

PROVA



Mathematica

di Raffaello De Masi

Nel 1958, frequentavo allora le medie, comprai attraverso la pubblicità su una rivista un apparecchio di plastica che consentiva, almeno secondo quanto diceva la pubblicità, calcoli matematici fulminei pur senza adottare tecniche comuni alle allora nascenti calcolatrici tascabili (si fa per dire, visto che avevano le dimensioni di un mattone, erano a led rossi o verdi, e divoravano le batterie come piraña). L'acquisto si rivelò rappresentato da un aggeggio della grandezza di un regolo calcolatore tascabile, che, attraverso un ingegnoso sistema di slitte di plastica spesse come stuzzicadenti, e mutuando l'algoritmo di soluzione dall'abaco-pallottoliere degli egizi, permetteva, effettivamente, con un poco di esercizio continuo, di eseguire con una certa speditezza addizioni e sottrazioni (alle

moltiplicazioni e divisioni non arrivai mai, anche perché la pazienza e la tenacia di esercizio non sono il mio forte).

Ho tenuto, comunque, da parte per molti anni questo aggeggio, anche per rendere omaggio alla ingegnosità dell'inventore; più pratico di un regolo, ovviamente nel campo delle quattro operazioni, comodo da portare, permetteva di fare, oltre tutto scena con le compagne di scuola, affascinate da questo «calcolatore automatico», come veniva definito nel foglio d'istruzioni.

Signori, abbiamo appena parlato della Topolino Fiat! Permetteteci di presentare adesso la Ferrari Mondial del computer algebrico-matematico, il programma che è oggetto di questo articolo!

Pensate che TK!Solver faccia cose egregie nel campo della risoluzione algebrica? Vi meravigliate della sua capacità

di elaborare grafici da una funzione definita in una delle sue finestre! Preparatevi a fare un salto di qualità senza pari; stiamo per entrare non in un programma, ma direttamente in un ambiente matematico!

Il package

Quando Marco mi ha mandato il pacchetto con la preghiera (si fa per dire!!) di eseguire la relativa prova, non mi preoccupai più di tanto e pensai: «Ecco qualche nuova elucubrazione notturna di qualche svitato che ha sviluppato una nuova astrusità per prolungare il calvario di qualche inoffensivo redattore di riviste di informatica». La cosa che più mi pareva strana era la composizione per lo meno anomala del package: cinque dischi di programma più un demo, uno

Mathematica

*A System for Doing Mathematics
by Computer versione 1.1*

Produttore:
Wolfram Research, Inc
P.O. BOX 6059 Champaign, IL 61821 U.S.A.

Distributore:
A.I.S. Spa, Via Rombon 11, 20134 Milano

Prezzi:

Vers. per Mac Plus	L. 1.200.000
Vers. per Mac II	L. 1.800.000
Vers. per MS-DOS 386	L. 1.600.000
Vers. per 386 con 387	L. 2.200.000
Vers. per 386/Windows	L. 2.800.000

striminzito manuale di istruzioni, una serie di fogli volanti dell'ultima ora, e un pesante volume, elegantemente rilegato, scritto dall'implementatore del programma e gran patron della software house costruttrice, pubblicato però da Addison-Wesley.

La cosa si è fatta ancora più curiosa quando ho aperto il manuale d'istruzioni; esso serve solo come sottomano per l'utente, e rimanda per tutta la fase di autoistruzione al libro appena citato; allora ho pensato di vederci un po' più chiaro in questa procedura un poco strana, e, quasi per incanto mi sono trovato a navigare in qualcosa di assolutamente impensabile.

Il pacchetto, riassumiamolo, è composto dal volume appena citato, di ben 750 pagine, di un manuale utente stringato (circa 150 pagine, rappresentate in buona parte da note di riferimento ai menu e alle opzioni), di 6 dischetti (di cui 3 di programma, uno di utility e uno di esempi, oltre al demo, che gira anche su macchine di 1 M), di un manuale di avvio (in cui viene descritto tra l'altro, un virus checker originale), un manuale avanzato, una serie di aggiornamenti, garanzie, fogli di registrazione e prenotazioni per gli upgrade, e così via!

Abbisogna di almeno 4 mega di memoria centrale, anche se, con due, è possibile leggere e visualizzare alcuni esempi presenti sull'ultimo dischetto. È possibile a parte richiedere anche il demo separato, che illustra le caratteristiche principali e alcuni dei risultati più interessanti e appariscenti del pacchetto.

Il programma

Il package, all'apertura, mostra una finestra in cui sono ricordati gli implementatori del pacchetto, in tutto otto, compreso il gran patron Stephen Wolfram. Ma che cosa è Mathematica?

Semplice, almeno a parole, la risposta: Mathematica è un sistema generale, un ambiente per eseguire calcoli matematici. Esso può essere utilizzato in diversi modi e a differenti livelli; la sua potenza sta nel fatto che non si limita, come una normale calcolatrice, a eseguire calcoli numerici, ma può sviluppare calcoli e operazioni algebriche, simboliche e grafiche.

Ma Mathematica è anche un linguaggio, in possesso di sue definizioni e statement, e in cui l'utente può porre e definire sue istruzioni. È possibile quindi scrivere veri e propri programmi, lavorando combinatamente non solo con numeri, ma anche con funzioni algebriche, notazioni simboliche ecc. tutti combinati insieme.

È possibile ancora utilizzare Mathematica come linguaggio per rappresentare conoscenze matematiche; il gran vantaggio è lo scarso formalismo necessario per eseguire ciò; tanto per intenderci è praticamente possibile ricavare da libri o riviste espressioni e relazioni matematiche, e introdurle direttamente nell'ambiente, quasi del tutto senza modifiche. In questo caso l'approccio alla formula è totalmente invertito rispetto a tutti gli altri linguaggi; qui occorre adattare ciò che si inseriva all'idioma, in Mathematica è sufficiente «istruire» il programma su come trattare la formula stessa.

Infine, e non è tutto, Mathematica è costruito in modo da lavorare in collaborazione con altri programmi; è possibile preparare Mathematica per approntare output verso altri pacchetti, o per ricevere dati da essi; ciò è anche possibile in quanto il package rispetta standard diffusi e sperimentati, come Unix e PostScript, che possono essere considerati, nel loro genere, gli standard de facto.

Ma analizziamo un poco più da vicino quanto abbiamo finora detto. Avevamo infatti espresso diverse affermazioni che ci interessa verificare e dimostrare.

Con Mathematica, come con un linguaggio di intelligenza artificiale, si «dialoga»; il manuale e il volume di accompagnamento usano, per rappresentare tale dialogo, alcune convenzioni che qui riportiamo, anche per meglio identificare ciò che successivamente scriveremo.

Le linee che iniziano con «In» sono quelle che l'utente (Input) batte alla tastiera; quelle indicate con «Out» (Output) rappresentano la risposta di Mathematica. Un esempio di calcolo matematico potrebbe essere rappresentato dalla richiesta:

Trovare il valore numerico di $\log(4\pi)$

Espresso in linguaggio Mathematica avremo:

In[1]=N[log[4 Pi]]

[1], dopo In, indica che si tratta del primo quesito, il simbolo di [=] è posto automaticamente, N indica al programma che si desidera un valore numerico, il resto è abbastanza intuitivo se si considera solo che [Pi] è una costante, che l'ambiente già conosce e che equivale a π .

Avremo come risposta:

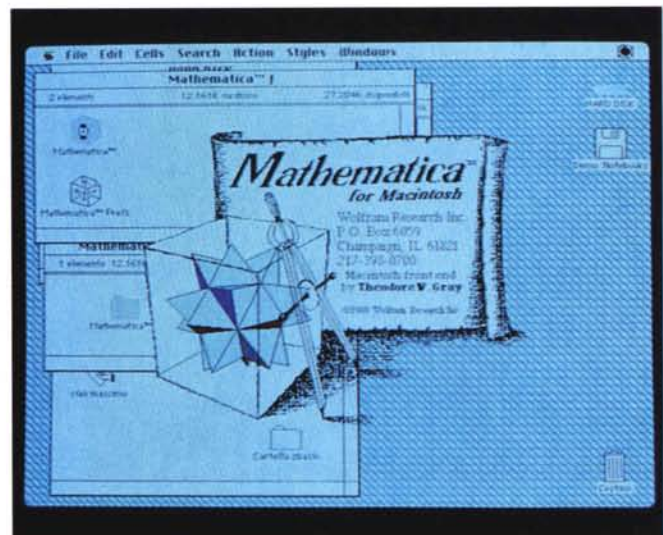
Out[1]=2.53102

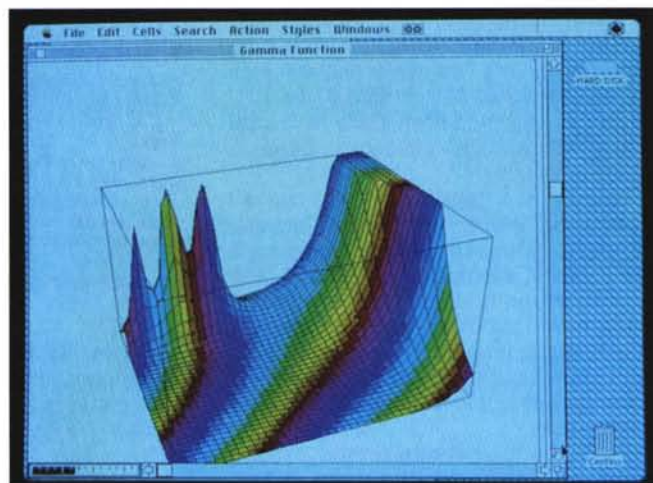
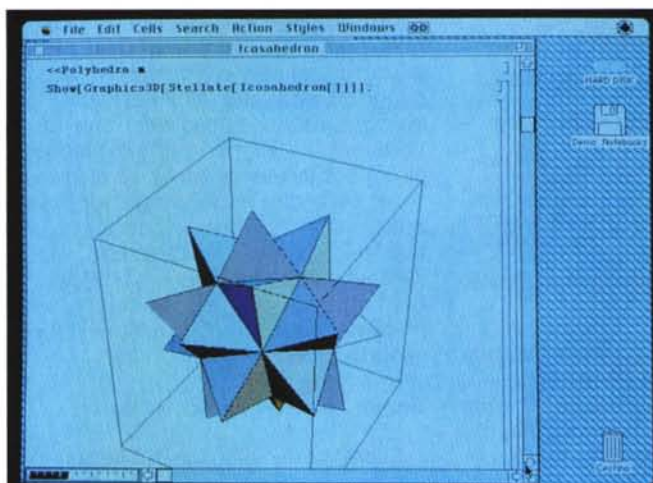
Battiamo adesso:

In[2]=n[log[4 Pi], 40]
Out[2]=53102424696929070297789159426-9411847798

Il 40, dopo 4 Pi permette di scegliere il numero di cifre decimali, che nel nostro caso sono state 40. Al più semplice livello Mathematica esegue calcoli

La videata di introduzione al pacchetto.





Due esempi di grafica tridimensionale.

numerici, né più né meno come una calcolatrice comune, ma con due vantaggi; quello di adottare la precisione voluta, da una parte, e quello di possedere una potenza di calcolo ben superiore anche alle più potenti calcolatrici (ivi comprese le famigerate HP), in quanto include un ampio range di funzioni matematiche, incluse le più sofisticate funzioni della fisica matematica.

Mathematica lavora non solo con singoli numeri, ma con strutture anche complesse. È possibile, ad esempio, eseguire operazioni su matrici, o accedere agli statement già implementati e relativi ai calcoli di algebra numerica lineare; è possibile utilizzare agevolmente il pacchetto, ad esempio, per la ricerca delle trasformate di Fourier, dei valori di incognita dei polinomi di grado (n), per la ricerca degli autovalori, per

l'uso delle più potenti e diffuse funzioni statistiche e finanziarie.

È possibile ancora eseguire operazioni matematiche su funzioni particolari, come integrazione numerica e ricerca dei minimi.

Altrettanto semplice è eseguire calcoli simbolici: una delle maggiori comodità di Mathematica è la possibilità di maneggiare formule simboliche allo stesso modo di formule algebriche e numeriche. In questo modo è possibile espandere, fattorizzare e semplificare polinomi ed espressioni razionali. Come dicevamo precedentemente, è possibile risolvere equazioni polinomiali o sistemi di equazioni virtualmente di qualunque grado; ancora, ritornando all'esempio precedente, è possibile ottenere risultati algebrici da qualunque tipo di operazione su matrice.

Un esempio in proposito è dato dalla ricerca dell'integrale:

$$\int x^4/(x^2-1)dx$$

batteremo alla tastiera:

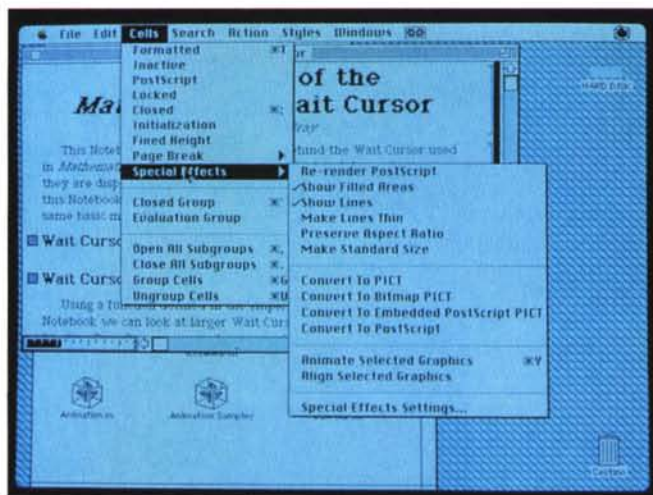
```
In[3]=x^4/(x^2-1)
Out[3]=x^4/
      -1+x^2
```

continueremo poi dicendo:

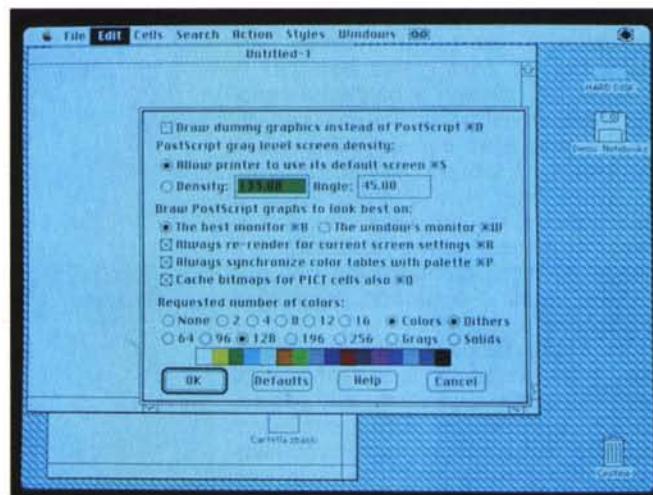
```
In[4]=Integrate[%x]
Out[4]=x+x3/3+log[-1+x]/2-Log[1+x]/2
```

L'unica cosa non immediatamente chiara in tutto ciò è il simbolo %, che indica alla formula che ci si sta riferendo all'ultimo output.

Ancora, come se non bastasse, è



La disponibilità di effetti per l'output.



Le possibilità di display; in output i colori possono essere trasformati in retini Post Script.

possibile usare Mathematica per eseguire e sviluppare grafici in 2 e 3 dimensioni; ad esempio, si desidera plottare la funzione di $\sin(x,y)$ per x e y compresi tra 0 e π ; batteremo:

```
In[5]=Plot3D[Sin(x,y),{x,0,Pi},{y,0,Pi}].
```

L'ordine genera un disegno in tre dimensioni della funzione seno (x, y) nell'intervallo desiderato, intervallo che per le due variabili viene indicato in parentesi graffa (quello delle opzioni è un campo vastissimo dell'ambiente). È possibile plottare funzioni o liste di dati, in due o tre dimensioni, e, in questo ultimo caso, ottenere risultati abbastanza realistici, includenti ombreggiature, colori, o effetti speciali di illuminazione.

È possibile usare Mathematica anche per eseguire disegno geometrico; in questo caso è sufficiente introdurre la rappresentazione simbolica di un poligono, ad esempio, per produrre un output dello stesso, che, nel caso di figure solide, può essere in tre dimensioni (si vedano, ad esempio, le bellissime figure).

Mathematica è ancora, come dicevamo, un linguaggio di programmazione, molto somigliante, per certi versi, al C. Al contrario di questo, però, è un interprete, per cui è possibile lanciare immediatamente il programma appena battuto e usufruire degli innegabili vantaggi dei linguaggi non compilati.

Come linguaggio si tratta, almeno dal punto di vista matematico, di qualcosa di veramente mostruoso; praticamente non esiste funzione o regola matematica che non vi sia già implementata o che non sia ottenibile con facili passaggi; immaginiamo di cercare, ad esempio, i numeri primi compresi in un intervallo; batteremo:

```
In[6]:=f[n.]:=Table[Prime[i], {i,n}]
```

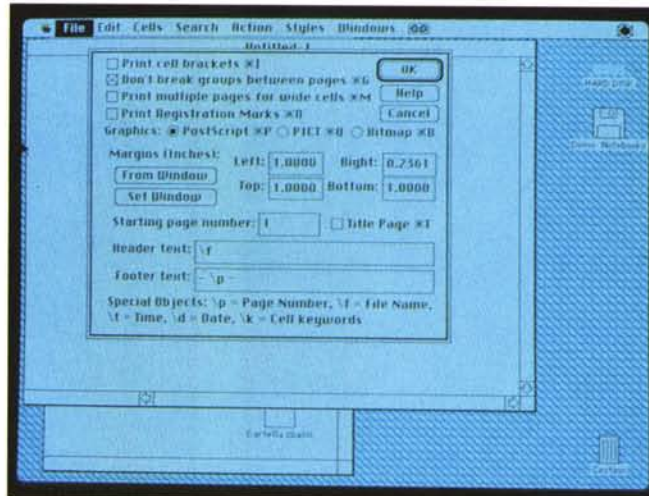
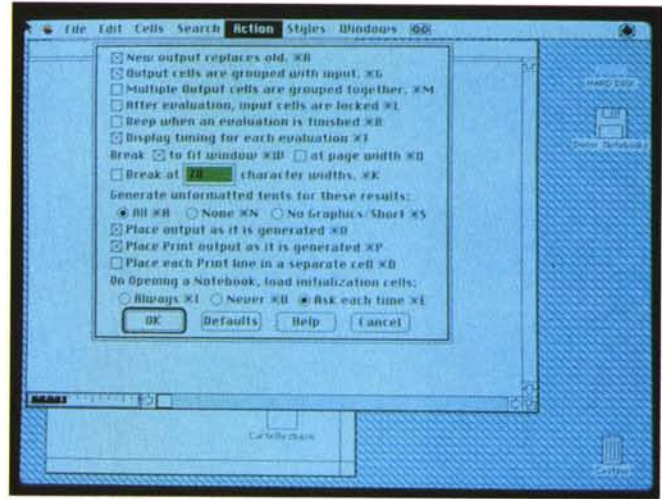
Abbiamo cioè definito una funzione $f[n]$ destinata a calcolare i primi n numeri primi della scala naturale. La disponibilità dell'interprete ci consente di testare immediatamente la definizione così da avere:

```
In[7]:=f[10]
Out[7]={{2,3,5,7,11,13,17,19,23,29}}
```

che rappresenta appunto la tabella dei primi dieci numeri primi.

Dicevamo ancora che Mathematica è un sistema per rappresentare «conoscenze» matematiche, vale a dire, in altri termini, un metodo rappresentativo per l'uso di formule matematiche. In questa ottica è fondamentale tener conto delle cosiddette «regole di trasforma-

*Le possibilità di
settaggio delle
«azioni» da eseguire
durante l'impiego del
Kernel.*



La finestra di output
«Pagesetup».

Mathematica, cosa è

Mathematica esiste in versione sia per Macintosh (quella che abbiamo provato) che per MS-DOS.

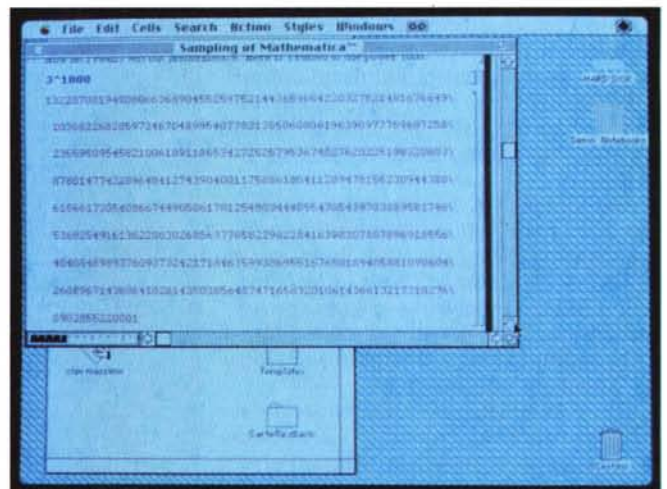
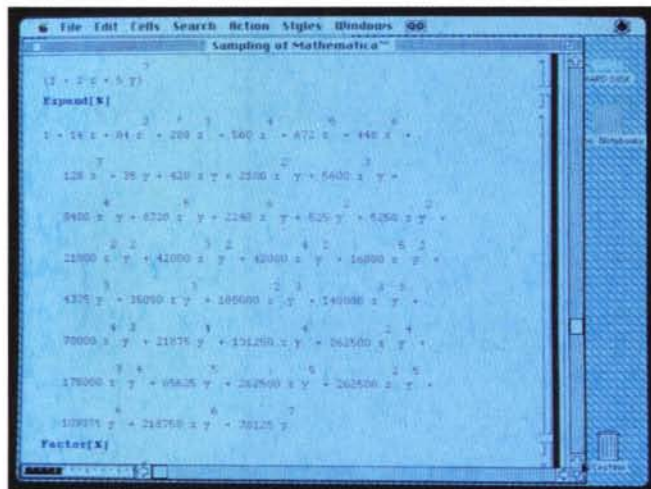
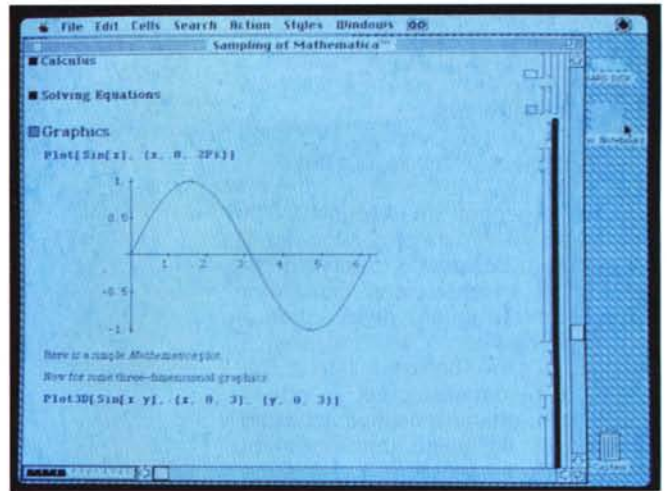
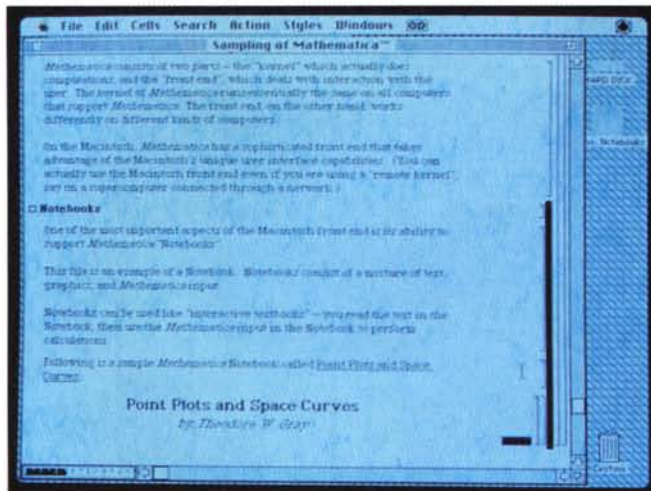
È ottenuto da un programma redatto in C, lungo, per il blocco del «kernel», circa 150.000 linee. L'interfaccia Mac è lunga circa 15.000 linee che si aggiungono alle precedenti. Il codice sorgente è stato trattato con una versione object-oriented del C originale Wolfram, e poi precompilato in standard C.

La sezione sviluppata autonomamente dalla Wolfram era necessaria in quanto doveva contenere implementate alcune assunzioni circa il calcolatore cui era destinato. Ciò ha consentito di bypassare la fase di aggiustaggio alla macchina, e ha reso praticamente eguali tra di loro le diverse versioni del listato originale. Come risultato, almeno per quanto attiene al «kernel», le diverse versioni di Mathematica provengono praticamente dallo stesso codice sorgente.

Mathematica utilizza una tecnica originale di gestione della memoria in modo dinamico, cosa che consente di utilizzare praticamente tutta la memoria disponibile sul particolare computer, escludendo qualche operazione di «garbage collection».

Mathematica inoltre rappresenta una sintassi di diversi tipi di software; infatti possiede caratteristiche proprie di:

- linguaggi interattivi numerici, come il Basic
- sistemi e ambienti numerici specializzati, come TKISolver, Eureka, MathCad, MathLab, ecc.
- sistemi di elaborazione algebrica, come Macsuma, Maple, Reduce, SPM, ecc.
- linguaggi grafici interpretati, come PostScript
- linguaggi di manipolazione di liste simboliche, come APL, Lisp, o Prolog
- linguaggi di programmazione strutturati, come il C.



Alcuni esempi di tecniche risolutive, numeriche e grafiche.

zione», in base alle quali è possibile trasformare, appunto, espressioni di una forma in espressioni di forma diversa. Si tratta di regole destinate, attraverso una serie di convenzioni, a rappresentare molti campi di relazioni matematiche.

In altri termini viene, in Mathematica adottata una serie di convenzioni, peraltro completamente compatibili con le normali conoscenze matematiche, che permettono al programma di comprendere e acquisire «conoscenze» algebriche e logiche proprio dell'ambiente numerico. Le convenzioni sono poche, e piuttosto intuitive, né la lettura della formula «trasformata» cozza con la logica o con le conoscenze matematiche già in possesso dell'utente.

Ancora, Mathematica è un ambiente completo di calcolo e di elaborazione di dati numerici. È questo il campo di applicazione principe del pacchetto, che permette di redigere, organizzare, docu-

mentare e lanciare i calcoli e i programmi redatti dall'utilizzatore; è se così si può definire il background di base dell'ambiente come un sistema operativo inserito in un sistema operativo, ed è composto di due parti principali.

La prima, il «kernel», il motore, il cuore, il nucleo, il nocciolo del sistema è comune a tutte le implementazioni del programma stesso (si ricordi che di Mathematica esistono implementazioni diverse, come già detto, operanti sotto Unix, OS2, MS-DOS, oltre che in ambiente Mac), ed è quasi completamente trasparente all'utente; esso si occupa essenzialmente di eseguire i calcoli, e, possedendo i medesimi algoritmi risolutivi nelle diverse implementazioni, funziona allo stesso modo e secondo gli stessi criteri su tutte le macchine.

La seconda, il cosiddetto «front-end» che, soprattutto in Mac, è rappresentato da una interfaccia estremamente sofisticata, rappresenta l'ambiente di interazio-

ne con l'utente; esso, ovviamente, varia da sistema operativo a sistema operativo; nel nostro caso, in Macintosh intendendo, il front-end mette a disposizione dell'utente i tool grafici e di I/O tipici di Mac; c'è da dire che gli implementatori dimostrano, in questo caso, una approfondita conoscenza del toolbox, soprattutto per quanto attiene alla manipolazione del testo. La maggior parte dei front-end di Mathematica supportano i cosiddetti «notebook», blocchi degli appunti che contengono stralci di testo, grafica e definizioni di Mathematica. Si tratta di veri e propri subprogrammi, o utility se così si vuole, inseribili direttamente nell'ambito del programma principale quale facility di programmazione e documentazione: l'esempio più tipico dell'uso di un notebook è quello di leggere il testo (di una definizione, di una funzione, di una regola) dal foglio d'appunti e di inserirlo, tal quale o modificato e commentato per la bisogna, nel programma che si sta

La programmazione in Mathematica

Tutto quanto abbiamo detto nel testo dell'articolo ha dimostrato, almeno lo speriamo, la grande potenza del pacchetto; consentiteci qui, un po' in disparte di parlare di una delle caratteristiche che a noi è parsa più entusiasmante, la possibilità di programmazione. Mi rendo conto che questo potrebbe essere un argomento di non eccessivo interesse per tutti i lettori, ma la possibilità di avere a disposizione un idioma capace di utilizzare tutto il ben di dio built-in nel package è troppo forte perché di esso non si dica almeno qualche parola.

Esistono tre differenti approcci alle tecniche di programmazione di Mathematica. La scelta di una di esse dipende dal patrimonio culturale dell'utente e, in minore misura, dal tipo di problema che occorre risolvere.

Tutti gli approcci sono, ad una analisi finale, equivalenti, almeno per quanto attiene ai risultati, ma uno di essi può essere più efficiente nella soluzione di un problema di un altro, o attagliarsi in maniera più efficace al modo di ragionare dell'utente.

I tre tipi di programmazione possono essere così definiti e riassunti:

- programmazione procedurale, basata su algoritmi che specificano, passo passo, le operazioni da eseguire;
- programmazione funzionale, che indica una serie di funzioni da applicare per la risoluzione del problema;
- programmazione matematica, basata essenzialmente su pure relazioni matematico-algebriche.

Il primo approccio è quello tipico dei vecchi linguaggi tradizionali dell'informatica, come il Pascal, il Basic, il Fortran. La più semplice esecuzione di un programma procedurale consiste in una sequenza di comandi che vengono eseguiti uno per uno in ordine. Lo status del programma in un determinato punto dipende dal valore delle variabili attualmente in uso. Si tratta di un sublinguaggio comunque di altissima efficienza, ben più efficace, ovviamente nei limiti di competenza, degli

idiomi precedentemente indicati. La seconda tecnica si basa essenzialmente sulla caratteristica di Mathematica di offrire la possibilità di costruire il proprio programma in pezzi, invece di costruirne uno solo, monolitico; in altre parole è possibile realizzare blocchi isolati, funzioni (si noti la gran somiglianza col C) che poi concorrono insieme alla soluzione finale. Non necessariamente si tratta di una tecnica più efficiente e produttiva, ma la possibilità di avere un programma rappresentato da blocchetti nidificati tra loro consente una modularità e una efficienza di manutenzione molto più efficace. In questa ottica Mathematica è particolarmente versato nella tecnica object-oriented, particolarmente efficace nell'organizzare codici particolarmente complessi (pensate solo alle 150.000 righe del package).

L'idea di base di questo metodo è quella di collezionare insieme non funzioni che eseguono lo stesso genere di operazioni (tecnica propria dei moduli), ma funzioni che agiscono sullo stesso genere di oggetti. Abbreviando il discorso, che qui non è ovviamente possibile trattare, diremo, forse un po' semplicisticamente che con questa tecnica si introducono e definiscono i diversi tipi di oggetto, e quindi si specificano i «metodi» per eseguire su di essi le differenti operazioni. L'ultima forma di programmazione, estremamente specializzata, si basa su una redazione del codice cosiddetta rule-based. Una delle più potenti caratteristiche di Mathematica è la sua abilità di eseguire operazioni specificando una collezione di regole di trasformazioni. L'idea di base è che ogni volta che Mathematica incontra una regola o una espressione, essa tenta di trasformarla in una di sua conoscenza.

Questa tecnica è estremamente sofisticata e fornisce risultati inimmaginabili con le due precedenti, ma abbisogna di conoscenze profonde di analisi del problema, per cui è possibile ritenerle appannaggio solo di programmatori avanzati.

nella maniera più efficiente ed efficace possibile. Nulla manca per una perfetta comprensione del pacchetto e differenti modi sono previsti per l'accesso; persone senza alcuna conoscenza di tecniche di programmazione, esperti con avanzate cognizioni matematiche, tecnici con esperienza avanzata anche nel campo dei linguaggi di programmazione... Ricercatori, studenti, addetti e non ai lavori possono indifferentemente raggiungere, anche all'estrema facilità d'uso, i massimi livelli possibili; viene richiesta solo una conoscenza matematica di base adeguata ad una scuola superiore.

L'uso del programma è estremamente facile, almeno all'inizio; la cosa mi-

gliore è affidarsi alla lettura ordinata e regolare del manuale, che introduce in maniera organica ed efficace alle varie componenti del sistema. Si parte da una introduzione essenzialmente pratica di Mathematica, in cui vengono illustrate le diverse tecniche numeriche, la costruzione dei calcoli, l'illustrazione della gran messe di formule già comprese nel sistema e la tecnica di costruzione di altre, le liste e la loro manipolazione, l'interfacciamento con altri sistemi. Seguono operazioni più avanzate, con la definizione formale delle regole di trasformazione, gli operatori speciali, la programmazione procedurale, i tool avanzati di I/O, la manipolazione delle risorse. L'ultima fase è riservata alle tecniche più specialistiche e avanzate, comprendenti le funzioni matematiche speciali, quelle razionali e polinomiali, le tecniche di manipolazione delle equazioni, i limiti e le serie, l'algebra lineare, le operazioni numeriche su dati e funzioni.

Una sezione a parte è dedicata alle metodologie di programmazione e alla illustrazione del linguaggio formale di Mathematica, con l'interfacciamento con altri programmi. La seconda parte del libro è rappresentata da una guida di riferimento formale, e da una nutrita serie di esempi-tutorial, da cui sono stati presi quelli che vedete nelle figure.

Conclusioni

Mathematica è un tool di sviluppo in ambiente dedicato di estrema potenza (ne è espressione significativa la dimensione stessa del codice) rapido, veloce, efficiente, complesso quanto basta per non scoraggiare il principiante e per invogliare l'esperto. Frutto di un lavoro di programmazione a dir poco monumentale, complesso e di eccezionale qualità, è anche estremamente elastico, user-friendly, e dotato di capacità grafiche a dir poco superiori a tutto quello che si era finora visto in commercio.

Rappresenta il campo ideale di sperimentazione nella ricerca, mentre non ci pare che possa sostituire agevolmente programmi come TK! o Eureka, e questo non perché questi siano più efficienti di esso ma per la differenza che esiste e esisterà sempre tra pacchetti orizzontali e verticali.

Palestra eccezionale per lo studente, si trasforma, in mano a professionisti o ricercatori, in mezzo di lavoro insostituibile e difficilmente eguagliabile. Il costo, naturalmente, è in assoluto piuttosto elevato: ma, a ben pensarci, non potrebbe esser altrimenti e, d'altra parte, è giustificato sia dalla qualità del prodotto sia dal fatto che non si tratta certo di un oggetto di largo consumo. ■

sviluppando. Infine, Mathematica permette di redigere output utilizzabili direttamente in altri ambienti di programmazione, come Fortran e C. Produce anche documenti di tipo TEX, e testo e grafica in formato (già lo avevamo accennato) PostScript.

L'uso del programma

Il pacchetto è organizzato in modo estremamente pratico ed efficace: attento all'uso del manuale-libro e l'accesso all'immenso patrimonio del software complementare al programma principale, l'utente viene a poco a poco introdotto nel cuore del programma-ambiente