a cura di Corrado Giustozzi

Introduzione alla grafica

Gli elaboratori elettronici hanno cessato da tempo di essere dei semplici manipolatori di numeri o generatori automatici di bollette del telefono: oramai, con la realtà virtuale, si cerca di stimolare attraverso i computer tutti i nostri sensi e la grafica computerizzata è una tecnologia molto consolidata in tutte le sue forme. In questo articolo cerchiamo di dare un'idea (forzatamente parziale e incompleta) delle possibilità grafiche di Mathematica nei vari campi applicazione

di Francesco Romani

La struttura della grafica

La produzione di output grafici al calcolatore avviene essenzialmente in due modi diversi: da un lato esistono applicazioni specifiche per i particolari campi di applicazione: CAD bidimensionale, CAD 3D, grafici matematico-statistici, animazione, ritocco fotografico etc.; d'altra parte, qualunque linguaggio di programmazione interfacciato con un sistema operativo e un hardware adeguati permette di generare qualunque tipo di grafico. In pratica, per qualunque esigenza specifica esiste un prodotto che permette di farla al meglio (ad esempio le applicazioni creative sono più semplici con i programmi di disegno) ma la sinergia tra la grafica programmabile e la potenza di calcolo permette una enorme flessibilità e vastità di applicazioni intelligenti. La scelta di base per la grafica adottata dalla Wolfram è quella, generale, della più totale programmabilità da parte dell'utente.

Uno dei principali problemi quando si desidera ottenere output grafici al calcolatore è quello della interfaccia tra il software e i dispositivi fisici di uscita. La estrema varietà di quest'ultimi (monitor video, televisori PAL, stampanti, plotter etc.) e le solite ragioni storiche e commerciali rendono difficile una standardizzazione uniforme della materia e, quasi ogni mese, escono nuovi formati e nuove periferiche. *Mathematica* risolve il problema in un modo abbastanza macchinoso che permette però una buona compatibilità con le piattaforme su cui è implementato.

Per comprendere l'uso delle primitive grafiche bisogna notare che in Mathematica molte funzioni hanno un significato puramente simbolico (vedi riquadro). La primitiva

```
Line[{{a,b},{c,d}}]
```

è la rappresentazione simbolica di una linea che congiunge il punto (a,b) con il punto (c,d) del piano cartesiano, e la funzione **Line** non ha alcuna implementazione.

La funzione **Graphics** (anch'essa non implementata) riceve come argomento una lista di primitive grafiche e costituisce la rappresentazione simbolica di un grafico bidimensionale. In modo analogo esistono altre funzioni (**Graphics3D**,**SurfaceGraphics3D**,**ContourGraphics**, **DensityGraphics**, **GraphicsArray**) che costituiscono la rappresentazione simbolica di grafici di altro tipo.

La funzione **Show** (per cui esiste una implementazione interna) mostra sullo schermo il grafico associato ad una rappresentazione simbolica. Facendo una analogia con le operazioni aritmetiche sui numeri razionali, **Show** svolge una funzione analoga a quella di **Plus**: riceve una rappresentazione simbolica (una espressione) e disegna il grafico associato a quella rappresentazione.

Ad esempio il seguente programma:

```
Show[Graphics[
  {Line[{{0,0},{1,1}}],
   PointSize[0.08],
   Point[{0.2,0.8}],
   PointSize[0.14],
   GrayLevel[0.5],
   Point[{0.8,0.2}]}]
```

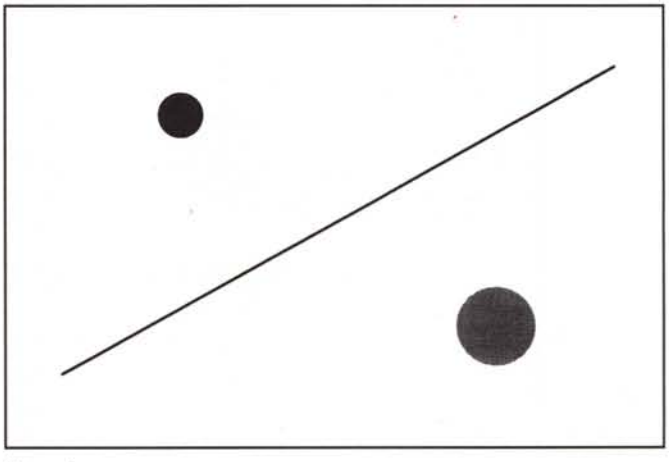

Figura 1

disegna una linea e due pallini, uno nero e uno grigio.

In una macchina (come il Macintosh e il NeXT) che usi il linguaggio di descrizione grafica PostScript della Adobe **Show** traduce la espressione **Graphics[...]** ricevuta come argo-

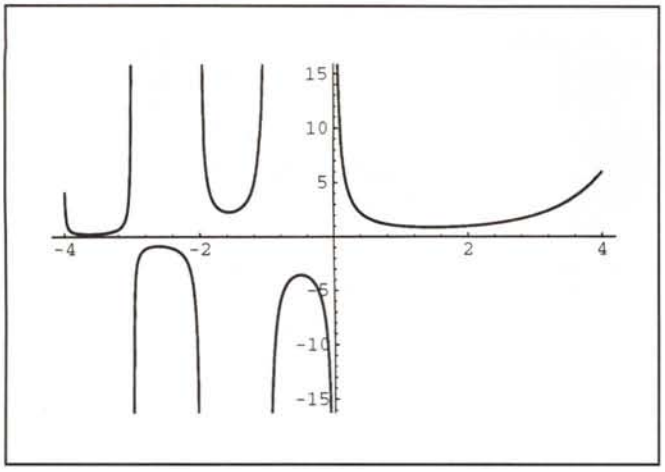

Figura 2: Plot [Gamma[x],{x,-4,4}]

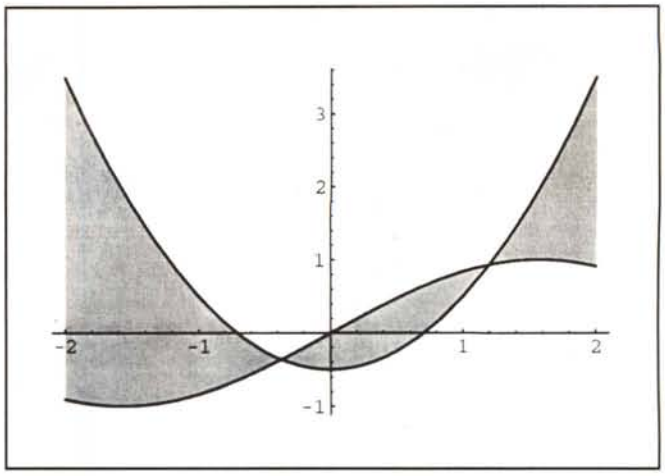

Figura 4: FilledPlot[{Sin[t],t^2-0.5},{t,-2,2}]

mento in una rappresentazione PostScript che viene memorizzata nel documento.

A questo punto il *Front-End*, attraverso il linguaggio grafico della macchina su cui si lavora (il PICT per il Mac, lo stesso PostScript per il NeXT, le primitive X-Windows per altre macchine Unix, una — ahimé terribile — simulazione a caratteri per i terminali TTY) mostra sullo schermo il grafico desiderato (o qualcosa che gli somiglia). Inoltre la rappresentazione PostScript può essere vista esplicitamente se si apre il notebook come file di tipo testo o si chiede al *Front-end* di sformattare la cella).

Se si vuole stampare il grafico su una periferica PostScript non ci sono ulteriori problemi, altrimenti è necessaria un'altra conversione di formato. Sulle macchine dotate di interfaccia tipo Mac la cella grafica può essere copiata e "incollata" in altre applicazioni. In particolare sul Macintosh, a partire dalla versione 2.1 si possono trasformare grafici singoli o intere animazioni nel formato Quicktime e gestirli senza problemi nelle applicazioni multimediali ora disponibili in gran copia. Tanto per fare un esempio, in meno di un minuto si può produrre un documento Microsoft Word 5 che mostra (sullo schermo) animazioni di grafici tridimensionali di *Mathematica*.

Questa spiegazione avrà probabilmente sconcertato il lettore, ma — vi assicuro — in pratica, e finché non si cerca di fare operazioni particolari come la generazione di immagini ad alta qualità per la stampa professionale, l'intero processo è abbastanza trasparente. Per di più esistono numerose funzioni grafiche ad alto livello che permettono di risolvere molti problemi senza neppure venire a conoscenza della struttura interna dei grafici di Mathematica.

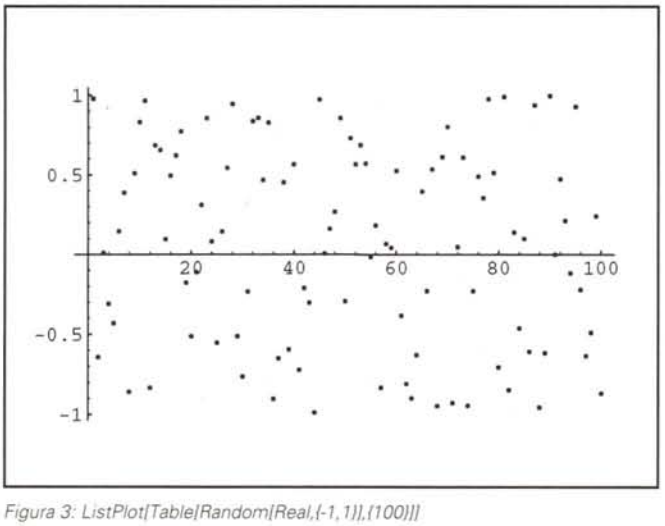

Figura 3: ListPlot[Table[Random[Real,{-1,1}],{100}]]

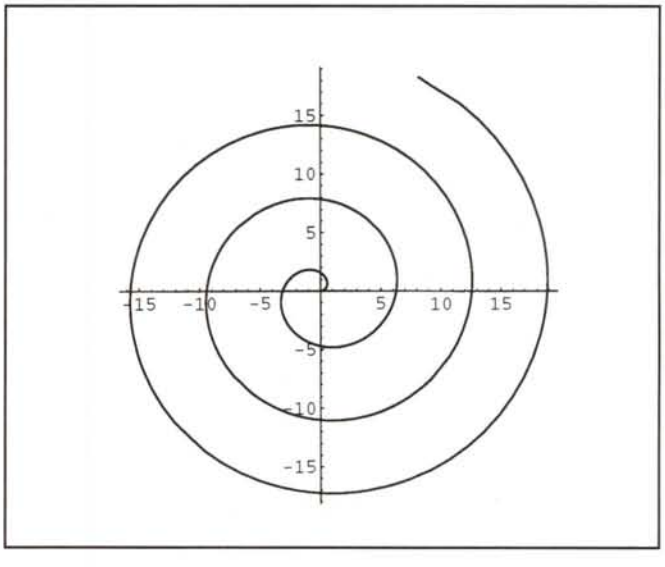

Figura 5: ParametricPlot[{t Cos[t], t,0,20},AspectRatio≥1] ▶

Grafici bidimensionali

La applicazione grafica più naturale per il matematico è il semplice plottaggio di una funzione di variabile reale (anche con eventuali singolarità) in un intervallo dato. La funzione **`Plot`** soddisfa tale necessità. **`Plot[Gamma[x],{x,-4,4}]`** traccia il grafico della funzione Gamma tra -4 e 4 e le singolarità non creano problemi.

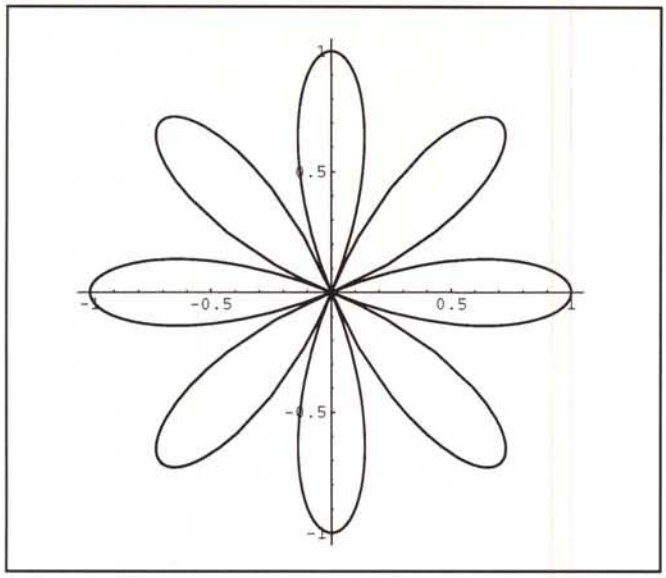

Figura 6: PolarPlot[Cos[4t],{t,0,2Pi}]

Naturalmente fare il diagramma cartesiano di una funzione è un esercizio banale che tutti i programmatori hanno affrontato prima o poi, ma la potenza di questo strumento sta nell'essere associato ad un sistema che possiede incorporate quasi tutte le funzioni speciali usate in matematica e fisica.
Una semplice lista di numeri può essere plottata con **ListPlot.**
Una applicazione più complicata che si trova nel package `Graphics`FilledPlot`` permette di riempire lo spazio compreso tra i grafici di due funzioni.
Classiche sono anche la rappresentazione parametrica e polare di funzioni in due dimensioni. L'esempio di Figura 5 mostra il grafico di una spirale mentre Figura 6 presenta un esempio di rodonea ottenuta con la routine **PolarPlot** del package `Graphics`Graphics``.
Usando esplicitamente la Show è possibile combinare diversi grafici, in questo caso la rodonea diventa una margherita.
Si noti l'uso dell'opzione **DisplayFunction->Identity** che permette di non far disegnare il grafico prima della sovrapposizione.

Per quanto riguarda la scelta del colore lo specialista può dettare esplicitamente le coordinate di colore nei sistemi RGB, CMY o YIQ-NTSC (ovviamente se sa usare uno di tali sistemi). Per gli utenti comuni il package `Graphics`Colors`` contiene i nomi inglesi dei colori più usati: nel nostro esempio **Yellow** è il nome simbolico del colore giallo: una variabile predefinita il cui valore è **RGBColor[1,1,0]** (la specifica del giallo nel sistema RGB).

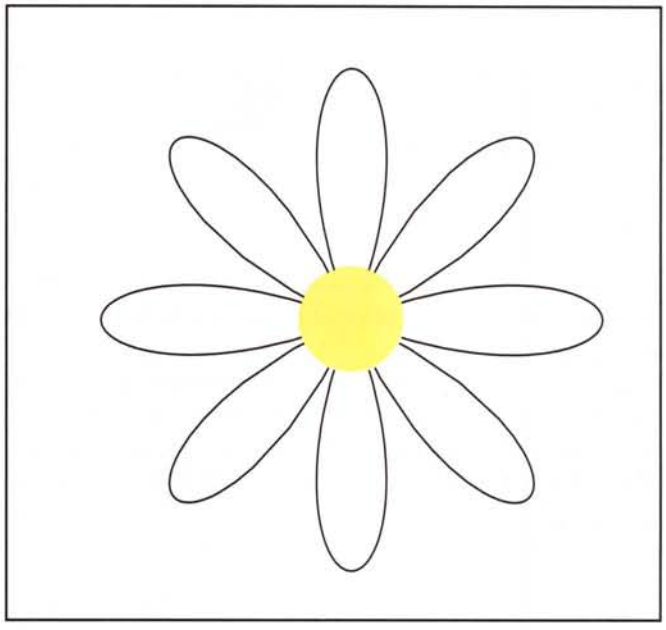

Figura 7: [PolarPlot[Cos[4t],{t,0,2Pi}, DisplayFunction≥Identity], Graphics[Yellow, PointSize[0.2], Point[{0,0}]]], Axes≥False, DisplayFunction≥$DisplayFunction]

Grafici tridimensionali e rappresentazioni di curve tridimensionali

Una funzione di due variabili (ad esempio z = sin(x y) può essere rappresentata in due dimensioni in molteplici modi di-

Rappresentazioni simboliche in Mathematica

Mathematica è un linguaggio interpretato, la azione base dell'interprete consiste nel ricevere una espressione in input e valutarla. Le espressioni hanno la forma di una testa (head) e una sequenza di argomenti. Se la head è definita per quegli argomenti viene applicata la definizione e restituito il risultato, altrimenti l'espressione resta non valutata.

Per esempio:

```
In[1]:=
Plus[3*5,6*7]
Out[1]=
57
In[2]:=
FullForm[Plus[3*5,a]]
Out[2]=
Plus[15,a]
In[3]:=
Pippo[3*5,6*7]
Out[3]=
Pippo[15, 42]
```

Questo permettere di utilizzare funzioni non definite come contenitori per dare la rappresentazione simbolica di qualche oggetto particolare. Per esempio la forma interna del numero razionale 2/3 è Rational[2,3]. La funzione di somma è definita in modo da applicare le regole della somma in modo diverso a seconda del tipo di numeri con cui ha a che fare, se applicata a espressioni con head Rational e argomenti interi effettua il calcolo della somma operando sul numeratore ed il denominatore nel modo usuale:

```
In[4]:=
Plus[Rational[2,3], Rational[1,3]]
Out[4]=
1
```

In modo analogo Show opera su espressioni con head Graphics (che a loro volta contengono primitive grafiche) generando gli opportuni grafici.

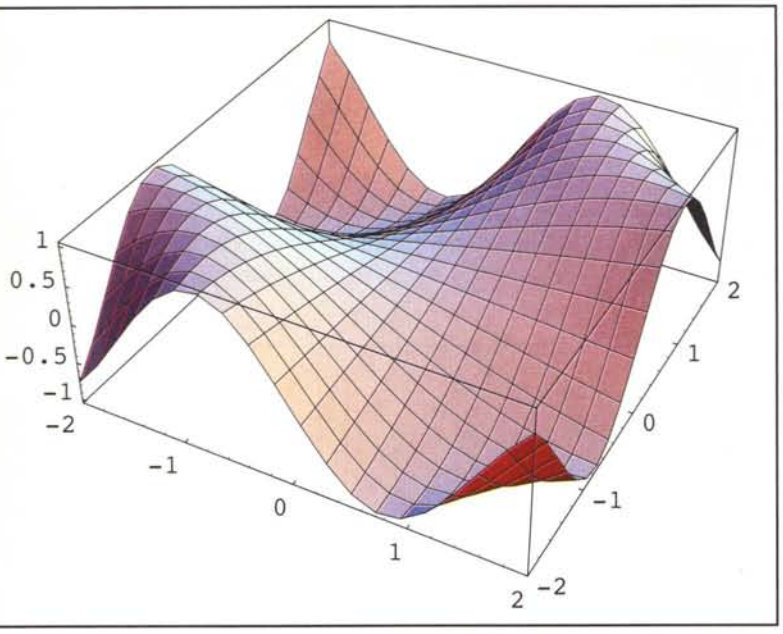

Figura 8: Plot3D[Sin[x y], {x,-2,2}, PlotPoints≥20]

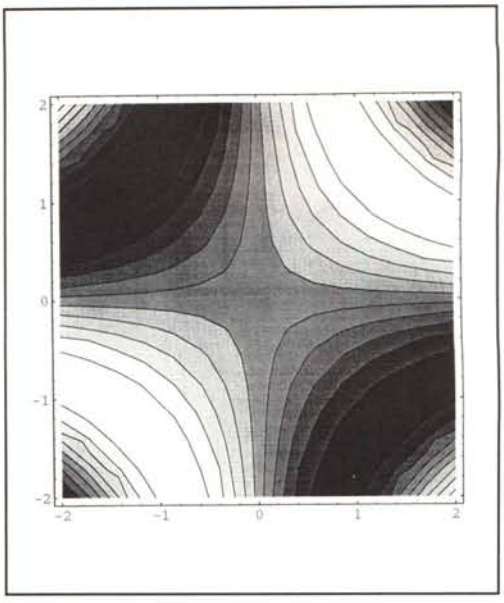

Figura 9: ContourPlot[Sin[x y], {x,-2,2},{y,-2,2}]

versi, ognuno dei quali evidenzia alcune informazioni, nascondendone altre.

Plot3D genera un oggetto di tipo **SurfaceGraphics** che può essere rappresentato in bianco e nero o a colori con le ombreggiature e il colore che rappresentano a scelta, o il valore di una ulteriore variabile, o una illuminazione simulata.

Sia il punto di vista prospettico che il punto di illuminazione possono essere variati.

ContourPlot mostra una mappa a curve di livello, le aree tra le curve possono venire colorate con livelli di grigio o di colore differenti, vedi Figura 9. Per generare un grafico di questo tipo il programma usa un approccio adattivo per agganciare e seguire le curve di livello. Un approccio meno raffinato, che peraltro permette di ottenere risultati altrettanto interessanti è quello di campionare la funzione nei nodi di un reticolo e colorare di conseguanza i vari quadratini. DensityPlot realizza questo tipo di grafico. I risultati sono tanto più accurati quanto maggiore è il numero dei punti usati ma il tempo di elaborazione e la memoria necessaria crescono inevitabilmente come il quadrato del numero di punti di un lato del reticolo. Nella Figura 10 si vede un esempio con 50 punti per lato, abbastanza accurato per apprezzare l'andamento della funzione, anche se è ancora evidente la struttura dei quadratini.

Un altro interessante modo per visualizzare nello spazio una curva o una superficie è quello della rappresentazione parametrica. **ParametricPlot3D** riceve una lista di tre funzioni che rappresentano le tre coordinate spaziali. Se le tre funzioni sono in una sola variabile viene tracciata una curva, se le funzioni sono di due variabili, viene tracciata una superficie. Ad esempio il seguente sistema di equazioni parametriche

$$\begin{cases} x = \cos t \\ y = s\,\mathrm{int} \\ z = t/10 \end{cases}$$

Con una piccola modifica si definisce un cilindro

Dove trovare di più

Per ovvi problemi di Copyright non è possibile ripubblicare i migliori esempi di grafici ottenuti con Mathematica e presentati in letteratura. In genere tutte le pubblicazioni che trattano di Mathematica sono piene di ottimi grafici. In particolare si può vedere:

S. Wolfram. Mathematica. A System for Doing Mathematics by Computer. Addison Wesley,1991 (II Edition).
Oltre alla trattazione completa delle primitive grafiche il libro contiene una galleria di disegni a colori stampati in alta risoluzione.
Mathematica Technical Report: Guide to Standard Mathematica Packages.
Compreso nella confezione del linguaggio, contiene la descrizione dei pacchetti applicativi standard che vengono forniti a corredo. È presente la descrizione, con esempi di funzionamento, di numerose funzioni grafiche aggiuntive.

T.W. Gray and J. Glynn. Exploring Mathematics with Mathematica. Addison Wesley,1991.
Il libro contiene centinaia di grafici in bianco e nero e a colori (con graziose animazioni). I grafici e i programmi che li hanno generati sono disponibili anche sul CD-ROM allegato al libro.

D. Vvedensky. Partial Differential Equations with Mathematica. Addison Wesley,1991.
È un corso universitario sulle equazioni differenziali a derivate parziali, Mathematica è stata usata estesamente per illustrare tecniche e soluzioni e presentare esempi, specialmente per i grafici delle funzioni.

R. Maeder. Programming in Mathematica. Addison Wesley,1991 (II Edition).
Molti capitoli del libro sono dedicati alla scrittura del package di grafica per le funzioni complesse ComplexMap di cui trattiamo nel testo. Tutti i programmi definiti in questo libro sono inclusi nella distribuzione standard di Mathematica.

The Mathematica Journal. Rivista trimestrale edita dalla Miller Freeman Inc.
In ogni numero viene presentata una galleria di disegni a colori e nel dischetto allegato sono inclusi i programmi di generazione.

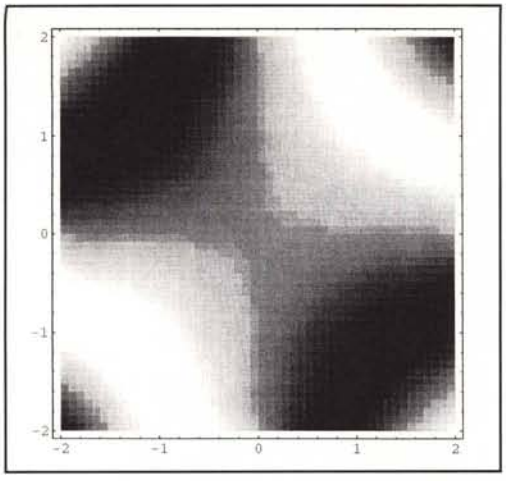

Figura 10: DensityPlot[Sin[x y],{x,-2,2},{y,-2,2},PlotPoints≥50,Mesh≥False]

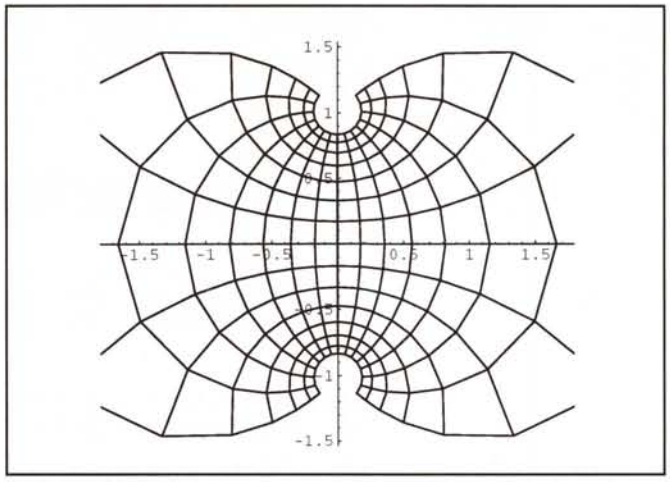

Figura13: CartesianMap [Tan,{-1.2,1.2},{-1.2,1.2}]

$$\begin{cases} x = \cos t \\ y = s\text{int} \\ z = u \end{cases}$$

e facendo variare t da 0 a 3/2 π (tre quarti di giro) si ottiene la figura di un "tubo tagliato"

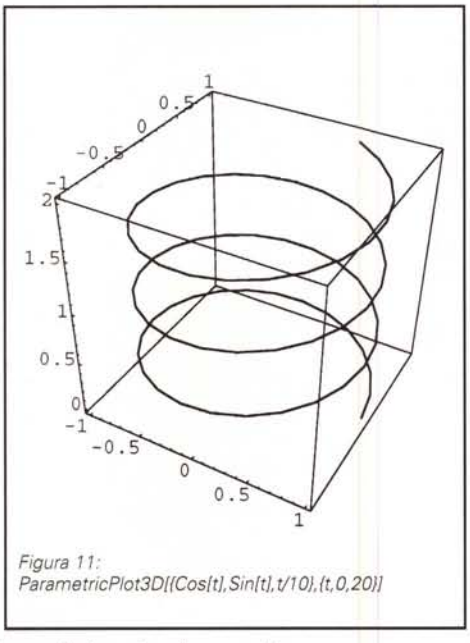

Figura 11: ParametricPlot3D[{Cos[t],Sin[t],t/10},{t,0,20}]

- si considera f(z) come una trasformazione del piano in se stesso (ovvero una funzione che associa ad ogni punto del piano complesso un nuovo punto del medesimo) e si disegna una griglia di punti a maglia quadrata dopo la trasformazione attraverso la f(z).Quest'ultimo tipo di grafico è stato implementato da Roman Maeder ed è contenuto nel Package `Graphics` ComplexMap`. Nelle due ultime figure, ottenute con la routine **CartesianMap**, si vede la griglia trasformata con f(z) = Tan(z) (vedi Figura 13) e con la più famosa delle funzioni di variabile complessa, la Zeta di Riemann:

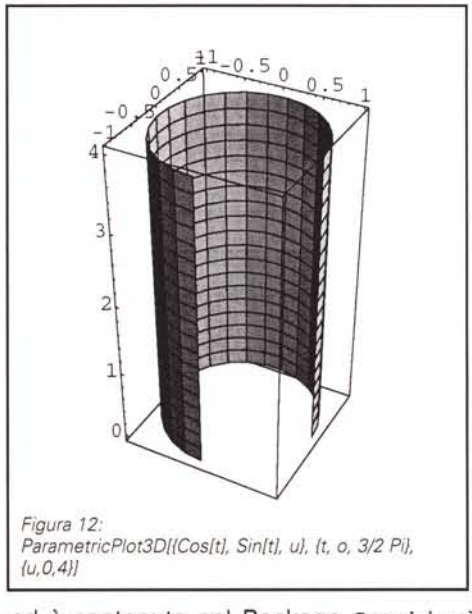

Figura 12: ParametricPlot3D[{Cos[t], Sin[t], u}, {t, o, 3/2 Pi}, {u,0,4}]

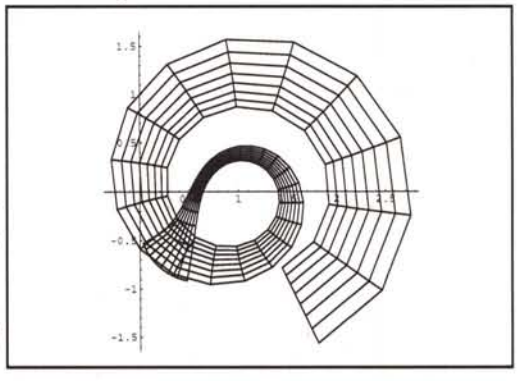

Figura 14: CartesianMap [Zeta, {0.2,0.9,0.1}, {1,19,0.5}]

$$\zeta(z) = \sum_{n=1}^{\infty} n^{-z}$$

MC

Francesco Romani è raggiungibile tramite Internet all'indirizzo romani@di.unipi.it

Rappresentazione di funzioni complesse

Una funzione complessa di variabile complessa (ad esempio $f(z) = x^2 - \mathbf{i}\, y$, con $z=x + \mathbf{i}\, y$) associa coppie di valori reali (la parte reale e la parte immaginaria del risultato f(z)) a coppie di valori reali (la parte reale e la parte immaginaria di z). Una rappresentazione grafica di una tale funzione pone non pochi problemi e le soluzioni, tutte solo parzialmente soddisfacenti sono molteplici:

-plottare il valore assoluto di f(z) in funzione di x e y. Si ottiene un grafico come quello di figura 8 ma si perde ogni informazione sull'argomento (la fase) di f(z).

-plottare il valore assoluto di f(z) in funzione di x e y, associando un colore che dipende dall'argomento di f(z). Questo si può fare facilmente in *Mathematica* e un bell'esempio basato sulla funzione Gamma, costituisce, l'illustrazione di copertina della prima edizione del manuale di Wolfram.

- plottare separatamente la parte reale e la parte immaginaria di f(z) in funzione di x e y. Si ottengono due grafici invece di uno.