

# Ciambelle, nodi e fiorellini

Un titolo più serio sarebbe stato *Rendering Tridimensionale con Mathematica*, ma poi non me lo leggeva nessuno! L'idea base è far vedere una galleria di grafici un po' strani che rappresentano in modi diversi oggetti tridimensionali

Francesco Romani

## Ciambelle

Il nome matematico della ciambella è "toro" (in inglese torus). Nel pacchetto `Graphics`Shapes`` sono presenti sotto forma di intrecci di poligoni le principali forme tridimensionali, tra cui appunto il toro:

```
In[1]:=
<<Graphics`Shapes`
```

```
In[2]:=
?Cylinder
```

Cylinder[r:1, h:1, n:20r] is a list of n polygons approximating an open cylinder centered around the z-axis with radius r and half height h.

```
In[3]:=
?Cone
```

Cone[r:1, h:1, (n:20r)] is a list of n polygons approximating a cone centered around the z-axis with radius r and extending from -h to h.

```
In[4]:=
?Torus
```

Torus[r1:1, r2:0.5, n:20r1, m:20r2] is a list of n\*m polygons approximating a torus centered around the z-axis with radii r1 and r2.

```
In[5]:=
?Sphere
```

Sphere[r:1, n:20r, m:15r] is a list of n\*(m-2)+2 polygons approximating a sphere with radius r.

```
In[6]:=
?MoebiusStrip
```

MoebiusStrip[r1:1, r2:0.5, (n:20r1)] is a list of 2n polygons approximating a moebius strip centered around the z-axis with radii r1 and r2.

```
In[7]:=
?Helix
```

Helix[r:1, h:0.5, m:2, n:20r] is a list of

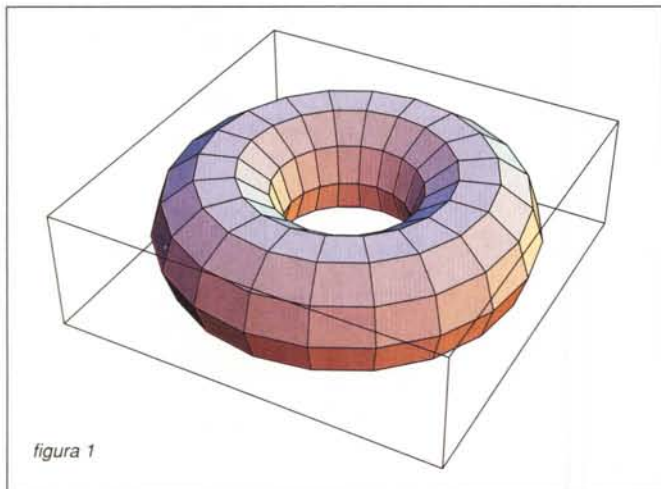


figura 1

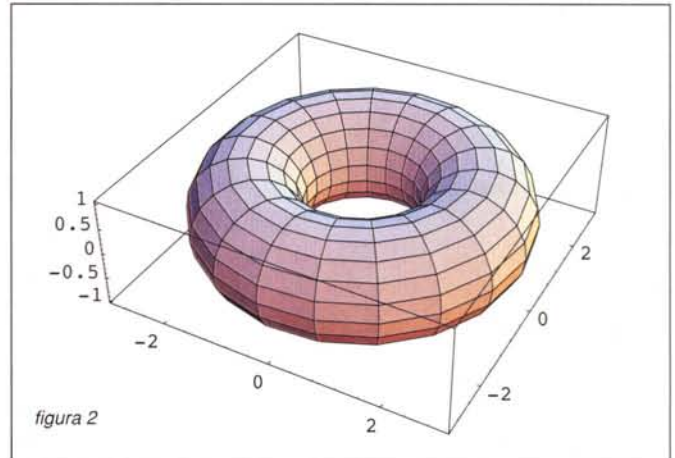


figura 2

n\*m polygons approximating a helix with half height h and m turns.

```
In[8]:=
```

```
?DoubleHelix
```

DoubleHelix[r:1, h:0.5, m:2, n:20r] is a list of n\*m polygons approximating a double helix with half height h and m turns.

Vediamo il toro (rappresentato in figura 1):

```
In[9]:=
```

```
Show[Graphics3D[Torus[]];
```

Un altro modo di generare una ciambella è prendere la superficie ottenuta ruotando una circonferenza intorno ad un asse (come in figura 2). Il pacchetto `Graphics`SurfaceOfRevolution`` permette di giocare con le superfici di rotazione.

```
In[10]:=
```

```
<<Graphics`SurfaceOfRevolution`
```

```
In[11]:=
```

```
SurfaceOfRevolution[{Cos[x]+2, Sin[x]},
{x, 0, 2 Pi}];
```

Un terzo modo per generare una mezza ciambella (un contenitore da budino capovolto) è quello di definire una funzione di x,y che rappresenta l'altezza di un mezzo toro appoggiato su un piano.

```
In[12]:=
```

```
budino=Compile[{x,y},
Sqrt[Max[0, 1-(Sqrt[x^2+y^2]-3)^2]]]
```

```
Out[12]=
```

```
CompiledFunction[{x,y},
Sqrt[Max[0, 1-(Sqrt[x^2+y^2]-3)^2]],
-CompiledCode-]
```

Poiché la funzione dovrà essere richiamata molte volte abbiamo preferito compilarla. Adesso budino può essere usata come una qualunque funzione di due variabili:

```
In[13]:=
```

```
budino[3, 1.7]
```

```
Out[13]=
```



0.893939

Vediamone un normale grafico tridimensionale (vedi figura 3).

In[15]:=

```
Plot3D[budino[x,y],{x,-5,5},{y,-5,5},
PlotPoints->30];
```

Un modo completamente diverso di rendere una funzione di due variabili sono i cosiddetti *Random Dot Stereograms*, basati su un effetto ottico che si verifica quando si tenta di mettere a fuoco una superficie su cui compare una trama regolare. I fortunati dotati di una buona visione stereoscopica avranno notato come talvolta, fissando un pavimento di mattonelle o una qualche griglia, sembri di vedere una superficie più distante, come attraverso uno strato di acqua. È possibile sviluppare una teoria matematica di questo effetto ottico e creare immagini per punti che fissate con una certa attenzione rivelano un disegno tridimensionale nascosto. Il Pacchetto **RDS.m** (presentato nel numero dell'Inverno 1991 del Journal of Mathematica) implementa la funzione **RDSPlot**. L'uso è lo stesso di **Plot3D** e viene prodotta un'immagine a punti che (a spese di un po' di mal di testa) diviene un grazioso (?) budino tremolante (vedi figura 4).

In[16]:=

```
<<"RDS.m"
```

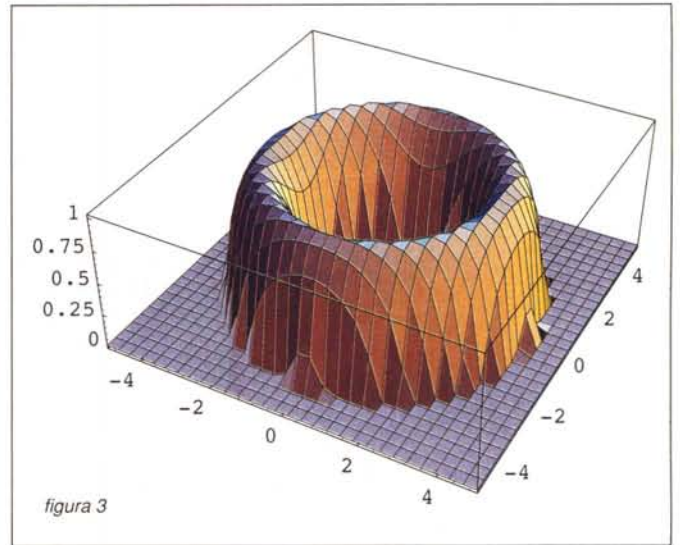


figura 3

In[17]:=

```
Show[RDSPlot[budino[x,y],{x,-5,5},{y,-5,5},
AspectRatio->1,PlotPoints->100]];
```

P.S. Per provare a vederlo bene fissate i due pallini in basso, sfocate la vista (ne dovrete vedere 4) e aggiustate la messa a fuoco per vederne 3, ora alzate lo sguardo senza cambiare

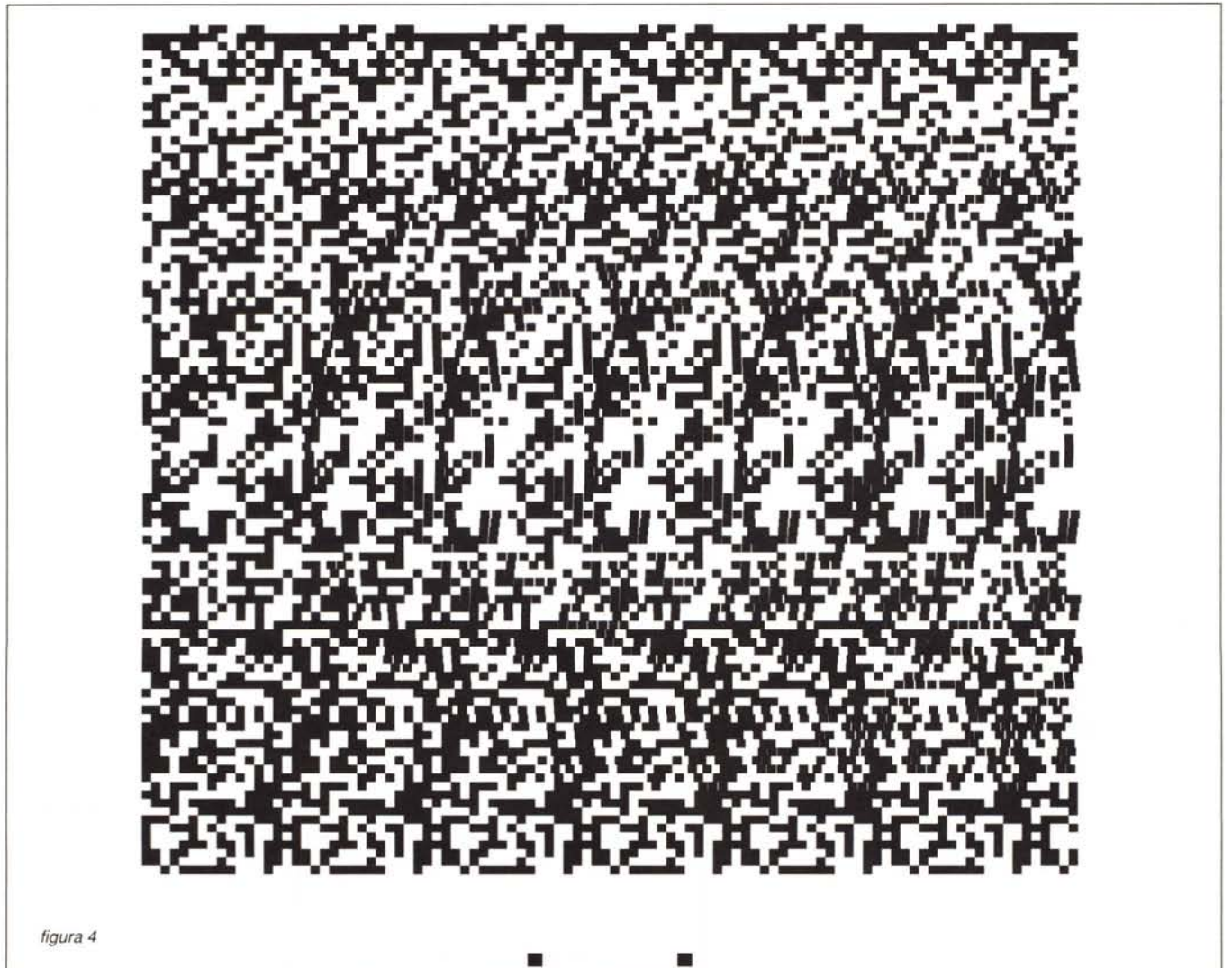
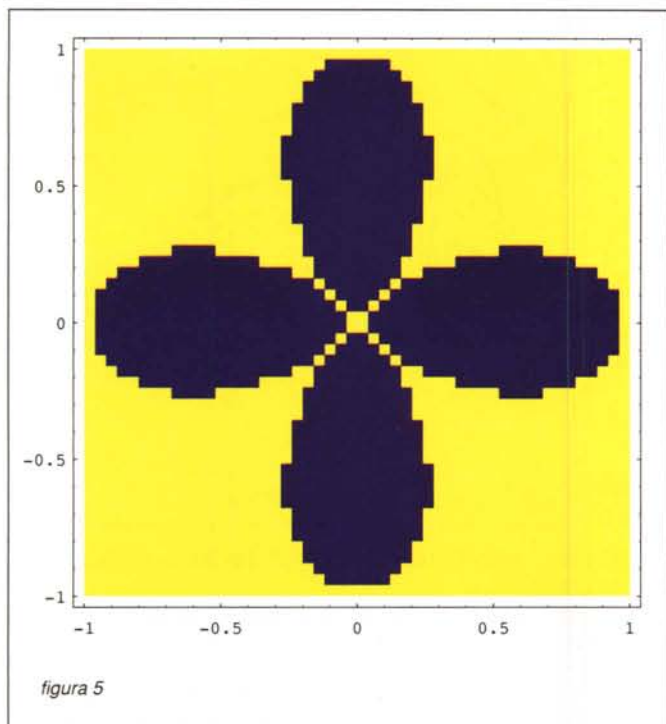


figura 4



messa a fuoco (è questa la parte difficile) e oplà. Se non vi riesce fate provare degli amici finché non vi convincete che non vi ho imbrogliato.  
 P.P.S. Non è un pesce d'Aprile ritardatario!

**Fiorellini**

Vi ricordare la margherita pubblicata su un vecchio numero ? Costruiamo una funzione di due variabili che vale 1 dentro i petali e 0 fuori.

```
In[1]:=
f=Compile[{x,y},
  If[Sqrt[x^2+y^2]<
    Abs[Cos[2 ArcTan[x/y]]],
    1,0]];

```

Il disegno ottenuto con **DensityPlot** (figura 5) è proprio brutto:

```
In[2]:=
DensityPlot[f[x,y],{x,-1,1},
  {y,-1,1},PlotPoints->50,
  Mesh->False]

```

Ma lo stereogramma di figura 6, generato da **RDSPlot**, dà proprio l'idea di un fiorellino ritagliato nella carta e sospeso per aria.



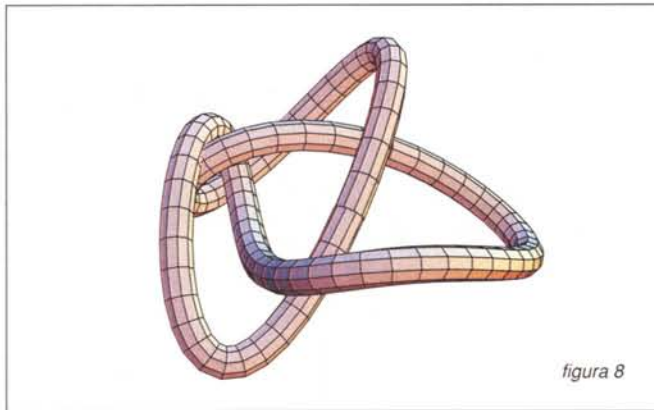


```
In[3]:=
Show[RDSPlot[f[N[x],N[y]],{x,-1,1},{y,-1,1},
AspectRatio->1,PlotPoints->100]
```

**Nodi**

Una linea nello spazio si può rappresentare con tre funzioni che danno le tre coordinate secondo un parametro. Se le tre funzioni hanno lo stesso periodo, la linea si chiude su se stessa e rappresenta un nodo. Oggetti di questo tipo possono essere studiati, classificandoli a meno di trasformazioni che ne cambiano la forma. In parole povere se fate un nodo qualunque con una corda di plastica e poi ne saldate le estremità, qualunque manovra facciate con la corda senza tagliarla, vi ritrovate in mano sempre lo stesso nodo.

Per via matematica si scrivono facilmente tre funzioni di periodo  $2\pi$  e il nodo si disegna subito con la funzione **ParametricPlot3D**.



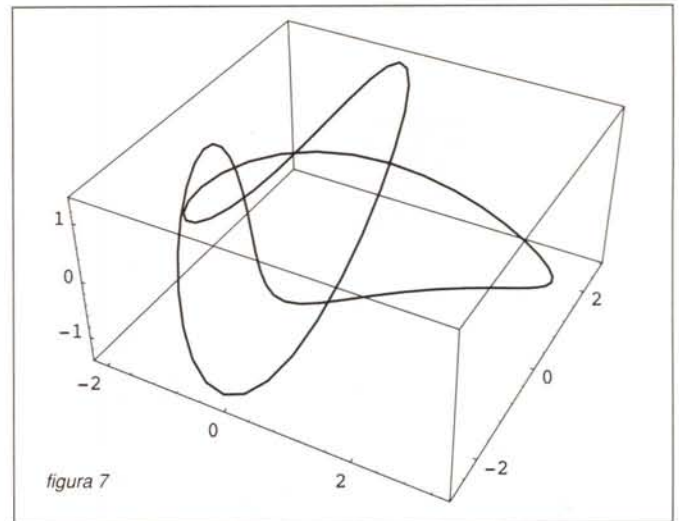
```
In[1]:=
<<Graphics`ParametricPlot3D`
ParametricPlot3D[
{4 Cos[t + N[Pi]] / 3 + 2 Cos[2 t],
4 Sin[t] / 3 + 2 Sin[3 t],
Sin[4 t] + Sin[2 t] / 2},
{t, 0, 2 Pi}];
```

P. Boyland and S. Dickson hanno scritto una funzione che disegna un tubo intorno ad una linea nello spazio rendendo molto più gradevole il risultato dal punto di vista artistico. Il programma si trova nel notebook **Knots.m** nel folder *Sample Notebooks* distribuito insieme a *Mathematica*. Una volta eseguita la cella di inizializzazione si può usare la funzione **tube**.

```
In[2]:=
?tube
tube[f, {t, tmin, tmax, tnum}, circumnum, radius,
options] generates a tube with given radius
about the curve defined by f[t]. The mesh of
the tube has circumnum vertices in the meridian
```

**Bibliografia**

S. Wolfram, *Mathematica. A System for Doing Mathematics by Computer*, II Edition. (Addison Wesley, 1991).  
 Dror Bar-Nathan, *Random Dots Stereograms*, The Mathematica Journal. Voll. 1, n. 3, Inverno 1991.  
 F. Romani, *Introduzione alla grafica*. MCmicrocomputer, n. 128, aprile 1993.



direction and tnum vertices in the longitudinal direction.

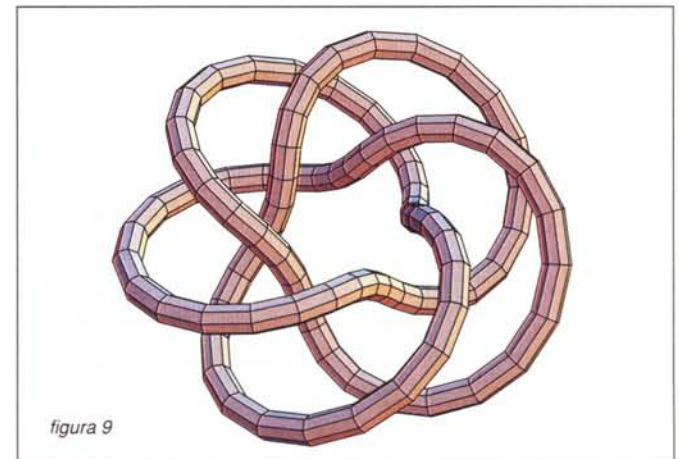
Il nodo della figura precedente acquista tutto un altro aspetto.

```
In[3]:=
Show[tube[
{4 Cos[t + N[Pi]] / 3 + 2 Cos[2 t],
4 Sin[t] / 3 + 2 Sin[3 t],
Sin[4 t] + Sin[2 t] / 2},
{t, 0, 2 Pi, 90}, 9, .25],
Boxed->False];
```

Nel notebook **Knots.m** c'è anche una funzione che genera una famiglia di nodi sempre più complicati al crescere di due parametri p e q.

```
In[4]:=
pk[p_, q_] := tube[
{Cos[t] (1 + .5 Cos[(q/p) t]),
Sin[t] (1 + .5 Cos[(q/p) t]),
.5 Sin[(q/p) t]},
{t, 0, 2 Pi p, 90}, 8, .1];
```

Finiamo la rassegna con un nodo molto vistoso.



```
In[5]:=
Show[pk[3,5],
ViewPoint->{1.600, 0.235, 2.972},
Boxed -> False];
```



# MC *microcomputer* MONOGRAFIE

Le Monografie di MC, una formula studiata esplicitamente per venire incontro in modo pratico ed economico alle esigenze di coloro che desiderano vedere trattati in maniera più estesa ed approfondita alcuni degli argomenti che mensilmente compaiono sulle pagine di MCmicrocomputer.

Nelle Monografie periodicamente verranno affrontati i temi di importanza più rilevante del panorama dell'informatica amatoriale e professionale, con il necessario approfondimento e l'ampio respiro che sulle pagine della rivista non si possono avere.

Quando possibile, a seconda del tema, le Monografie verranno accompagnate da un supporto magnetico contenente materiale di sussidio al testo: una videocassetta o un floppy contenente eventuali listati.

La formula della distribuzione in edicola consente di mantenere elevata la reperibilità delle Monografie mantenendo i prezzi a livelli popolari. In pratica le

Monografie avranno i vantaggi sommati di una rivista e di un libro,

senza gli svantaggi di nessuno dei due.

La prima uscita delle Monografie è dedicata alla OOP e comprende un libro ed una videocassetta. Nel video Phil Khan, fondatore e presidente della Borland, illustra in modo elementare i concetti di base della OOP senza tuttavia entrare nel dettaglio delle tecniche, né delle implementazioni; il libro, scritto da Corrado Giustozzi e Sergio Polini, offre un inquadramento più rigoroso ed approfondito

**Richiedi subito  
le MONOGRAFIE  
di MCmicrocomputer  
mediante l'apposito  
tagliando**



della OOP nel suo contesto applicativo. In più, all'interno di ogni confezione un'offerta promozionale della Borland per l'acquisto dei compilatori OOP Borland a prezzi eccezionali.

Per acquistare le Monografie di MCmicrocomputer utilizzate il modulo d'ordine pubblicato nelle ultime pagine della rivista.