

Mathematica 4.0

Dopo meno di tre anni dall'uscita della versione 3 è stata rilasciata la versione 4.0 di *Mathematica*, con alcune importanti novità, stavolta concentrate più che altro nel *Kernel*. Vediamo in dettaglio alcune delle caratteristiche più interessanti.

Introduzione

Installazione e requisiti di sistema

Mathematica arriva con un CD-Rom multipiattaforma, 3 manuali, tra cui il nuovo libro: *The Mathematica Book*, (quarta edizione). Nel pacco sono presenti le informazioni necessarie per la registrazione e la installazione. È necessaria una password specifica per la macchina con cui si lavora che si può ottenere contattando la Wolfram o l'importatore.

Mathematica 4.0 è venduta come licenza di classe A per Windows 95, 98, NT, Linux e Macintosh, e come licenza di classe B (a costo più elevato) per Silicon Graphics, IBM RS/6000, Sun Sparc, Sun Solaris, HP-UX, Digital Unix.

Sul *Macintosh* una installazione completa richiede circa 160 Mbyte di disco e almeno 20 Mbyte di RAM liberi sono necessari per fare girare qualcosa. In pratica non vale la pena di fare tentativi con meno di 64 Mbyte di RAM installata. Io ho usato un *PowerPc* G3 a 400Mhz con 320 (trecentoventi) Mbyte di RAM. L'installazione è stata rapida e senza problemi.

Le principali novità

Maggiore efficienza

È stata curata l'implementazione di molti algoritmi numerici e della rappresentazione di matrici numeriche a precisione di macchine, questo dovrebbe permettere di raggiungere per molti problemi l'efficienza di un programma compilato a basso livello conservando la facilità d'uso di un sistema altamente sofisticato.

Nuove Funzioni

La primitiva grafica **Raster**[] permette la rappresentazione di immagini bitmap a colori. Internamente sono rappresentate come un lista di triple (i valori **RGB** dei singoli bit, più alcune opzioni). Un esempio della forma interna di un grafico **Raster** è il seguente:

```
Raster[{{{60, 84, 32}, {63, 87, 35}, {64, 88, 38}, {58, 82, 32}, {47, 70, 24}, {38, 61, 15}, ..., {48, 39, 30}}}, {{0, 0}, {384, 256}}, {0, 255}, ColorFunction -> RGBColor]
```

La funzione **ListConvolve**[**k**,**h**] permette di fare la convoluzione tra due liste rendendo una lista **l** definita come:

$$l_s = \sum_{r=1} k_r h_{s-r}$$

con gli indici che variano in modo da non uscire dalle liste **k** e **h**. Per esempio:

```
In[2]:=
ListConvolve[{X, Y, Z}, {2, 3, 4, 5, 6}]
```

```
Out[2]=
{4 X + 3 Y + 2 Z, 5 X + 4 Y + 3 Z, 6 X + 5 Y + 4 Z}
```

Altre opzioni permettono molte alternative tra cui la convoluzione ciclica.

Analogamente la funzione **ListCorrelate**[**k**,**h**] permette di fare la correlazione tra due liste rendendo una lista **l** definita come:

$$l_s = \sum_{r=1} k_r h_{s+r}$$

con gli indici che variano in modo da non uscire dalle liste **k** e **h**. Per esempio:

```
In[2]:=
ListCorrelate[{X, Y, Z}, {2, 3, 4, 5, 6}]
```

```
Out[2]=
{2 X + 3 Y + 4 Z, 3 X + 4 Y + 5 Z, 4 X + 5 Y + 6 Z}
```

ListConvolve e **ListCorrelate** usano, quando possibi-

le, gli algoritmi FFT che riducono la complessità di un fattore $n/\log_2 n$ (tanto per intenderci in una convoluzione di ordine 10000 c'è un risparmio di circa un fattore 700).

L'analisi simbolica di Fourier non è più in un *package* ma è stata implementata nel Kernel. Sono implementate anche le Trasformate Z e di Laplace.

```
In[3]:=
FourierTransform[Exp[-t^2], t, ω]
```

Out[3]=

$$\frac{1}{\sqrt{2}} e^{-\frac{\omega^2}{4}}$$

```
In[4]:=
FourierTransform[1, t, ω]
```

Out[4]=

$$\sqrt{2\pi} \text{DiracDelta}[\omega]$$

```
In[5]:=
ZTransform[Exp[-t], t, z]
```

Out[5]=
e z / (e z - 1)

```
In[6]:=
LaplaceTransform[Exp[-t], t, s]
```

Out[6]=
1 / (1 + s)

Vediamo ora un esempio più complicato. Consideriamo un tono di frequenza crescente $\text{Cos}[\omega_0(1 + \alpha t)t]$ e limitiamolo all'intervallo di tempo (-1,1). la funzione $\text{UnitStep}[t]$ vale 0 per $t < 0$ e 1 per $t > 1$ e, insieme alla $\text{DiracDelta}[t]$, è riconosciuta dai programmi di integrazione simbolica usati nelle trasformate.

```
In[7]:=
f[t_] :=
UnitStep[t+1]UnitStep[1-t]Cos[80(1 + t/3)t];
Plot[f[t], {t,-1.2, 1.2},
AspectRatio -> 1/5,
PlotPoints -> 100];
```

(Vedi Figura 1)

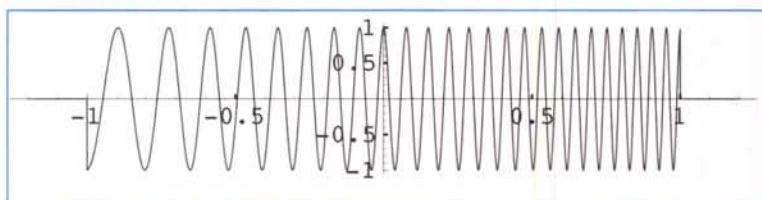


Figura 1

Se ne possono fare le tre trasformate. La trasformata di Fourier fa un po' paura.

```
In[8]:=
FF=FourierTransform[
Cos[ω0(1 + α t)t]UnitStep[t+1]UnitStep[1-t], t, ω]
```

Out[8]=

$$\frac{1}{4\sqrt{2}\sqrt{\alpha}\sqrt{\omega_0}} \left((-1)^{1/4} e^{-\frac{i(\omega+\omega_0)^2}{4\alpha\omega_0}} \left[e^{\frac{i(\omega^2+\omega_0^2)}{2\alpha\omega_0}} \text{Erfi}\left[\frac{(-1)^{3/4}(\omega-\omega_0-2\alpha\omega_0)}{2\sqrt{\alpha}\sqrt{\omega_0}}\right] + i \left(\text{Erfi}\left[\frac{(-1)^{1/4}(\omega+\omega_0-2\alpha\omega_0)}{2\sqrt{\alpha}\sqrt{\omega_0}}\right] + i e^{\frac{i(\omega^2+\omega_0^2)}{2\alpha\omega_0}} \text{Erfi}\left[\frac{(-1)^{3/4}(\omega-\omega_0+2\alpha\omega_0)}{2\sqrt{\alpha}\sqrt{\omega_0}}\right] - \text{Erfi}\left[\frac{(-1)^{1/4}(\omega+\omega_0+2\alpha\omega_0)}{2\sqrt{\alpha}\sqrt{\omega_0}}\right] \right) \right] \right)$$

Il grafico del valore assoluto di FF nei due casi $\omega_0=80, \alpha=1/3$ (in rosso) e $\omega_0=80, \alpha=1/300$ (in blu) da un'idea di quello che succede nel dominio della frequenza.

```
In[9]:=
Plot[Evaluate[{
Abs[FF/.{ω0->80,α->1/3}],
Abs[FF/.{ω0->80,α->1/300}]}],{ω, 0, 200},
PlotStyle -> {Red, Blue},
PlotPoints -> 100,
PlotRange -> All];
```

(Vedi Figura 2)

La trasformata Z della stessa funzione è invece stranamente semplicissima.

```
In[10]:=
ZTransform[
Cos[ω0(1 + α t)t]UnitStep[t+1]UnitStep[1-t], t, z]
```

Out[10]=
(z+Cos[ω0(1 + α)])/z

La trasformata di Laplace è brutta come quella di Fourier e va il risparmio.

Possibilità di importazione/esportazione

Sono stati introdotti i comandi **Import[filename, tipo]** ed **Export[filename, dati, tipo]** che supportano numerosi formati. Possibili valori di **tipo** per l'importazione di dati testuali sono i seguenti

- "Lines" linee di testo;
- "List" linee di numeri (anche in formato C o For-

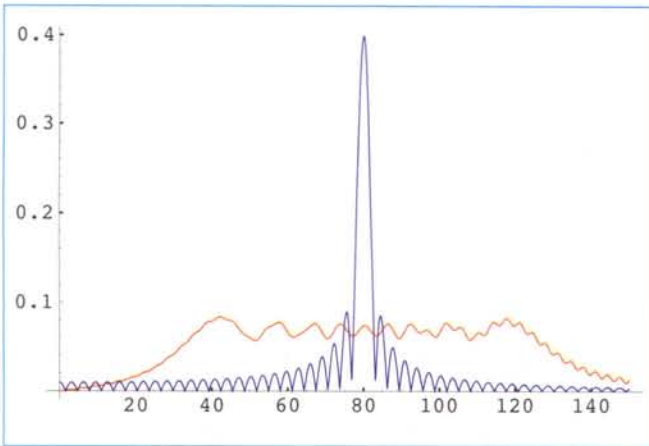


Figura 2

tran) o stringhe;

- **"Table"** array bidimensionale di numeri (anche in formato C o Fortran) o stringhe;
- **"Text"** stringa di caratteri;
- **"UnicodeText"** stringa di caratteri Unicode;
- **"Words"** words separate da spazi o CR

I seguenti valori di **tipo** permettono il trattamento di dati della forma **Graphics[]**, (le estensioni tra parentesi vengono riconosciute automaticamente):

- **"EPS"** Encapsulated PostScript (.eps);
- **"EPSI"** Encapsulated PostScript with image preview (.epsi);
- **"EPSTIFF"** Encapsulated PostScript with TIFF preview

I seguenti valori di **tipo** permettono il trattamento di dati



Figura 3

della forma **Raster[]**, (le estensioni tra parentesi vengono riconosciute automaticamente):

- **"BMP"** Microsoft bitmap format (.bmp);
- **"GIF"** GIF and animated GIF (.gif);
- **"JPEG"** JPEG (.jpg, .jpeg);
- **"PBM"** portable bitmap format (.pbm);
- **"PGM"** portable graymap format (.pgm);
- **"PNG"** PNG format (.png);
- **"PNM"** portable anymap format (.pnm);
- **"PPM"** portable pixmap format (.ppm);
- **"PSImage"** PostScript (.psi);
- **"TIFF"** TIFF (.tif, .tiff);
- **"XBitmap"** X window system bitmap.

I seguenti valori di **tipo** permettono il trattamento di suoni (trattano espressioni della forma **Sound[SampledSoundList[]]**):

- **"AIFF"** AIFF format (.aif, .aiff)
- **"AU"** μ law encoding (.au)
- **"SND"** sound file format (.snd)
- **"WAV"** Microsoft wave format (.wav).

Per esempio i comandi seguenti importano un file JPEG (riducendo i colori a 256) lo visualizzano sul monitor e lo esportano in formato GIF.

```
In[1]:=
parrots = Import["WORK:Parrots.jpeg"];
Show[parrots];
Export["WORK::Parrots.gif", parrots, "GIF"];
```

Il risultato è visibile in **Figura 3**.

Compatibilità

Il formato dei Notebook è compatibile con la versione 3. I notebook della versione 2 vanno convertiti prima di essere aperti con la 4 e i notebook della versione 4 vanno salvati nel formato vecchio prima di essere aperti con la 2.

Il codice è in larga parte compatibile con tutte le versioni i problemi riguardano le funzioni aggiunte nelle versioni 3 e 4 e le funzioni della versione 2 non più supportate.

Pacchetti Applicativi

Per chi è utente registrato di pacchetti applicativi sono disponibili gratuitamente gli aggiornamenti alla versione 4 (in pratica un nuovo pacchetto completo). Basta fare riferimento alla URL:

www.wolfram.com/products/applications/updates

Facciamo un po' di prove di velocità

Per mettere alla prova l'efficienza di *Mathematica* 4 ho rispolverato il benchmark che avevo usato nel lontano 1994 (*Lepri e Tartarughe*, MC n 136), riadattandone i parametri. Ecco le nuove prove con i relativi parametri.

- **Pigreco**: calcolo di **50000** cifre di π .
- **Fourier Transform**: il calcolo della trasformata discreta di Fourier su **40000** campioni complessi.
- **NIntegrate**: il calcolo dell'integrale tra **0** e **2** di $x \cos^2 x$, lavorando con **250** cifre decimali e chiedendo **200** cifre decimali di precisione nel risultato.
- **Symb Inv**: inversione simbolica di una matrice **7x7** (per la cronaca la matrice è $\{\sin(\pi i j/9)\}_{ij}$).
- **Cos[matrice]**: calcolo di una matrice random **1000x1000** e applicazione della funzione **Cos[]** a tutti i suoi elementi.
- **Num Inv**: inversione numerica, in precisione di macchina, di una matrice **500x500**.
- **HPr. Inv**: inversione numerica di una matrice **100x100**, lavorando con **100** cifre decimali di precisione.
- **Rat. Inv**: inversione esatta di una matrice **60x60** di numeri razionali.
- **Serie**: il calcolo simbolico dello sviluppo in serie con **35** termini di $\cos(\pi x^2+1)/\cos(x-1)$.
- **Integrate**: integrazione simbolica del polinomio risultante dallo sviluppo in serie di $\cos(\pi x^2+1)/\cos(x-1)$ con **10** termini.
- **Loop**: tre **Do** annidati per eseguire **1000000** di volte

l'istruzione nulla.

I risultati sono riportati nella tabella, (quelli più interessanti sono evidenziati in neretto):

	2.2	3	4
Pigreco	20.43	22.13	1.90
Fourier Transform	1.10	0.96	0.11
NIntegrate	10.28	10.31	10.80
Cos[matrice]	33.21	28.10	1.83
Symb Inv	2.60	4.90	4.51
Num. Inv	8.61	13.83	13.15
HPr. Inv	20.31	26.11	29.51
Rat. Inv	26.45	22.68	26.95
Serie	46.58	69.88	1.28
Integrate	29.58	33.86	13.33
Loop	8.90	5.85	5.65

In totale:
 versione 2.2, max RAM 45.95 Mb, Tempo totale 208.07 sec.
 versione 3, max RAM 47.03 Mb, Tempo totale 238.65 sec.
 versione 4, max RAM 27.50 Mb, Tempo totale 109.04 sec.

La versione 4 è straordinariamente più veloce per il calcolo di Pigreco e per la trasformata di Fourier (vedremo presto alcune interessanti applicazioni). Anche il calcolo simbolico di uno sviluppo in serie e l'integrazione simbolica sono molto più efficienti (almeno nei casi in esame). Ci si perde qualcosa per l'inversione numerica e simbolica rispetto alla versione 2. Globalmente (per il test proposto) il risparmio di tempo e di RAM realizzato utilizzando la versione 4 è circa del 50%.

Conclusioni

Vale la pena aggiornarsi alla versione 4? Chiaramente la scelta è soggettiva. L'upgrade è consigliabile a tutti coloro che dispongono di una macchina potente con tanta RAM e la vogliono sfruttare al massimo con *Mathematica*. Chi passa alla 4 dalla 2 deve mettere in conto la necessità di convertire ed adattare i Notebook.

In ogni caso, una volta trascorso il necessario periodo di ambientamento, *Mathematica* 4 può dare grandissime soddisfazioni a chiunque voglia affrontare con il calcolatore complessi problemi matematici (e sappia dove mettere le mani).

MS

Indirizzi utili

Produttore: www.wolfram.com; e-mail: info@wolfram.com

Distributore Italiano: SciSoft, Via Amendola, 12, 10121 Torino, Tel: 011-5176580, Fax: 011-5176293; e-mail: info@scisoft.it

Bibliografia

Stephen Wolfram, **The Mathematica Book, 4rd ed.** (Wolfram Media/Cambridge University Press, 1999)