} \frac{1}{e^t + t^2} dt$$

non troviamo niente.

In questi casi (quando cioè la primitiva non è calcolabile elementarmente, e non è nemmeno una funzione speciale, conosciuta da Mathematica), conviene definire la funzione integrale come integrale calcolato *numericamente*, cioè porre:

```
H[x_] = NIntegrate[1 / (Exp[t] + t^2), {t, 0, x}]
```

```
NIntegrate::nlim : t = x is not a valid limit of integration.
```

```
NIntegrate::nlim : t = x is not a valid limit of integration.
```

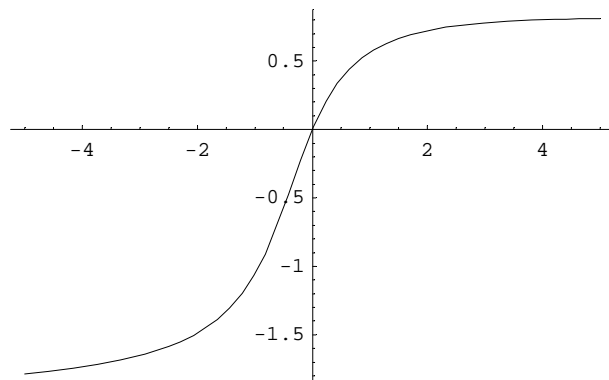
```
NIntegrate[ $\frac{1}{e^t + t^2}$ , {t, 0, x}]
```

Mathematica ha protestato, perché gli abbiamo chiesto di calcolare *numericamente* l'integrale definito tra due estremi di cui il secondo *non è un numero*, ma è x! Ovviamente non può farlo; però ora Mathematica sa chi è H[x] e può calcolarla in un punto, o anche tracciarne il grafico:

**H[3]**

0.780494

```
Plot[H[x], {x, -5, 5}]
```



- Graphics -

Nota: il grafico di una funzione integrale è un esempio di operazione che può risultare "impegnativa" per un vecchio PC, come accennato all'inizio. Nel dubbio, meglio salvare prima di premere shift+invio.

### 2.7.8. Equazioni differenziali

Mathematica può risolvere le equazioni differenziali (ed anche i problemi di Cauchy) che noi sapremmo risolvere a mano, con carta e penna. Ma ci mette molto meno.

La sintassi con cui si scrive un'equazione differenziale è illustrata dall'esempio seguente:

```
y''[x] + 3 y'[x] - 2 y[x] == Sin[x]
```

Notare che la funzione incognita va sempre scritta nella forma  $y[x]$  (cioè indicando la variabile indipendente, che potrebbe anche non chiamarsi  $x$ ), le derivate si indicano con  $y'[x]$ ,  $y''[x]$ ,  $y'''[x]$ ,..., e tra i due membri dell'equazione ci vogliono DUE segni di  $=$ . Per la derivata seconda, usare due segni ' di fila, e NON il segno di virgolette ".

La sintassi con cui si chiede a Mathematica di risolvere un'equazione differenziale è:

`DSolve[equazione, y[x], x]`

**Esempi.**

**Equazione del second'ordine, lineare, a coefficienti costanti, anche non omogenea:**

`DSolve[y''[x] + 3 y'[x] - 2 y[x] == Sin[x], y[x], x]`

$$\left\{ \left\{ y[x] \rightarrow e^{\frac{1}{2}(-3-\sqrt{17})x} C[1] + e^{\frac{1}{2}(-3+\sqrt{17})x} C[2] + \frac{e^{\frac{1}{2}(-3+\sqrt{17})x} \left( -\frac{4 e^{-\frac{1}{2}(3+\sqrt{17})x} \cos[x]}{((-3-2i)+\sqrt{17})((3+2i)+\sqrt{17})} - \frac{2(-3+\sqrt{17}) e^{-\frac{1}{2}(-3+\sqrt{17})x} \sin[x]}{((-3-2i)+\sqrt{17})((-3+2i)+\sqrt{17})} \right)}{\sqrt{17}} \right. \right. \\ \left. \left. + \frac{e^{\frac{1}{2}(-3-\sqrt{17})x} \left( -\frac{4 e^{\frac{1}{2}(3+\sqrt{17})x} \cos[x]}{(3-2i)+\sqrt{17}} + \frac{2(3+\sqrt{17}) e^{\frac{1}{2}(3+\sqrt{17})x} \sin[x]}{(3-2i)+\sqrt{17}} \right)}{\sqrt{17}} \right\} \right\}$$

Come si vede, Mathematica dà l'integrale generale, chiamando  $C[1]$  e  $C[2]$  le costanti arbitrarie di integrazione.

**Equazione del prim'ordine, lineare:**

`DSolve[y'[x] + x y[x] == x, y[x], x]`

$\left\{ \left\{ y[x] \rightarrow 1 + e^{-\frac{x^2}{2}} C[1] \right\} \right\}$

**Equazione del prim'ordine a variabili separabili:**

`DSolve[y[x] y'[x] == x^2, y[x], x]`

$\left\{ \left\{ y[x] \rightarrow -\frac{\sqrt{2x^3 + C[1]}}{\sqrt{3}} \right\}, \left\{ y[x] \rightarrow \frac{\sqrt{2x^3 + C[1]}}{\sqrt{3}} \right\} \right\}$

**Problema di Cauchy:**

La sintassi è:

`DSolve[{equazione, condizioni}, y[x], x]`

dove le condizioni sono scritte con la stessa sintassi dell'equazione:

`DSolve[{y''[x] + 3 y'[x] - 2 y[x] == 0, y[0] == 1, y'[0] == 2}, y[x], x]`

$\left\{ \left\{ y[x] \rightarrow \left( \frac{1}{2} - \frac{7}{2\sqrt{17}} \right) e^{\frac{1}{2}(-3-\sqrt{17})x} + \left( \frac{1}{2} + \frac{7}{2\sqrt{17}} \right) e^{\frac{1}{2}(-3+\sqrt{17})x} \right\} \right\}$

### 2.7.9. Serie di Fourier

Non esistono comandi appositi per scrivere la serie di Fourier di una funzione: si tratta di combinare le varie tecniche viste fin qui: si definiscono le due successioni dei coefficienti, come opportuni integrali (esatti o approssimati? Dipende!), poi si definisce la successione delle somme parziali della serie di Fourier, quindi si può plottare questa per certi  $n$ ...

Si veda il notebook apposito.