

**Scripts e applicazioni semplici di
guida nella grafica scientifica con**

GNU PLOT

ver. 6.0.1.0 (su Win-11)

claudio magno

revisione
06 giugno 2024

CM_Portable MATH Notebook Series™

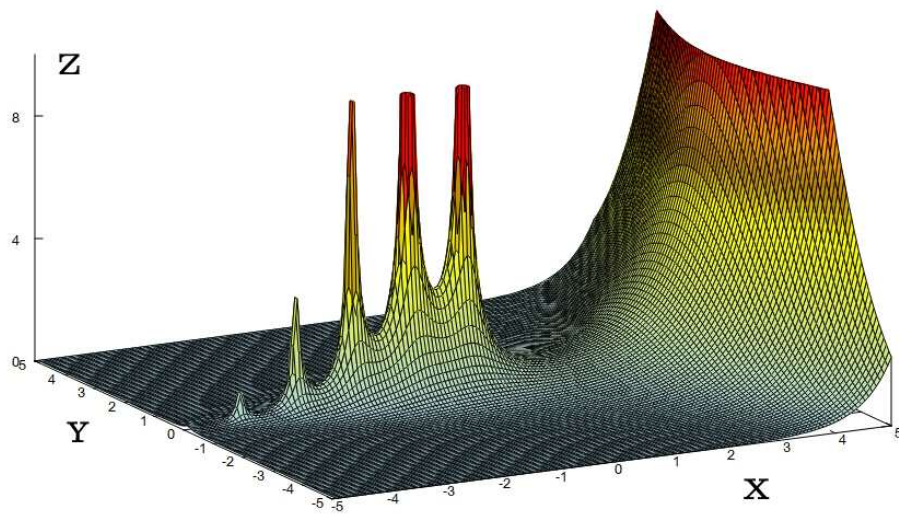


Grafico in \mathbb{R}^3 di $(x; y) \mapsto z = |\Gamma(w)|$, con $w = x + iy$ (v. PP. 26, 97).

INDICE

INTRODUZIONE	P. III
GRAFICA SCIENTIFICA CON GNUPLOT	P. 1
ALTRI ESEMPI E MODELLI MATEMATICI	P. 23
RACCOLTA DI SCRIPTS 2-DIM	P. 32
RACCOLTA DI SCRIPTS 3-DIM	P. 68

INTRODUZIONE

Questo è un notebook buttato giù, a tempo perso, da un utente ‘intermittente’ (si dica pure: *dilettante*) per navigatori curiosi (seri nella loro curiosità ma non necessariamente ‘navigati’) e per altri utenti occasionali. Certo, c’è di molto meglio nel Web ... Però, sapendo che GNUPLOT è in evoluzione continua, vale la pena, di tanto in tanto, aggiornarsi sul suo stato dell’arte e sulla sua migliorata ‘amichevolezza’ (?) ... almeno, in un’ottica da ‘smanettone’.

GNUPLOT è scaricabile da

<http://www.gnuplot.info/> .

Gli *scripts* contenuti in questo notebook richiedono dei Ctrl+C ⇒ Ctrl+V sul proprio dispositivo, dai quali, poi, l’utente possa imparare esercitando la *maggior attenzione ai dettagli possibile*: GNUPLOT fa cose davvero notevoli ma non perdona sbavature di scrittura né sviste sintattiche, neppure minime! È un programma molto potente ma scarno da sembrare un po’ ostico, inizialmente. Come mi pare evidente, richiede attenzione e pratica, insieme con un minimo di informazione e di *operatività* in Analisi Matematica, nel senso anglosassone di ‘Advanced Calculus’ ^(†). Pertanto, non è da aspettarsi che un approccio *en-passant* con GNUPLOT susciti ‘colpi di fulmine’ travolgenti, soprattutto, mi si perdoni, in chi reagisce d’istinto al dettaglio fine – non di rado, *inesorabilmente decisivo!* – come se fosse un ‘sasso nella scarpa’: bisogna entrare *pazientemente* nella logica del programma ...

Commenti e spiegazioni-flash compaiono qui e là negli *scripts* ‘pronto-uso’ presentati; però, questo non basta: l’utente dovrebbe ricorrere, di norma, alla *Guida*, quella in linea, sulla barra degli strumenti, o da scaricare, e.g., da:

<http://www.gnuplot.info/documentation.html> ,

per chiarirsi la sintassi corretta e i comandi meno ovvi o dubbî. Inoltre, lo studio di *Gnuplot in Action* (scaricabile anche dalla mia PAGINA_BIBLIOTECA: https://www.cm-physmath.net/libr_page.html), come di altri ‘tutorials’ reperibili nel Web, dovrebbe rafforzare progressivamente la confidenza con il programma e un suo uso più consapevole ed efficace. Al solito, *skill to do comes from doing!* Anch’io sto sempre ‘ripassando’ il poco già imparato ...

Gli *scripts*, che ho costruito originariamente come files .TXT (in font ARIAL UNICODE MS), sono stati da me ritestati senza problemi con GNUPLOT 6.0.1.0 su WINDOWS™ 11, x64, ver. 22H2. Una volta copiati e incollati su un *file-base* .TXT, possono essere modificati e testati a loro volta: è un esercizio istruttivo, giusto per incominciare a ‘prendere la mano’. I comandi *dichiarativi*, i.e., quelli di gestione degli *scripts*, sono stati ridotti e standardizzati in sequenze *intenzionalmente rigide* e noiosamente *ripetitive*; cripticità avanzate, ‘salti da canguro’ o furbizie criptiche da ‘veri intenditori’ sono escluse. Ciò dovrebbe facilitare la confidenza in chi inizia e l’*esportabilità* degli *scripts* stessi. Lo scopo dell’impresa sarà, forse, modesto ma, penso, non del tutto disprezzabile: quello di ottenere un grafico decente 2- o 3-dim da uno script pensato e costruito in modo elementare ma *analiticamente corretto*. Il resto può aspettare ...

I grafici generati sono stati *copiati* direttamente nella *clipboard* della finestra di lavoro di PAINT.NET, un programma *free* per il ritocco delle immagini e il disegno. Lì, se necessario, i grafici hanno ricevuto nuova centratura, nitidezza e correzioni cromatiche, ridimensionamento, etc. .

PAINT.NET richiede l’*installazione preliminare* in WINDOWS dell’ambiente (*gratuito*) nativo ed evoluto di lavoro di WINDOWS, i.e., di MICROSOFT.NET FRAMEWORK. La versione più appropriata al proprio sistema è scaricabile da:

<https://dotnet.microsoft.com/en-us/download/dotnet-framework/net48> .

Poi, le immagini grafiche – *copiate e incollate* ‘on the fly’ nel documento .DOCX definitivo di MS WORD 365® – sono state *centrate* in una tabella-frame [1×1] (casella singola), con le opzioni seguenti: *margini nulli, bordi cancellati e adattamento automatico al contenuto*.

Buona ‘smanettata’ ma, soprattutto, buona grafica! (sempre in attesa della prossima versione stabile)

C M

mail me: cldmgn47@gmail.com

(†) Come testi non solo autorevoli di riferimento teorico ma anche attenti alle applicazioni, si possono consultare, e.g.:

- PAGANI, C. D. - SALSA, S., *Analisi Matematica*, VOL. 2, ZANICHELLI (-MASSON) (RIST. 1998);
- APOSTOL, T. M., *Calculus*, VOL. 2, 2ND ED., JOHN WILEY & SONS (1969);
- HILDEBRAND, F. B., *Advanced Calculus for Applications*, 2ND ED., PRENTICE-HALL, INC. (1976).

Riferimenti da cui ricavare equazioni in \mathbb{R}^2 e in \mathbb{R}^3 per la pratica nell’implementazione grafica, sono, e.g.:

- <https://www.mathcurve.com/> ;
- STOKA, M. - PIPITONE, V., *Esercizi e Problemi di Geometria*, CAP. 8, P. 773-801, CEDAM (1991).

Grafica scientifica con GNUPLOT

GNUPLOT è un programma interattivo per il tracciamento di grafici di funzioni, basato su comandi da tastiera interpretati istantaneamente. Questo notebook intende fornire alcune conoscenze di base per un ‘pronto-uso’ di GNUPLOT mediante esempi. Se ne può approfondire la conoscenza digitando il comando **h** o premendo il tasto **Help** (sulla barra delle funzioni della *finestra di editing*) o consultando lo *User Manual (PDF)*, scaricabile da: <http://www.gnuplot.info/>.

1. Avvio

Il programma si apre sulla *finestra di editing*, nell’ambiente interattivo di GNUPLOT caratterizzato dal *prompt* di inizio-riga

```
gnuplot>
```

In generale, per i comandi, vanno utilizzati i caratteri *minuscoli* della tastiera; pertanto, si controlli *sempre* che il tasto [BLOC MAIUSC] (o [CAPS LOCK]) sia *disattivato*.

2. Configurazione delle finestre di editing e di grafica e altre opzioni importanti

Sia la *finestra di editing* che la *finestra terminale grafica* (questa seconda si apre all’invio dei comandi **plot** e **splot**, che creano e visualizzano i grafici) presentano, in alto a sinistra, un’icona, cliccando la quale, si apre un menù che contiene il comando *Options*, con cui è possibile copiare il contenuto della finestra grafica nella *memoria temporanea (clipboard)* di Windows, scegliere la visualizzazione degli ambienti di editing e grafico e stampare il contenuto della finestra grafica.

Per verificare il tipo di terminale grafico corrente, digitare **show term**. La versione più recente di GNUPLOT (ver. 5.4.3.0) attiva, di default, il terminale grafico virtuale **qt**, una bi-piattaforma per i sistemi operativi Windows® e Linux®. Un adattamento accettabile al sistema Windows è ottenibile estendendo il terminale di default **qt** con la riga di comando iniziale **set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced** in *ogni* script. Ciò consente l’uso del set *esteso* dei caratteri (apici, pedici, etc.), una *scelta* di *dimensioni* appropriate al proprio video – w(idth), h(eight), in pixel – della finestra grafica e la scelta del *font dimensionato* per la stessa.

È prassi sensata quella di preparare una ‘brutta copia’ delle istruzioni, scrivendole su un file .TXT prima del loro *copia-incolla* nella finestra di editing e dell’esecuzione (*Invio* ↵).

Inoltre, è possibile salvare un’intera sessione di GNUPLOT in un file di testo mediante il comando *File_Save...* e, in seguito, rieseguirla richiamandola con il comando *File_Open...*.

3. Richiamo delle righe di istruzione già eseguite

Durante una sessione di GNUPLOT, se si vogliono rieseguire comandi introdotti in una riga di istruzioni precedente, è sufficiente richiamare la riga desiderata con i tasti direzionali [↑], [↓] e, quindi, il tasto *Invio* [↵] quando compare tale riga (così da modificarla prima della sua esecuzione).

4. Grafico 2-dim di una o più funzioni della forma esplicita $y = f(x)$

Si tratta di funzioni non-parametriche, tali che la variabile y è esplicitabile vs. la variabile x e, a ogni valore ammissibile di x , corrisponde *uno e un solo* valore di y . Una codifica possibile è:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafici delle Funzioni Goniometriche" # opzionale
unset parametric
set grid # applicazione della griglia alla finestra grafica (opzionale)
# set time # mostra data e orario di costruzione del grafico (opzionale)
set xlabel "X (radianti)"
set ylabel "Y (valori della funzione)"
set samples 250
```

```

set xrange [-pi:pi]
set yrange [-1.15:1.15] # Le funzioni cot, sec, csec, sech, coth, csech e le loro inverse non sono \
                           incluse nella libreria di GNUPLOT (v. 'Help', Expressions_Functions), quindi, \
                           è necessario assegnarle mediante le definizioni analitiche rispettive

cot(x)=1/tan(x)
set size square 1.0, 1.0 # parametri b, h (base vs. altezza dei lati della finestra grafica). \
                           1.0, 1.0 sono i parametri di scala X vs. Y – assegnabili dall'utente! – per \
                           l'aggiustamento della distorsione dell'immagine causata dallo schermo

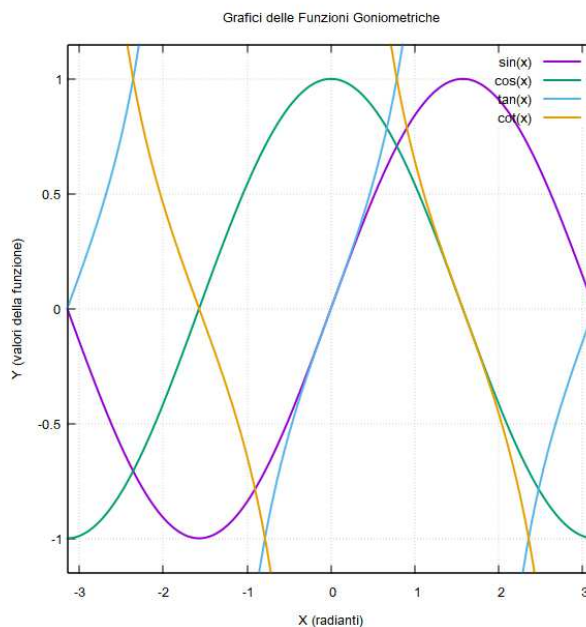
plot sin(x) lw 2, cos(x) lw 2, tan(x) lw 2, cot(x) lw 2 # lw=line width; lo spessore predefinito è 1

```

Si osservi la presenza di due tipi di comandi:

- i comandi **set** e **unset** che permettono, o meno, di assegnare al grafico caratteristiche tali da determinarne il tipo e l'aspetto;
- il comando **plot** che genera l'output del grafico sul terminale in uso.

In effetti, l'unico comando di set indispensabile è **unset parametric**, e soltanto per l'eventualità che, durante la stessa sessione di GNUPLOT, sia stato già assegnato il comando **parametric**.



Gli stessi grafici sono ottenibili con i comandi minimi

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
cot(x)=1/tan(x)
#
plot [-pi:pi] [-1.15:1.15] sin(x), cos(x), tan(x), cot(x)

```

I valori posti tra le parentesi quadre e separati dal segno ‘**!**’ sono, rispettivamente, quello *minimo* e quello *massimo* per x e per y definiti dall'utente (se omessi, GNUPLOT assegna quelli di default).

Il comando **set samples 250** fissa a 250 il numero di punti nei quali viene calcolata la funzione (se omesso, il numero è **100**, di default). Aumentando il numero di punti, aumenta l'accuratezza del grafico della funzione ma può aumentare sensibilmente il tempo necessario per calcolarne i valori; comunque, *l'output dipende anche dalla definizione grafica del terminale-video in uso.*

□

Si può ripetere l'esercizio determinando i grafici delle prime sei *Funzioni di Bessel Ordinarie di rango intero*, $J_0(x)$, $J_1(x)$, ..., $J_5(x)$. Un'idea immediata è quella di sfruttarne la rappresentazione generale in serie *uniformemente convergente* di potenze reali,

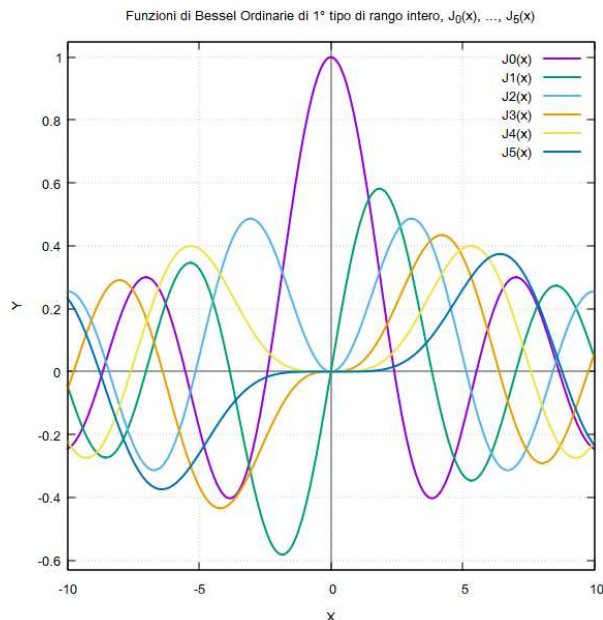
$$x \mapsto J_n(x) = \sum_{k=0}^{+\infty} \frac{(-1)^k}{2^{2k+n} k! (k+n)!} x^{2k+n} \equiv \left(\frac{x}{2}\right)^n \sum_{k=0}^{+\infty} \frac{(-1)^k x^{2k}}{2^{2k} \Gamma(k+1) \Gamma(k+n+1)}.$$

La convergenza uniforme in \mathbb{R} di tale serie consente l'espedito ovvio di prenderne una somma *ridotta sufficiente* per una grafica accettabile (il resto della serie è rapidamente *infinitesimo*) purché il calcolo del denominatore generale non provochi un *overflow* di memoria RAM. I limiti di calcolo di GNUPLOT impongono che sia $2^{2k} k! (k+n)! \leq 170!$. Da una verifica immediata, questo vincolo corrisponde all'insieme discreto $\{k\} = \{0, 1, 2, \dots, 82\}$ di valori interi gestibili dal programma. Se $k \leq 60$, e.g., la somma *ridotta* 60- sima della serie si scrive con la sintassi (cf/c. P. 24-31)

```
sum[k=0:60](-1)**k/(2.**(2*k+n)*k!*(k+n)!)*x**(2*k+n).
```

Con questa, si costruiscono somme ridotte più che sufficienti per tracciare i grafici delle *Funzioni di Bessel* richieste, che appariranno nel dominio comune della stessa finestra:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Funzioni di Bessel Ordinarie di 1° tipo di rango intero, J_0(x), ..., J_5(x)" # opzionale
# in 'title', ogni singolo carattere preceduto dal trattino basso, _ , appare come pedice
set xlabel "X"; set ylabel "Y" # opzioni 'set' (e 'unset') poste sulla stessa riga e separate da ';'
set yzeroaxis lt 1 lw 0.7 lc rgb "#000000"; set xzeroaxis lt 1 lw 0.7 lc rgb "#000000" # definizione degli \
    assi nel sistema di riferimento rettangolare (tipo linea (lt), spessore linea (lw), colore linea (lc))
set grid; set samples 500
J0(x)=sum[k=0:60](-1)**k/((2.**k*k!)**2)*x**(2*k)
J1(x)=sum[k=0:60](-1)**k/(2.**(2*k+1)*k!*(k+1)!)*x**(2*k+1)
J2(x)=sum[k=0:60](-1)**k/(2.**(2*k+2)*k!*(k+2)!)*x**(2*k+2)
J3(x)=sum[k=0:60](-1)**k/(2.**(2*k+3)*k!*(k+3)!)*x**(2*k+3)
J4(x)=sum[k=0:60](-1)**k/(2.**(2*k+4)*k!*(k+4)!)*x**(2*k+4)
J5(x)=sum[k=0:60](-1)**k/(2.**(2*k+5)*k!*(k+5)!)*x**(2*k+5)
set size square 1.0, 1.0 # parametri b, h (v. p. 2)
#
plot [-10.:10.] [-0.63:1.05] J0(x) lw 2, J1(x) lw 2, J2(x) lw 2, J3(x) lw 2, J4(x) lw 2, J5(x) lw 2
```



Dai comandi precedenti, è immediato estrarre quelli relativi, e.g., al solo *graf* (J_3):

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Funzione di Bessel Ordinaria di 1° tipo J_3(x)"
set xlabel "X"; set ylabel "Y"
```

```
set samples 500
set grid
plot [-20.:20.] [-0.63:1.05] sum[k=0:60](-1)**k/(2.**k*(k+3)!)*x**(2*k+3) lw 2
```

Circa i grafici delle J_n , il comando `set samples 500` fissa a 500 il numero di punti nei quali esse vengono calcolate: un eccesso di precisione gestito agevolmente da GNUPLOT.

5. Grafico 2-dim di una curva piana definita implicitamente dalla forma $f(x,y)=0$

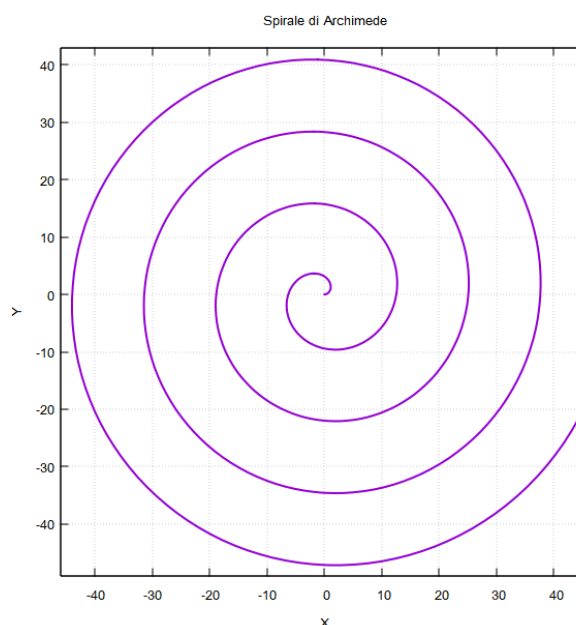
Quando l'equazione $f(x,y)=0$ è tale per cui risulta impossibile esplicitare entrambe le variabili x e y univocamente – poiché, e.g., per lo stesso valore di x , y può assumere più valori, come nel caso dell'equazione generale di una conica – per tracciare il grafico della *curva implicita*, si dovrà ricorrere a una sua rappresentazione parametrica, *razionale* o *polare*.

La forma parametrica associata all'equazione implicita $f(x,y)=0$ è: $x = x(t)$, $y = y(t)$, dove $\{x, y\}$ è la coppia *cartesiana* $\{\langle \text{ascissa} \rangle, \langle \text{ordinata} \rangle\}$ mentre t è il parametro, dal quale, a loro volta, x e y dipendono, e.g., l'angolo piano-polare. Ecco tre esempi:

5.1 La Spirale di Archimede

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set grid # opzionale
set title "Spirale di Archimede"
set xlabel "X"; set ylabel "Y"
unset key # nasconde, da qui in poi, l'espressione matematica di qualsiasi grafico
set samples 500
set parametric
rho(t)=2*t # raggio piano-polare (funzione dell'anomalia t (=phi))
x(t)=rho(t)*cos(t); y(t)=rho(t)*sin(t)
set size square 1.0, 1.0 # parametri b, h (v. p. 2)
#
plot [0.:8*pi] [-46.:46.] [-49.:43.] x(t), y(t) lw 2 # grafico delle variazioni SIMULTANEE di x(t), y(t) vs. t; \
# ranges della coordinata angolare t e delle componenti \
# cartesiane parametriche nel piano X×Y; \
# spessore 2; colore non assegnato (default).
```

`unset parametric`



Il comando `set samples 500` fissa a 500 il numero di campionamenti equidistanti per l'intervallo del parametro t e quindi fissa a 500 il numero di coppie di valori $\{x, y\}$, calcolati e utilizzati per tracciare il grafico.

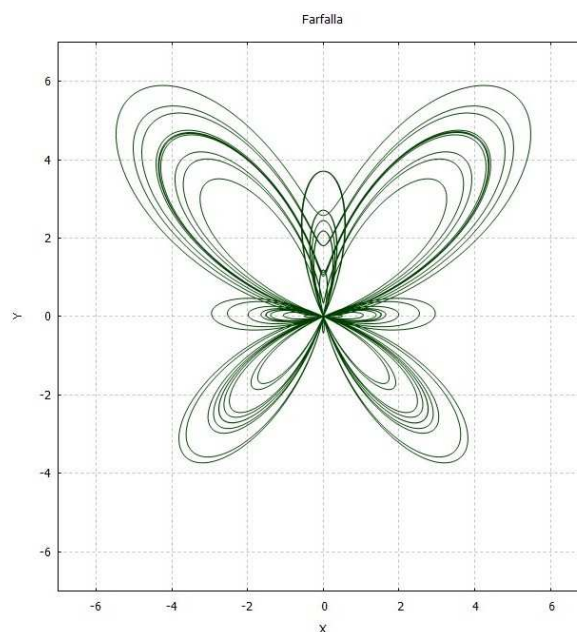
Lo stesso grafico (visibilmente *deformato*, però, in assenza dell'istruzione `set size appropriata` per il terminale video in uso) è ottenibile con i comandi minimi

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set parametric
set samples 500
#
plot [0.:8*pi] [-46.:46.] [-49.:43.] 2*t*cos(t), 2*t*sin(t)
#
unset parametric
```

5.2 La 'Farfalla'

In questo esempio, diversamente da quello precedente, le assegnazioni dei t -domini di $x(t)$ e di $y(t)$ per il tracciamento del grafico appaiono esplicitamente in tre righe di comando distinte:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Farfalla"
set xlabel "X"; set ylabel "Y"
unset key # nasconde l'espressione matematica del grafico
set grid
set parametric
set trange [-10*pi:10*pi]; set xrange [-7.:7.]; set yrange [-7.:7.]
set samples 15000
set size square 1.0, 1.0 # parametri b, h (v. p. 2)
r=3./2 # raggio piano-polare costante in formato REALE, scrivendo 3., NON 3)
#
x(t)=-r*(2*cos(4*t)+(sin(t/12))**5-exp(cos(t)))*sin(t)
y(t)=-r*(2*cos(4*t)+(sin(t/12))**5-exp(cos(t)))*cos(t)
#
plot x(t), y(t) lc rgb "#004000" # grafico delle variazioni simultanee di x(t) e di y(t) vs. t; colore personalizzato
#
unset parametric
```

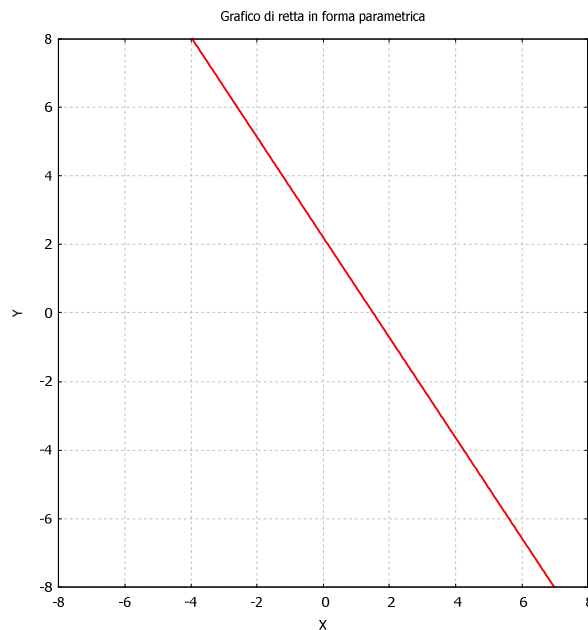


5.3 Grafico di una *retta* in forma parametrica

Questa forma, di origine *vettoriale*, richiede la conoscenza preventiva delle coordinate di *due* punti *distinti* della retta, $P \equiv (a_1; b_1)$ e $Q \equiv (a_2; b_2)$. Se, e.g., $P \equiv (1.23; 4)$ e $Q \equiv (3\sqrt{3}/2; 6)$, allora,

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico di retta in forma parametrica"
unset key; set grid
set xlabel "X"; set ylabel "Y"
set size square 1.0, 1.0      # parametri b, h (v. p. 2)
set samples 1000
x1=-8.; x2=8.                # [x1:x2] è il range-x della finestra grafica, fissato dall'utente
y1=-8.; y2=8.                # [y1:y2] è il range-y della finestra grafica, fissato dall'utente
a1=-1.23; b1=4.               # (a1;b1) è la coppia di coordinate di un punto P qualsiasi della retta, \
                               # scelto dall'utente
a2=-3*sqrt(3)/2; b2=6.        # (a2;b2) è la coppia di coordinate di un punto Q qualsiasi della retta, \
                               # purché sia Q≠P (scelto dall'utente)
t1=-7.; t2=5*pi               # [t1:t2] è il range parametrico. Conviene che sia scelto di ampiezza \
                               # tale da lasciar estendere il grafico della retta oltre i bordi della finestra \
                               # [x1:x2]×[y1:y2]. \
                               # Le dieci costanti x1, x2, ..., t1, t2 devono essere in formato REALE

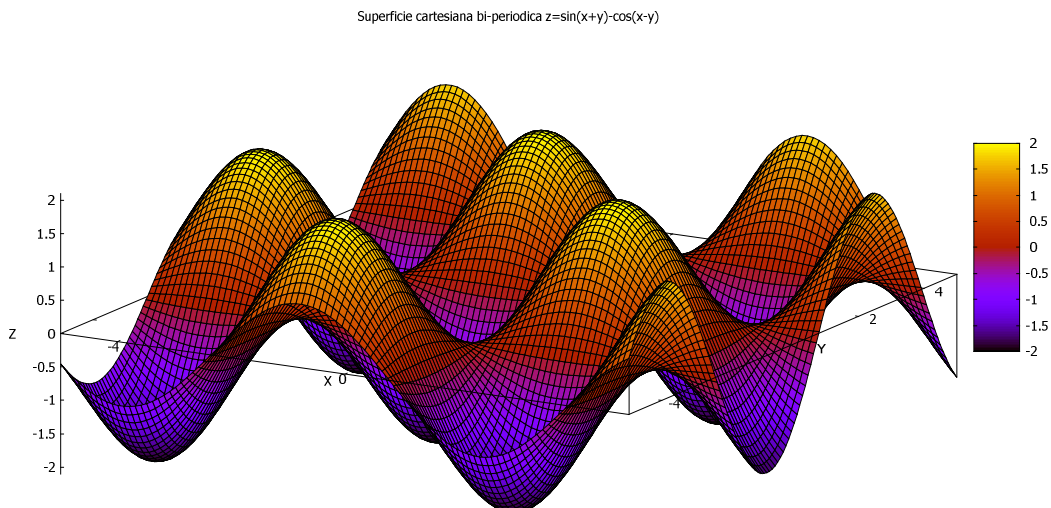
set parametric
x(t)=a1+(a2-a1)*t
y(t)=b1+(b2-b1)*t
#
plot [t1:t2] [x1:x2] [y1:y2] x(t), y(t) lc rgb "#f00000" lw 2
    # lc=line color, lw=line width (assegna colore e spessore personalizzati). \
    # Il colore e lo spessore personalizzati vanno specificati dopo la y-componente cartesiana \
    # parametrica di CIASCUN grafico
m=(b2-b1)/(a2-a1)             # coefficiente angolare (la retta è VERTICALE se a1=a2)
print m                       # calcola il valore di m (m=undefined value\variable (infinito) se a1=a2)
unset parametric
```



6. Grafico 3-dim di una funzione in forma esplicita $z = f(x, y)$

Si tratta di una funzione non-parametrica, i.e., nella quale la variabile z è esplicitata vs. x e y e, per ogni coppia $\{x, y\}$, si ha *un solo* valore di z . Tale funzione rappresenta una *superficie* 3-dim cartesiana ordinaria:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie cartesiana bi-periodica z=sin(x+y)-cos(x-y)" # opzionale
unset key
set xyplane at 0. # z-quota del piano X×Y
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-5.:5.]; set yrange [-5.:5.]; set zrange [-2.1:2.1] # (utente)
set samples 100, 100; set isosamples 120, 120 # densità di campionamento, della griglia
set pm3d depthorder hidden3d # (default)
set view 60, 30, 1, 1
#
z(x,y)=sin(x+y)-cos(x-y)
#
splot z(x,y) w pm3d # 'w'=with. Il comando 'w pm3d' va specificato per OGNI grafico richiesto.
```



Il significato dei vari comandi è facilmente interpretabile. Si osserva la presenza di `set samples 100,100` che fissa a 100 il numero di punti equidistanti nei quali vengono campionati entrambi gli intervalli di definizione delle variabili x e y . Questo significa che la funzione $z = f(x, y)$ viene calcolata per 10000 coppie di valori $\{x, y\}$. Il comando `set isosamples 120, 120` fissa a 120 il numero di linee a $x = \text{costante}$ e a $y = \text{costante}$, rispettivamente, che generano il *reticolo* (o *griglia*) della superficie (di default, tale numero è 10). Il comando che genera un grafico 3-dim, i.e., una superficie, è `splot`; gli intervalli per i valori di x, y, z sono *opzionali*, se si desidera modificare quelli di default. In GNUPLOT, è disponibile il comando composto `set pm3d depthorder hidden3d`, che nasconde quelle parti di superficie che, nella realtà, rimangono *nascoste* all'osservatore ma che visualizza il *reticolo* di sostegno delle linee generatrici.

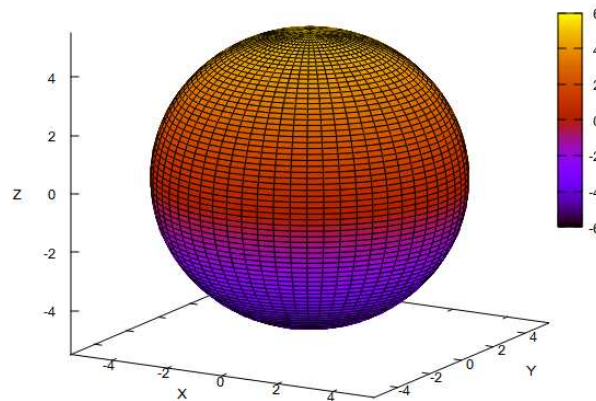
■

7. Grafico 3-dim di una superficie definita implicitamente dalla forma $f(x,y,z)=0$

Il grafico 3-dim di una superficie definita *implicitamente* da $f(x, y, z) = 0$ possiede le coordinate *cartesiane* (rettangolari) espresse *parametricamente* da $x = x(u, v)$, $y = y(u, v)$, $z = z(u, v)$. La superficie sferica di raggio $r = 5$ può essere rappresentata, e.g., con lo *script*

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie sferica di raggio r = 5 in coordinate sferiche" # opzionale
set xyplane 0.; set origin 0.23, 0. # z-quota del piano X×Y; coordinate (x;y) di inizio del grafico
unset key # nasconde l'espressione matematica del grafico
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 80, 80; set isosamples 60, 60
set parametric
set urange [0.:pi] # dominio principale dell'angolo polare
set vrange [0.:2*pi] # dominio principale dell'angolo azimutale
r=5. # i.e., x(u,v)=y(u,v)=z(u,v); raggio r costante, in formato REALE)
set xrange [-5.5:5.5]; set yrange [-5.5:5.5]; set zrange [-5.5:5.5]
set view 75, 30, 1, 1
set pm3d depthorder hidden3d
set size square 0.57, 1. # parametri b, h (v. p. 2)
#
splot r*sin(u)*cos(v), r*sin(u)*sin(v), r*cos(u) w pm3d # i.e., x(u,v), y(u,v), z(u,v)
#
unset parametric
```

Superficie sferica di raggio r = 5 in coordinate sferiche



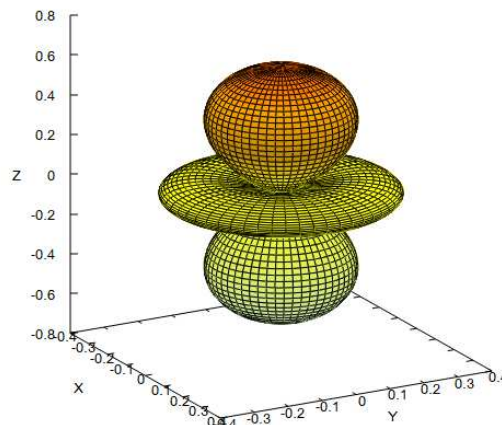
È importante osservare come, qui, i parametri siano *due*, u e v , e come il comando `set isosamples 60, 60` definisca il *reticolo* sulla superficie con 60 linee a $u = \text{costante}$ e 60 linee a $v = \text{costante}$. In particolare, se $x(u, v) \equiv y(u, v) \equiv z(u, v)$ definiscono una superficie a simmetria *sferica* vs. le coordinate parametriche u e v , con $u \equiv \theta \in [0, \pi]$ e $v \equiv \varphi \in [0, 2\pi]$, il *vettore-posizione* (vs. l'origine) *rappresentativo* \mathbf{r} corrispondente è esprimibile – *sempre* – attraverso le sue *proiezioni* (ortogonali) sugli assi cartesiani X , Y , Z nella forma (scrivendo $w := x \equiv y \equiv z$)

$$\mathbf{r}(u, v) := w(u, v) \sin u \cos v \hat{x} + w(u, v) \sin u \sin v \hat{y} + w(u, v) \cos u \hat{z}. \quad (1)$$

Esempi importanti di funzioni *a simmetria sferica*, sono forniti dalle *Funzioni Armoniche Sferiche reali* $Z_{l,m}(\theta, \varphi)$. Il grafico di $w(u, v) \mapsto Z_{2,0}(\theta, \varphi)$ può essere ottenuto con le istruzioni seguenti:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Armonica Sferica reale quadrupolare orto-normalizzata  $Z_{2,0} = f(u,v)$ "
unset key # nasconde l'espressione matematica del grafico
set xyplane 0.; set origin 0.23, 0.; set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 100, 100; set isosamples 55, 55
set size square 0.51, 1.0 # parametri b, h (v. p. 2)
set view 73, 61, 1., 1.
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
unset colorbox
set parametric
x(u,v)=sin(u)*cos(v) # coseno direttore di  $r$  vs. asse X
y(u,v)=sin(u)*sin(v) # " vs. asse Y
z(u,v)=cos(u) # " vs. asse Z
w(u,v)=sqrt(5./(16*pi))*(2*z(u,v)**2-x(u,v)**2-y(u,v)**2)
splot [0.:pi] [0.:2*pi] w(u,v)*x(u,v), w(u,v)*y(u,v), w(u,v)*z(u,v) w pm3d
# range di u, range di v, componenti cartesiane
unset parametric
```

Armonica Sferica reale quadrupolare orto-normalizzata $Z_{2,0} = f(u,v)$



Come esempio ulteriore, il grafico della superficie corrispondente all'Armonica Sferica reale orto-normalizzata $Z_{4,-3}(\theta, \varphi)$ è ottenibile con lo script

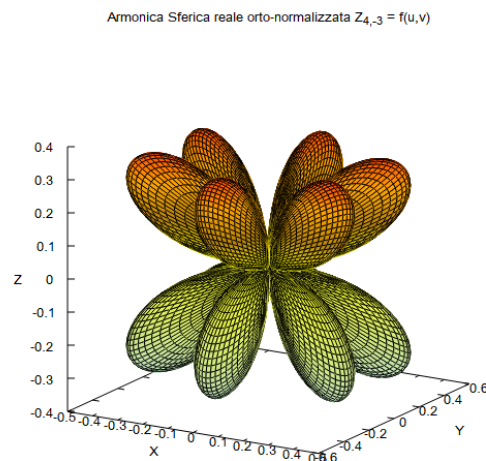
```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Armonica Sferica reale orto-normalizzata  $Z_{4,-3} = f(u,v)$ "
set xyplane 0.; set origin 0.23, 0.
set view 73, 31
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
unset key # nasconde l'espressione matematica del grafico
set parametric
set urange [0.:pi]; set vrange [0.:2*pi]
set samples 120, 120; set isosamples 200, 200
set size square 0.51, 0.9 # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
```

```

unset colorbox          # (opzionale)
x(u,v)=sin(u)*cos(v)    # coseno direttore di  $\mathbf{r}$  vs. asse X
y(u,v)=sin(u)*sin(v)    # " " vs. asse Y
z(u,v)=cos(u)           # " " vs. asse Z
#
f(u,v)=sqrt(315./(32*pi))*sin(u)**3*cos(u)*sin(3*v)
#
splot f(u,v)*x(u,v), f(u,v)*y(u,v), f(u,v)*z(u,v) w pm3d

```

Ne risulta il grafico 3-dim seguente:



Siano $u \equiv \varphi \in [0, 2\pi]$, $v \equiv \rho \in \mathbb{R}^+$ e siano date le funzioni $(u; v) \mapsto g(u, v) \wedge z(u, v)$. Se \mathbf{r} traccia una superficie a simmetria *cilindrica* (e.g., vs. l'asse Z), risulta (cf/c Eq. (1), p. 9)

$$\mathbf{r}(u, v) := g(u, v) \cos u \hat{\mathbf{x}} + g(u, v) \sin u \hat{\mathbf{y}} + z(u, v) \hat{\mathbf{z}}, \quad (2)$$

parametrica in u, v e separata nelle sue componenti cartesiane (se $g(u, v) = \rho_0 \equiv \text{costante}$, allora, \mathbf{r} traccia $\text{graf}(z)$ sulla superficie cilindrico-circolare di raggio ρ_0). Applicazioni interessanti sono

la superficie ‘*Catenoide*’, applicata in modelli gravitazionali:

```

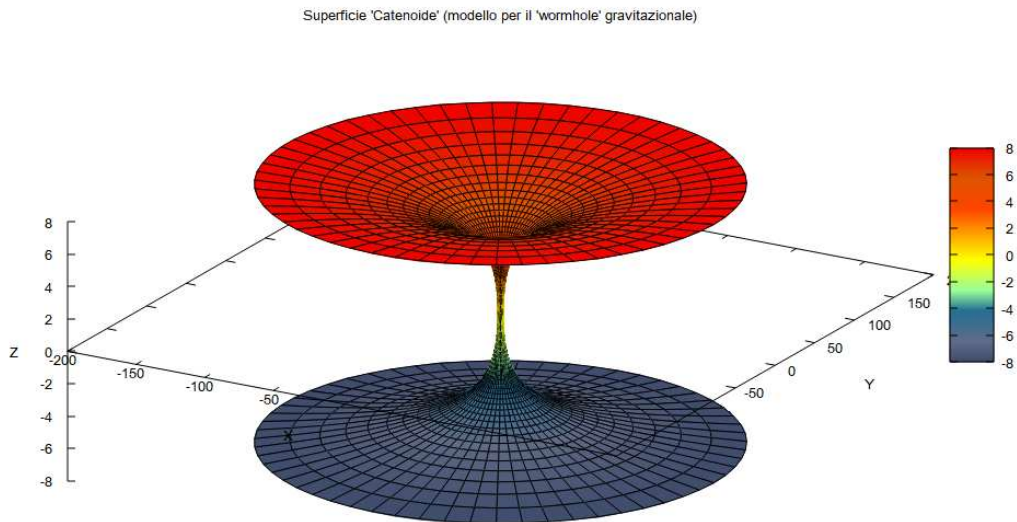
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie 'Catenoide' (modello per il 'wormhole' gravitazionale)"
unset key          # nasconde l'espressione matematica del grafico
set xyplane at 0.;
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 60, 60; set isosamples 70, 70
set size square 1., 1. # parametri b, h (v. p. 2)
set parametric
set urange [0.:2.*pi] # range della coordinata angolare di rotazione (phi) nel piano XxY
set vrange [-8.:8.] # range della coordinata v (= z), la distanza lungo l'asse di simmetria
a=1.5 # a (>0), parametro di deformazione
x(u,v)=a*cosh(v/a)*cos(u) # x-componente di  $\mathbf{r}(u,v)$ 
y(u,v)=a*cosh(v/a)*sin(u) # y-componente di  $\mathbf{r}(u,v)$ 
z(u,v)=v # z-componente di  $\mathbf{r}(u,v)$ 
set pm3d depthorder hidden3d

```

```

set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", 10 "#ffff00", \
                    15 "#ff4500", 18 "#dd5500", 21 "#f00000") # colorbox personalizzata
# unset colorbox # opzionale
set view 51, 30, 1, 1
#
splot x(u,v), y(u,v), z(u,v) w pm3d
#
unset parametric

```

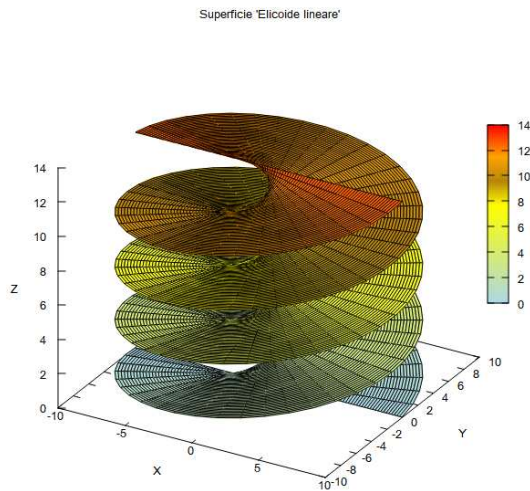


e la superficie 'Elicoide' sia *lineare* (cf/c Superficie di Dini, P. 80) che *iperbolica*:

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie 'Elicoide lineare'"
unset key # nasconde l'espressione matematica del grafico
set xyplane 0.; set origin 0.23, 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set ticslevel 0
set samples 120, 120; set isosamples 100, 100
set size square 0.6, 1. # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# unset colorbox # opzionale
set view 60, 30, 1, 1
set parametric
g(u,v)=u # funzione di avvolgimento (winding function) lineare, il raggio u=rho
x(u,v)=g(u,v)*cos(v) # x-componente di r(u,v)
y(u,v)=g(u,v)*sin(v) # y-componente di r(u,v)
z(u,v)=v # z-componente di r(u,v), l'anomalia v=phi
#
splot [-10.:10.] [0.:4*pi] x(u,v), y(u,v), z(u,v) w pm3d
#
unset parametric

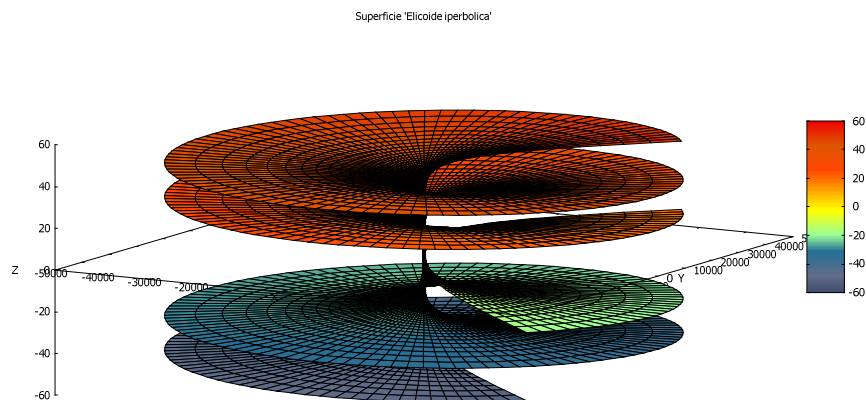
```



```

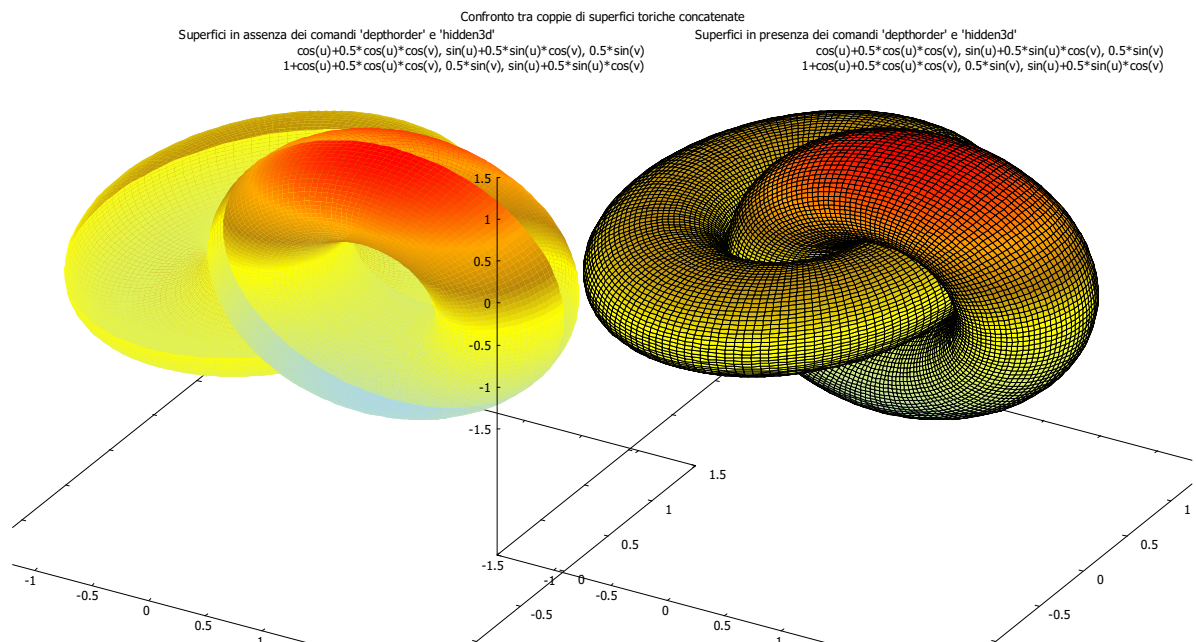
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie 'Elicoide iperbolica'"
set xyplane at 0.; set xlabel "X"; set ylabel "Y"; set zlabel "Z";
unset key
# unset tics # opzionale
set samples 200, 200; set isosamples 160, 160;
set size square 1., 1.
set pm3d depthorder hidden3d
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", 10 "#ffff00", \
                    15 "#ff4500", 18 "#dd5500", 21 "#f00000") # colorbox personalizzata
# unset colorbox # opzionale
set view 70, 30, 1, 1
set parametric
c=-3.*sqrt(3.)/2; s=7./2
x(u,v)=c*sinh(v)*sin(u)+s*cosh(v)*cos(u) # x-componente di r(u,v)
y(u,v)=-c*sinh(v)*cos(u)+s*cosh(v)*sin(u) # y-componente di r(u,v)
z(u,v)=c*u+s*v # z-componente di r(u,v)
#
splot [-2*pi:2*pi] [-10.:10.] x(u,v), y(u,v), z(u,v) w pm3d
unset parametric

```



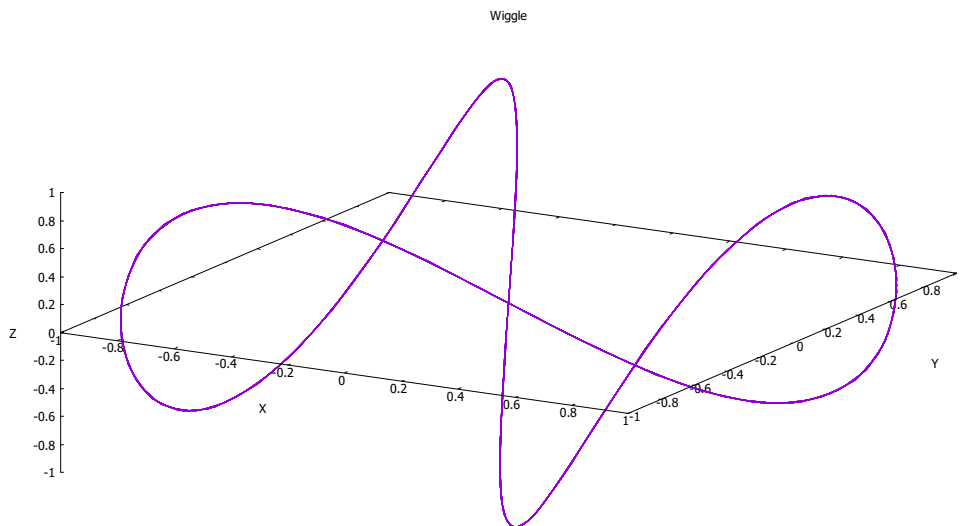
L'effetto grafico del comando `pm3d depthorder hidden3d` può essere apprezzato meglio attraverso l'esempio classico seguente (impegnativo), di confronto tra superfici toroidali concatenate:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set multiplot title "Confronto tra coppie di superfici toriche concatenate"
# titolo condiviso da più sistemi di riferimento
set title "Superfici in assenza dei comandi 'depthorder' e 'hidden3d'" # 1.o titolo
# unset key # nasconde, da qui in poi, l'espressione matematica di qualsiasi grafico
set parametric
set urange [0.:2*pi]; set vrange [0.:2*pi]
set samples 150, 150; set isosamples 80, 80
set size square 0.55, 1.0 # parametri b, h (v. p. 2)
set xtics 0.5; set ytics 0.5; set ztics 0.5 # opzioni tacche di misura sugli assi e passi rispettivi
set view 60, 30, 1.5, 0.9
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# colorbox personalizzata del 1.o grafico
unset colorbox # eliminazione (opzionale) della 'palette' del 1.o grafico
splot (1.+0.5*cos(v))*cos(u), (1.+0.5*cos(v))*sin(u), 0.5*sin(v) w pm3d, \
      1.+(1.+0.5*cos(v))*cos(u), 0.5*sin(v), (1.+0.5*cos(v))*sin(u) w pm3d
#
set title "Superfici in presenza dei comandi 'depthorder' e 'hidden3d'" # 2.o titolo
# unset key # nasconde l'espressione matematica di tutti i grafici successivi al precedente
set origin 0.44, 0. # posizione dell'origine del 2.o grafico
set size square 0.55, 1.0 # parametri b, h (v. p. 2)
set colorbox vertical user origin 0.93, 0.15 size 0.02, 0.5
set format cb "%.1f" # precisione della posizione delle tacche sugli assi vs. la \
                     # colorbox (cb) di 1 cifra decimale (1 'floating point'), v. 'Help')
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# colorbox personalizzata del 2.o grafico
unset colorbox # eliminazione (opzionale) della 'palette' del 2.o grafico
splot (1.+0.5*cos(v))*cos(u), (1.+0.5*cos(v))*sin(u), 0.5*sin(v) w pm3d, \
      1.+(1.+0.5*cos(v))*cos(u), 0.5*sin(v), (1.+0.5*cos(v))*sin(u) w pm3d
unset multiplot # disattivazione della presenza simultanea di più sistemi di riferimento
unset parametric
```



La dipendenza delle tre coordinate rettangolari da *uno solo* dei parametri di superficie *predefiniti*, e.g., $v \Rightarrow (x; y; z) \equiv (x(v); y(v); z(v))$, riduce il grafico a una *linea* nello spazio $X \times Y \times Z$, ovviamente:

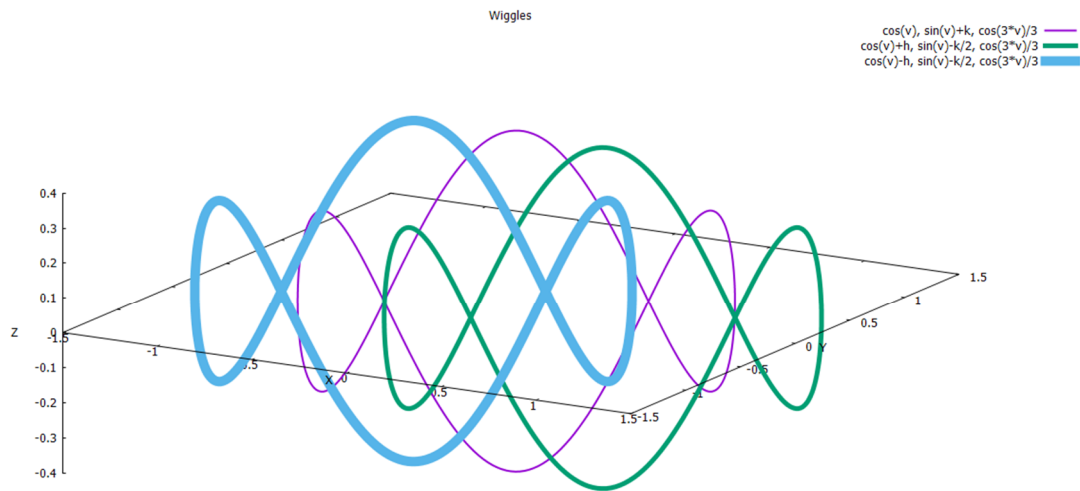
```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Wiggle"
unset key
set samples 300
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set vrange [0.:2*pi]
#
splot cos(3*v), sin(6*v), cos(9*v)      # se non è indicato, lo spessore predefinito della linea è lw = 1
#
unset parametric
```



Un confronto grafico *simultaneo* tra più linee parametriche in $X \times Y \times Z$ aventi *spessori diversi* (l'assegnazione di un colore differente è automatica dopo il tracciamento di ogni linea) può essere codificato come nell'esempio elementare seguente:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Wiggles"
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 300, 300; set isosamples 200, 200
set parametric
set vrange [0.:2*pi]
h=0.5          # parametro reale (utente)
k=sqrt(3.)/35  # parametro reale (utente)
#
splot cos(v), sin(v)+k, cos(3*v)/3 lw 2, \
      cos(v)+h, sin(v)-k/2, cos(3*v)/3 lw 5, \
      cos(v)-h, sin(v)-k/2, cos(3*v)/3 lw 10
#
unset parametric
```

L'apparenza grafica è mostrata nella figura sottostante. A *ogni* linea è assegnato il proprio spessore. Non essendo stato specificato alcun *colore* delle linee da parte dell'utente, esse vengono distinte cromaticamente dal programma secondo una sequenza interna predefinita.



Osservazioni generali

- Sulla *Main tool bar* della finestra grafica, cliccare sul tasto **Settings** e controllare se *togliendo* i segni di spunta dalle boxes **Antialias** e/o **Rounded line ends**, assegnati di default dal programma, l'immagine grafica migliora.
- Inoltre, diversamente da versioni precedenti del programma, il *colore di sfondo* scelto per la finestra grafica *non* è esportabile con il grafico.

■

8. Introduzione di quantità complesse nelle espressioni

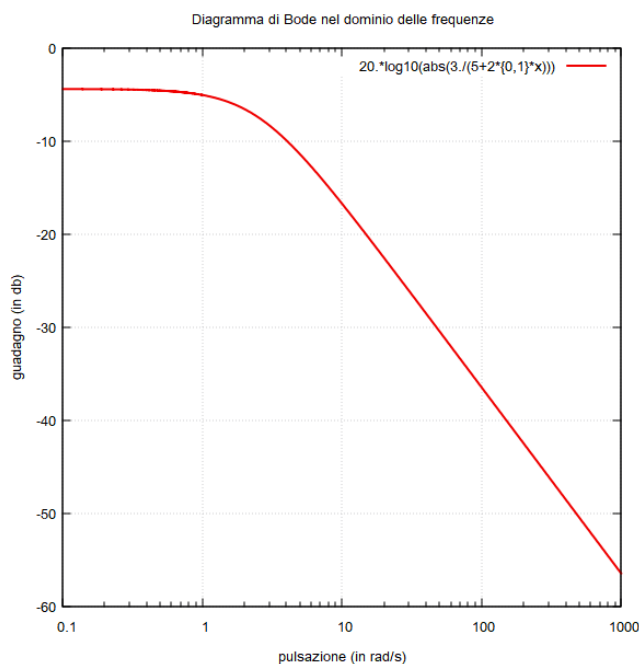
Le *costanti complesse* vengono espresse nella sintassi $\{\langle \text{real} \rangle, \langle \text{imag} \rangle\}$, dove, $\langle \text{real} \rangle$ e $\langle \text{imag} \rangle$ sono *costanti reali* che rappresentano, nell'ordine, la *parte reale* e la *parte immaginaria* di un *numero (costante) complesso*. Quindi, $\{0., 1.\}$ rappresenta l'*unità immaginaria* i (o j , nei testi di Teoria dei Sistemi e ingegneristici), mentre $\{0.6, -2.2625\}$ rappresenta il numero complesso $3./5 - i(21./8)$. Invece, una *variabile complessa* w – mai usare z (!) – deve essere definita $w(x, y) := x + \{0, 1\} * y$, i.e., come una *funzione* della coppia di coordinate $(x; y)$ (v 8.1, P. 17). Il simbolo convenzionale z sarà riservato, di norma, alla terza coordinata rettangolare *reale* (lo è di default in GNUPLOT).

La possibilità di usare costanti complesse e l'unità immaginaria, insieme con le funzioni **abs**, **arg**, **real**, **imag**, rende possibile tracciare i *diagrammi di Bode* e di *Nyquist* della *risposta in frequenza* di un dato sistema periodico, del quale, sia nota la *funzione di trasferimento*. Pertanto, per il sistema caratterizzato dalla funzione di trasferimento

$$H(s) = \frac{3}{5 + 2s} \Rightarrow H(i\omega) = \frac{3}{5 + 2i\omega},$$

il diagramma di Bode del guadagno (in decibel, db), per valori della pulsazione ω compresi tra 0.1 e 1000 [rad/s] (nel *dominio delle frequenze*, ω è l'*ascissa*) si può costruire come segue:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Diagramma di Bode nel dominio delle frequenze"
set grid
set xlabel 'pulsazione (in rad/s)'; set ylabel 'guadagno (in db)'
set samples 1000
set xrange [0.1:1000]; set yrange [-60.:0.]
set size square 1.0, 1.0      # parametri b, h (v. p. 2)
set logscale x                # assegnazione della scala logaritmica sull'asse X
#
plot 20.*log10(abs(3./(5+2*{0,1}*x))) lc rgb "#f00000" lw 2      # log10=logaritmo in base 10; \
                                                                    abs=valore assoluto
unset logscale x
```



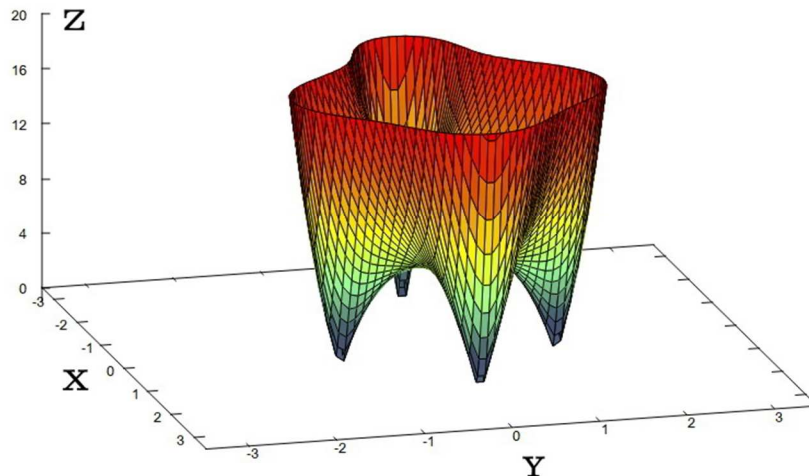
8.1 Rappresentazione 3-dim della funzione valore assoluto $z = |f(w)|$, con $w = x + iy$

Poiché \mathbb{R}^2 è isomorfo al piano complesso $\{x, y\} \equiv \{\Re(w), \Im(w)\}$, si può applicare la funzione $w := x + iy \mapsto f(x, y)$ dal dominio $\Omega \subseteq X \times Y (\equiv \mathbb{R}^2)$ al codominio $\mathcal{C}(\Omega) \subset \mathbb{C}$.

Prendendo come esempio il polinomio quartico $P(w) := 2w^4 - 5w^3 + 2w^2 + 4w - 11$, si ha

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico della funzione reale non-negativa z(x,y):=|P(w)|, con w=x+iy" # (opzionale)
unset key
set xyplane at 0.
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-3.5:3.5]; set yrange [-3.5:3.5]; set zrange [-1.:20.]
set xtics 1.; set ytics 1.; set ztics 4.
set samples 125, 125; set isosamples 80, 80
set pm3d depthorder hidden3d
set size square 1., 1. # parametri b, h (v. p. 2)
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox # elimina la vista della colorbox
set view 60, 75, 1, 1
#
w(x,y)=x+{0,1}*y # definizione della variabile complessa w come funzione di x e di y
P(x,y)=2*w(x,y)**4-5*w(x,y)**3+2*w(x,y)**2+4*w(x,y)-11. # espressione della funzione P(w) (polinomio complesso di 4° grado)
#
z(x,y)=abs(P(x,y)) # funzione reale cartesiana 'valore-assoluto' z=|P(w)|
#
splot z(x,y) w pm3d
```

Grafico della funzione reale non-negativa $z(x,y):=|P(w)|$, con $w=x+iy$



9. Tracciamento di punti isolati in coordinate cartesiane

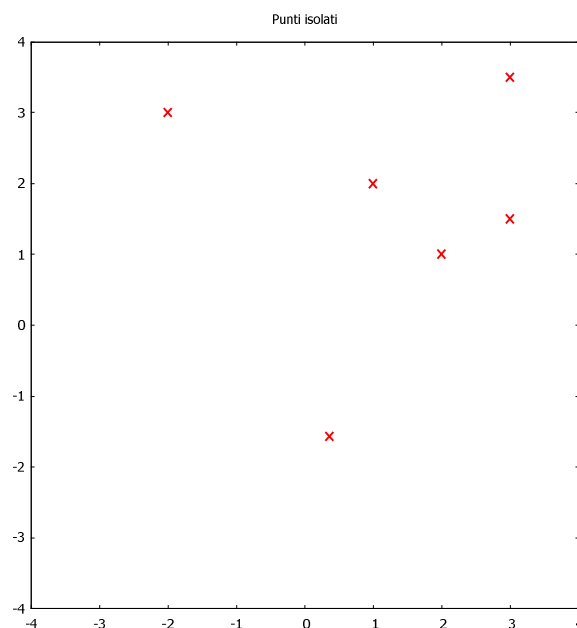
GNUPLOT traccia punti *isolati* in coordinate *cartesiane* trattandoli come linee. Di ciascun punto, si possono scegliere

- l'*immagine grafica* (**ls = line style**: 1, 2, 3, ..., i.e., +, x, *, ..., etc.), secondo una sequenza prefissata interna al programma,
- il *colore* (**lc = line color**), e.g., in formato RGB,
- lo *spessore* (**lw = line width**), crescente: 1, 2, 3,

Le coppie di coordinate cartesiane vanno tabulate *dopo* il comando **plot** della loro esecuzione grafica, scritte in formato *reale-decimale* e *troncate*, se espresse da un numero comunque eccessivo di cifre. Inoltre, *ogni* coppia *deve* essere seguita dal comando di arresto **e** (= ends).

Come esempio, una sintassi adeguata al caso del tracciamento di 6 punti in *stile* 'x' (ls 2), di *colore* rosso (lc rgb "#f00000") e di *spessore* '2' (lw 2) è la seguente:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Punti isolati"; unset key
set xrange [-4.:4.]; set yrange [-4.:4.]      # (utente)
set size square 1., 1.                      # parametri b, h (v. p. 2)
plot '-' w p ls 2 lc rgb "#f00000" lw 2, '-' w p ls 2 lc rgb "#f00000" lw 2, '-' w p ls 2 lc rgb "#f00000" lw 2, \
    '-' w p ls 2 lc rgb "#f00000" lw 2, '-' w p ls 2 lc rgb "#f00000" lw 2, '-' w p ls 2 lc rgb "#f00000" lw 2
# '-' w p   specifica le qualità di ogni punto-grafico (e.g., stile, colore, spessore)
1.      2.      # (x1;y1) = (1;2), in formato reale-decimale
e
2.      1.      # (x2;y2) = (2;1), in formato reale-decimale
e
3.      1.5     # (x3;y3) = (3;3/2)
e
3.      3.5     # (x4;y4) = (3;7/2)
e
0.368 -1.571    # (x5;y5) = (1/e;-pi/2), in formato reale-decimale (troncato)
e
-2.     3.      # (x6;y6) = (-2;3), in formato reale-decimale
e
```



10. Interpolazione di dati sperimentali con il comando **fit** e l'opzione **smooth**

Siano stati raccolti dei dati, con i quali, a ogni valore della grandezza x (ascissa), è associato un valore della grandezza y (ordinata) con una *deviazione standard* z (indicatrice della qualità dei dati $\{x, y\}$ espressa dal *peso* w di ciascun dato come $z \equiv 1/w^{1/2}$). In un sistema di assi cartesiani, si debbano riportare le coppie $\{x, y\}$ di dati sperimentali e, quindi, interpolare i punti mediante una funzione nota $y = f(x)$, di natura *prestabilita*. Nell'esempio seguente, è usato il comando **fit**:

10.1 salvare le terne $\{x, y, z\}$ in un file-dati di *testo*, e.g., TXT o un qualsiasi altro *editor*. Nel file, le righe di *commento* devono iniziare, al solito, con il carattere **#**. I dati devono essere incolonnati, e.g., con il tabulatore della tastiera; il file, **dataset1.txt**, sia stato posto nella cartella **datafolder**, altrove nel sistema.

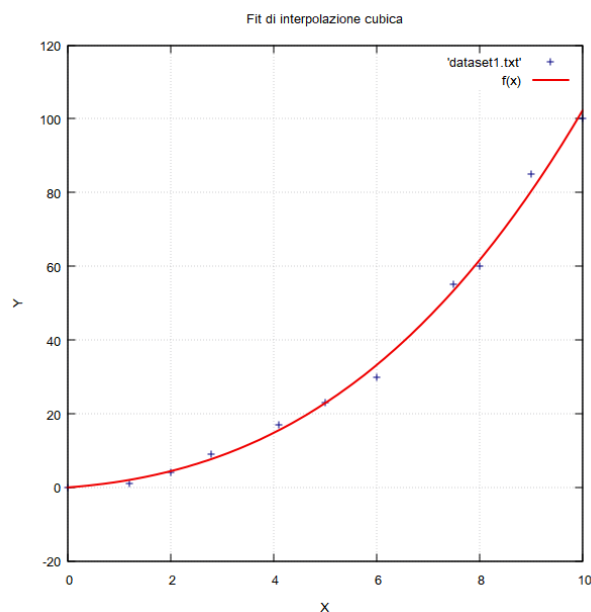
Nelle terne $\{x, y, z\}$, z può costituire la *deviazione standard* delle coppie $\{x, y\}$.

Più in generale, se si vuole *imporre* il passaggio della funzione interpolante per certi punti, si assegna ad essi un valore z di deviazione *minore* di quello degli altri punti;

10.2 assegnare la funzione di interpolazione parametrica dei dati, e.g., la funzione polinomiale *cubica* generale, $f(x) := a + bx + cx^2 + dx^3$. Le istruzioni stesse che assegnano la funzione interpolante, ne determinano anche i coefficienti parametrici a, b, c, d , ottimizzati vs. i dati sperimentali. Inoltre, i dati (punti) sono riportati nel piano del grafico insieme con la curva interpolante *ottimizzata* corrispondente;

10.3 individuare *preventivamente* il directory della cartella **datafolder** dalla finestra di *editing*, con il comando di ricerca **Change directory**, in giallo, posto sulla barra delle funzioni, poi, caricare le istruzioni seguenti:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Fit di interpolazione cubica"
set xlabel "X"; set ylabel "Y"; set grid; set size square 1.0, 1.0      # parametri b, h (v. p. 2)
#
f(x)=a+b*x+c*x**2+d*x**3
fit f(x) 'dataset1.txt' u 1:2:3 via a,b,c,d                          # u=using
#
plot 'dataset1.txt' lc "#000081", f(x) lc "#f00000" lw 2            # file contenente i dati da interpolare con f(x), \
                                                                    grafico di f(x)
```



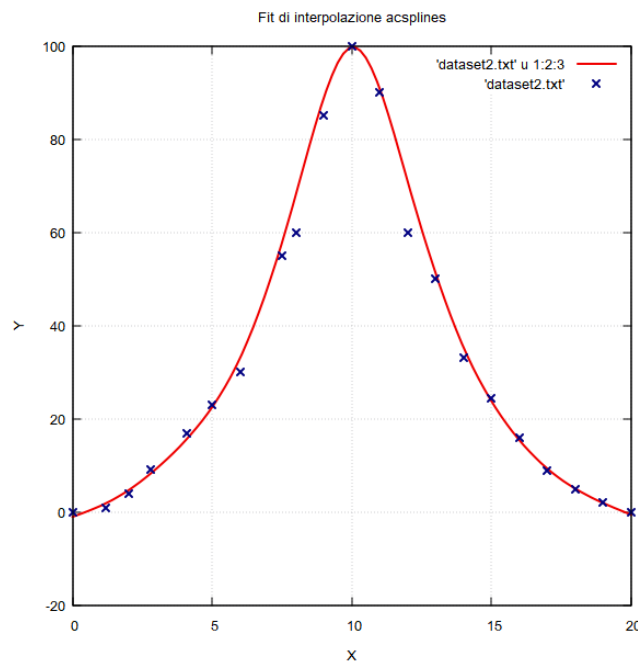
Se si desidera conoscere anche il valore che la funzione interpolante f assume per un certo valore di x (e.g., $x = 4.7 \equiv 47./10$), basta far seguire *immediatamente* all'esecuzione grafica il comando

`print f(4.7)` # comando – in formato reale! – per l'uso di GNUPLOT come calcolatrice

Nel caso in cui non si desideri interpolare con un tipo specifico di funzione è disponibile l'opzione `smooth` del comando `plot`. Tale opzione è applicabile a cinque metodi d'interpolazione: `acsplines`, `bezier`, `csplines`, `sbezier`, `unique`. Ancora, le terne $\{x, y, z\}$ *incolonnate* dei dati vanno salvate in un file di testo, e.g., `dataset2.txt`, anch'esso contenuto in *datafolder* e analogo a `dataset1.txt`. Ora, però, i dati z della 3^a colonna devono costituire – *obbligatoriamente* – gli scarti (errori) assegnati alle misure specifiche. Così, e.g. (ricordarsi del comando di ricerca `Change directory!`),

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set grid
set title "Fit di interpolazione acsplines"
unset parametric
set xlabel "X"; set ylabel "Y"
set size square 1., 1. # parametri b, h (v. p. 2)
#
plot 'dataset2.txt' u 1:2:3 smooth acsplines lc "#f00000" lw 2, 'dataset2.txt' lc "#000081" lw 2
# grafico della curva y='acsplines' (v. 'Help', per interpolazioni alternative e loro significati) \
# grafico dei dati da interpolare con 'acsplines'
```

Il risultato è rappresentato dal grafico seguente:



Quanto più viene aumentato il valore dello scarto z di un dato tanto più la curva interpolante si avvicina al punto, tendendo, però, ad assumere, un profilo globale *maggiormente irregolare*. ■

11. Visualizzazione degli errori (scarti) nei dati sperimentali

Una rappresentazione grafica elementare 2-dim degli errori *tabulati* di un insieme di dati richiede che GNUPLOT legga un certo *dataset3.txt*, contenente da 3 a 6 colonne, ancora dalla cartella *datafolder*. Poiché le colonne 1^a e 2^a sono *riservate* alle coppie $\{x, y\}$ di valori associati *più attendibili* (misurati o medî) dei dati, ogni colonna successiva conterrà gli *errori* corrispondenti (scarti) Δx o Δy . Ne risulteranno, così, le configurazioni fondamentali seguenti:

3 colonne:	<code>u 1:2:3 with xerrorbars</code>	$x, y, \Delta x$	con x-errori <i>simmetrici</i> (col. 3),
3 colonne:	<code>u 1:2:3 with yerrorbars</code>	$x, y, \Delta y$	con y-errori <i>simmetrici</i> (col. 3),
4 colonne:	<code>u 1:2:3:4 with xyerrorbars</code>	$x, y, \Delta x, \Delta y$	con x,y-errori <i>simmetrici</i> (col. 3 e 4),
4 colonne:	<code>u 1:2:3:4 with xerrorbars</code>	$x, y, \Delta x, \Delta y$	con x-errori <i>distinti</i> (col. 3 e 4),
4 colonne:	<code>u 1:2:3:4 with yerrorbars</code>	$x, y, \Delta y^-, \Delta y^+$	con y-errori <i>distinti</i> (col. 3 e 4),
6 colonne:	<code>u 1:2:3:4:5:6 with xyerrorbars</code>	$x, y, \Delta x^-, \Delta x^+, \Delta y^-, \Delta y^+$	con x, y-errori <i>distinti</i> (col. 3, 4, 5 e 6).

Esempî di comandi grafici:

- `plot 'dataset3.txt' u 1:2:3 with xerrorbars`
i dati della 3^a colonna sono considerati come scarti Δx ;
- `plot 'dataset3.txt' u 1:2:3 with yerrorbars`
i dati della 3^a colonna sono considerati come scarti Δy ;
- `plot 'dataset3.txt' u 1:2:($2+$3):($2-$4) with yerrorbars`
i dati della 3^a e della 4^a colonna – indicati *collettivamente e sequenzialmente* come (\$3) e (\$4) – sono presi come gli *scarti distinti*, superiore e inferiore, per *ciascun* dato-y (\$2);
- `plot 'dataset3.txt' u 1:2:($1+$3):($1-$3):($2+$4):($2-$5) with xyerrorbars`
i dati della 3^a e della 4^a colonna sono presi come $\pm \Delta x$ mentre le colonne 4^a e 5^a forniscono, rispettivamente, gli scarti *distinti*, superiore e inferiore, di *ciascun* dato-y (\$2).

Chiaramente, si possono usare funzioni e combinazioni arbitrarie a piacere:

- `plot 'dataset3.txt' u 1:2:($2*0.07) with yerrorbars`
rappresenta i dati con un errore su *ciascun* dato-y (\$2) del 7% ;
- `plot 'dataset3.txt' u 1:2:($1+4):($1-4):($2+$3):($2-$3) with xyerrorbars`
genera lo scarto *simmetrico costante* ± 4 su *ciascun* dato-x (\$1) e lo scarto *simmetrico* di *ciascun* dato-y, (\$2), con il dato corrispondente della 3^a colonna, (\$3).

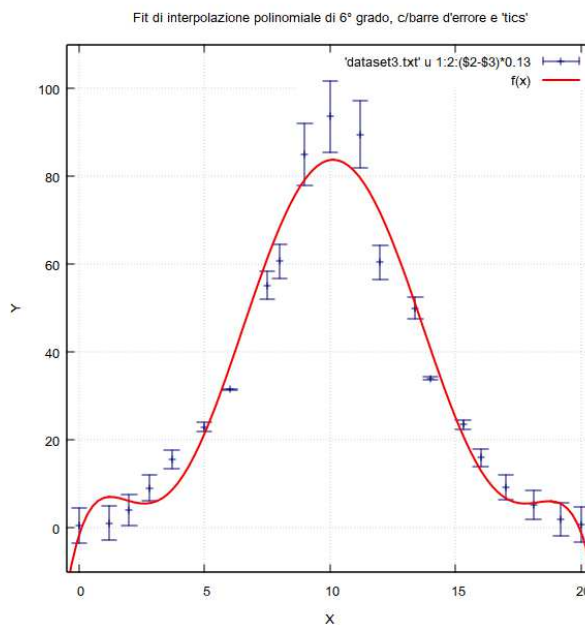
Si può convertire in *scala logaritmica* vs. *y* l'interpolazione dei dati contenuti in *dataset3.txt*, richiedendo un *raccordo lineare* dei punti sperimentali (ci si ricordi del tasto *Change directory*):

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Dati in scala semi-logaritmica vs. y"
set size 0.75, 1.
set logscale y          # definizione della scala logaritmica sull'asse Y
set bars 3.0           # assegnazione della dimensione dei 'tics' (trattini terminali) delle barre
#
plot 'dataset3.txt' u 1:2:($1+4):($1-4):($2+$3):($2-$3) with xyerrorbars lc rgb "#006000", \
'dataset3.txt' u 1:2 notitle w lines lc rgb "#f00000"      # w lines=con raccordo rettilineo dei dati
#
unset bars; unset logscale y
```

■

Come conclusione, ci si può sbizzarrire assemblando alcuni dei vari comandi presentati fin qui. Il loro significato – al livello elementare di questo notebook, almeno – non dovrebbe richiedere (?) chiarimenti ulteriori. Sarà solo necessario, da parte di chi desideri testarli, generare da sé dei file .TXT di dati arbitrari *incolonnati* con il tasto tabulatore e da individuare nel sistema con il tasto *giallo*, **Change directory**, sulla barra degli strumenti nella pagina di *editing*:

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
unset parametric
set title "Fit di interpolazione polinomiale di 6° grado, c/barre d'errore e 'tics'"
set grid
set size square 1., 1. # parametri b, h (v. p. 2)
set xlabel "X"; set ylabel "Y"
set bars 2.0 # 2.0=dimensione dei 'tics', i trattini centrale e terminali di taglio sulle barre
f(x)=a+b*x+c*x**2+d*x**3+e*x**4+f*x**5+g*x**6
fit f(x) 'dataset3.txt' via a,b,c,d,e,f,g
#
plot [-0.5:20.5] [-10:110] 'dataset3.txt' u 1:2:($2-$3)*0.13 with yerrorbars lc "#000081", \
      f(x) lc "#f00000" lw 2
#
unset bars
```



L'interpolazione polinomiale di 6° grado non sembra dare risultati soddisfacenti ai valori delle zone di 'coda' né nell'intorno del massimo dei dati analizzati. È forse opportuna la scelta di una funzione interpolatrice diversa.

Insieme con la guida in linea di GNUPLOT, possono risultare utili il 'tutorial' *Gnuplot in Action.pdf*, free-scaricabile dal Web, i documenti didattici del Prof. Carlo Meneghini, Un. degli Studi Roma-3, Tor Vergata, anch'essi reperibili dal Web, e la pagina

http://www.csse.uwa.edu.au/programming/gnuplot_demos/.

Un. of West Alabama, Livingston, AL.

Altri esempi e modelli matematici

Polinomi di Legendre e Funzioni Associate di Legendre

Polinomio orto-normalizzato di Legendre di grado n

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Polinomi orto-normalizzati di Legendre LP_6(x), LP_11(x)"
# unset key
set xlabel "X"; set ylabel "Y"
set grid; set origin 0.23, 0.
set samples 250
set size 0.55, 1.0
# Espressione generale (da attivare assegnando la coppia parametrica {l,m}):
# LP_n(x)=2.**-n*sum[k=0:floor(n/2)](-1)**k*(2*(n-k))/(k!(n-k)!(n-2*k)!)*x**(n-2*k)
# il valore di n deve essere definito per ogni LP_n(x)
#
# Esempio: grafici di PL_6(x) e di PL_11(x) (richiesta dell'utente):
#
h=6 # grado di LP_h(x)
LP_6(x)=2.**-h*sum[k=0:floor(h/2)](-1)**k*(2*(h-k))/(k!(h-k)!(h-2*k)!)
#
r=11 # grado di LP_r(x)
LP_11(x)=2.**-r*sum[k=0:floor(r/2)](-1)**k*(2*(r-k))/(k!(r-k)!(r-2*k)!)
#
plot [-1.05:1.05] [-1.05:1.05] LP_6(x) lw 2, LP_11(x) lw 2
```

Funzione Associata di Legendre di ordine l e rango m

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Funzione Associata di Legendre ALF_9_5(x)"
# unset key
set xlabel "X"; set ylabel "Y"
set grid; set origin 0.23, 0.
set samples 250
set size 0.55, 1.0
set xrange [-25.:25.] # range-x della finestra grafica (utente)
# set yrange [y:Y] # range-y della finestra grafica (utente)
#
# Espressione generale (da attivare assegnando la coppia parametrica {l,m}):
# ALF_l'm'(x)=(1.-x**2)**(m/2)*2.**-l*sum[k=0:floor((l-m)/2)](-1)**k*(2*(l-k))/(k!(l-k)!(l-m-2*k)!)
# i valori ammissibili di l e di m devono essere definiti per ogni ALF_l'm'(x). \
# L'indice k della somma varia sotto le condizioni simultanee: 0≤m≤l, max{k}=floor((l-m)/2)).
#
# Esempio: grafico di ALF_9'5'(x)
#
l=9 # ordine di ALF_l'm'(x) (utente)
m=5 # rango di ALF_l'm'(x) (utente)
ALF_9'5'(x)=(1.-x**2)**(m/2)*2.**-l*sum[k=0:floor((l-m)/2)](-1)**k*(2*(l-k))/(k!(l-k)!(l-m-2*k)!)
#
plot ALF_9'5'(x) lw 2
```


Fattore polare dell'Armonica Sferica orto-normalizzata di ordine l e rango m

Il grafico del fattore polare $\Theta(\theta)$ dell'Armonica Sferica orto-normalizzata $Y_{l,m}(\theta, \varphi)$ risulta, qualitativamente, identico o simmetrico vs. l'asse X a quello della Funzione Associata di Legendre $P_{l,m}(u)$, degli stessi ordine l e rango m . Infatti, $Y_{l,m}(\theta, \varphi)$ e $P_{l,m}(u)$, con $u = \cos \theta$, sono proporzionali tra loro nella convenzione di fase CSW (v., e.g., il notebook: **Strumenti, applicazioni e note di Metodi Matematici della Fisica**). Il codice è riportato qui di seguito, per completezza ... e, magari, a beneficio solidale di qualche 'artigiano solitario' in Fisica Quantistica.

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Fattore polare dell'Armonica Sferica orto-normalizzata 7'2'-esima"
# unset key
set xlabel "X"; set ylabel "Y"
set grid; set origin 0.23, 0.
set samples 250, 250
set size 0.55, 1.
set xrange [-25.:25.]          # x-range della finestra grafica (utente)
# set yrange [<y1>:<y2>]      # y-range della finestra grafica (utente)
#
# Espressione generale (da attivare assegnando la coppia parametrica {l,m}):
# pf_l'm'(x)=(-1)**m*2**-(l+1)*((2*l+1)*(l-m)!/(pi*(l+m)!))**(1/2)*(1-x**2)**(m/2)*\
#           sum[k=0:floor((l-m)/2)](-1)**k*(2*(l-k)!/(k!*(l-k)!*(l-m-2*k)!)*x**(l-m-2*k)
#           # i valori ammissibili di l e di m devono essere definiti per ogni pf_l'm'(x). \
#           L'indice k di somma varia sotto le condizioni simultanee 0≤m≤l, max{k}=floor((l-m)/2)]
#
# Esempio: grafico di pf_7'2'(x)
l=7          # ordine di pf_l'm'(x) (utente)
m=2          # rango di pf_l'm'(x) (utente)
pf_7'2'(x)=(-1)**m*2**-(l+1)*((2*l+1)*(l-m)!/(pi*(l+m)!))**(1/2)*(1-x**2)**(m/2)*\
#           sum[k=0:floor((l-m)/2)](-1)**k*(2*(l-k)!/(k!*(l-k)!*(l-m-2*k)!)*x**(l-m-2*k)
#
plot pf_7'2'(x) lw 2
```

■

Grafico 3-dim della Funzione $(x,y) \mapsto z = |\Gamma(w)|$, con $w = x + iy$ (v. P. I)

Le istruzioni presentate contengono un'assegnazione di 'tavolozza' cromatica (**palette**) alternativa a quella di default. Al solito, il carattere **#** in una riga precede una stringa interpretata da GNUPLOT, fino a fine-riga, come un commento e può, quindi, essere usato per *selezionare\deselezionare* istruzioni singole.

Il grafico risultante è mostrato all'inizio di questo notebook, a P. II. Si veda, anche, l'esempio generale 8.1, p. 17.

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Grafico della funzione  $z(x,y) := |\Gamma(w)|$ , con  $w = x + iy$ "
unset key
set xyplane at 0.
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-5.:5.]; set yrange [-5.:5.]; set zrange [0.:10.]
set xtics 1.0; set ytics 1.0; set ztics 4.0
set samples 150, 150; set isosamples 120, 120
set pm3d depthorder hidden3d
set size square 1.0, 1.0
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# colorbox personalizzata
unset colorbox # elimina la vista della colorbox
set view 63, 332, 1, 1
#
€=0.5772156649 # costante di Euler-Mascheroni (approx.)
G(x)=exp(-€*x)/x*exp(sum[n=1:1000]log(n*exp(x/n)/(x+n)))
z(x,y)=abs(G(x))*exp(-0.5*sum[k=0:1000]log(1.+y**2/(x+k)**2))
#
splot z(x,y) w pm3d
```

(†) Con GNUPLOT, ogni *produttoria* (qui, di Γ , è stata applicata la rappresentazione *moltiplicativa* di Weierstrass) deve essere riespressa come *esponenziale della somma e/o differenza dei logaritmi dei valori assoluti dei fattori*.

Grafici 3-dim delle funzioni $(x,y) \mapsto z = \operatorname{Re}(\Gamma(w))$ e $(x,y) \mapsto z = \operatorname{Im}(\Gamma(w))$, con $w = x + iy$ (v. P. 97)

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Grafico della superficie reale  $r(x,y) = \operatorname{Re}(\Gamma(w))$ , con  $w = x + iy$ "
unset key
set xyplane at 0
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-5.:5.]; set yrange [-5.:5.]; set zrange [-5.:5.]
set xtics 1.0; set ytics 1.0; set ztics 5.0
set samples 150, 150; set isosamples 120, 120
set pm3d depthorder hidden3d
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox # elimina la vista della colorbox
set view 56, 50, 1, 1
set size square 1.0, 1.0
€=0.5772156649 # costante di Euler-Mascheroni (approx.)
w(x,y)=x+{0,1}*y
#
r(x,y)=real(exp(-€*w(x,y))/w(x,y)*exp(sum[n=1:1000]log(n*exp(w(x,y)/n)/(w(x,y)+n))))
#
splot r(x,y) w pm3d
```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Grafico della superficie reale  $i(x,y)=\text{Im}(\Gamma(w))$ , con  $w=x+iy$ "
unset key
set xyplane at 0
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-5.:5.]; set yrange [-5.:5.]; set zrange [-5.:5.]
set xtics 1.0; set ytics 1.0; set ztics 5.0
set samples 150, 150; set isosamples 120, 120
set pm3d depthorder hidden3d
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox # elimina la vista della colorbox
set view 57, 54, 1, 1
set size square 1.0, 1.0
€=0.5772156649 # costante di Euler-Mascheroni (approx.)
w(x,y)=x+{0,1}*y
#
i(x,y)=imag(exp(-€*w(x,y))/w(x,y)*exp(sum[n=1:1000]log(n*exp(w(x,y)/n)/(w(x,y)+n))))
#
splot i(x,y) w pm3d

```

**Grafico 3-dim della Funzione Beta di Euler, $(x,y) \mapsto z=B(x,y)$,
con $x, y > 0$ (v. P. 98)**

Questa rappresentazione della Funzione B di Euler, combinazione formalmente elementare di Funzioni Γ , è, però, di esecuzione piuttosto *pesante* per GNUPLOT. È opportuno attendere qualche secondo dalla comparsa del grafico perché siano completate le operazioni interne di chiusura del programma. Questo, altrimenti, potrebbe bloccarsi.

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Grafico della Funzione Beta di Euler,  $z=B(x,y)$ "
unset key
set xyplane at 0.
# set xlabel "X"; set ylabel "Y"; set zlabel "Z" # assi cartesiani contrassegnati da lettere maiuscole, \
# poi, sostituite graficamente con 'Paint.Net'

set xtics 1.; set ytics 1.; set ztics 5.
set samples 150, 150; set isosamples 125, 125
set pm3d depthorder hidden3d
set size square 1., 1. # parametri b, h (v. p. 2)
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", 10 "#ffff00", \
15 "#ff4500", 18 "#dd5500", 21 "#f00000") # colorbox personalizzata
unset colorbox
set view 70, 115, 1, 1
#
€=0.5772156649 # costante di Euler-Mascheroni (approx.)
g(x)=exp(-€*x)/x*exp(sum[n=1:1000]log(n*exp(x/n)/(x+n)))
g(y)=exp(-€*y)/y*exp(sum[n=1:1000]log(n*exp(y/n)/(y+n)))
G(x,y)=exp(-€*(x+y))/(x+y)*exp(sum[n=1:1000]log(n*exp((x+y)/n)/(x+y+n)))
#
B(x,y)=g(x)*g(y)/G(x,y)
#
splot [0.:5.] [0.:5.] [0.:60.] B(x,y) w pm3d # la 'box grafica' e la funzione B sono dichiarate insieme \
in sequenza prescritta alternativa

```

Grafici 2-dim delle Funzioni di Bessel Ordinarie di 1° e di 2° tipo

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico della Funzione di Bessel Ordinaria di 1° tipo J0(x)"
unset key
set samples 500
set grid
set size square 1., 1.      # parametri b, h (v. p. 2)
#
J0(x)=sum[k=0:65](-1)**k/((2.**k*k!)**2)*x**(2*k)      # cf/c grafici a P. 3
#
plot [-20.:20.] [-0.7:1.1] J0(x)

```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
      # prima di eseguire lo script, specificare l'ordine $ > 0 nel simbolo J$(x)
unset key
set title "Grafico della Funzione di Bessel Ordinaria di 1° tipo J$(x)"      # opzionale
set samples 500
set grid
set size square 1., 1.      # parametri b, h (v. p. 2)
#
n=$
J$(x)=sum[k=0:65](-1)**k/(2.**k*(k+n)!*(k+n)!)*x**(2*k+n)      # cf/c grafici a P. 3
#
plot [-20.:20.] [-0.7:1.1] J$(x)

```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico della Funzione di Bessel Ordinaria di 2° tipo N0(x)"
unset key
set samples 500
set grid
set size 1., 1.
#
€=0.5772156649      # costante di Euler-Mascheroni (approx.)
N0(x)=2/pi* \
      (log(x/2) + € \
      + sum[k=1:65](-1)**k/((2.**k*k!)**2)*(log(x/2) + € - sum[p=1:k]1./p)*x**(2*k) \
      )
#
plot [0:20] [-0.7:1.1] N0(x)

```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
      # prima di eseguire lo script, specificare l'ordine $ > 0 nel simbolo N$(x)
set title "Grafico della Funzione di Bessel Ordinaria di 2° tipo N$(x)"      # opzionale
unset key
set samples 500
set grid
set size square 1., 1.      # parametri b, h (v. p. 2)
#
n=$
€=0.5772156649      # costante di Euler-Mascheroni (approx.)
N$(x)=2/pi* \
      ( \
      1./n!*(log(x/2) + € - 1./2*sum[p=1:n]1./p)*(x/2)**n \
      + (sum[k=1:65] (-1)**k/(2.**k*(k+n)!*(k+n)!*(log(x/2) + € - (sum[p=1:k]1./p) \

```

```

- 1./2*sum[p=1:n]1./(k+p))*x**(2*k+n)) \
- 1./2*sum[p=0:n-1](n-p-1)!/(2.** (2*p-n)*p!)*x**(2*p-n) \
)
#
plot [0.:20.] [-7.1:1.1] N$(x)

```

Grafici 3-dim delle Funzioni di Bessel Ordinarie di 1° e di 2° tipo, $r = (x^2 + y^2)^{1/2}$ (v. P. 99-100) (†)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Grafico di J_0(r)"
unset key
set xyplane at 0
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-20.1:20.1]; set yrange [-20.1:20.1]; set zrange [-0.6:1.1]
set xtics 4.0; set ytics 4.0; set ztics 0.5
set samples 100, 100; set isosamples 80, 80
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox
set pm3d depthorder hidden3d
set view 40, 30
set size square 1.0, 1.0
r(x,y)=sqrt(x**2+y**2)
#
J0(x,y)=sum[k=0:65](-1)**k/(2.**k*k!)**2*r(x,y)**(2*k)
#
splot J0(x,y) w pm3d

```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# prima di eseguire la routine, specificare l'ordine intero $ > 0 nel simbolo N$(x,y)
# set title "Grafico di J$(r)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-20.1:20.1]; set yrange [-20.1:20.1]; set zrange [-0.6:1.1]
set xtics 4.; set ytics 4.; set ztics 0.5
set samples 100, 100; set isosamples 80, 80
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox
set pm3d depthorder hidden3d
set view 40, 30
set size square 1., 1. # parametri b, h (v. p. 2)
n=$ # formato INTERO (senza punto), NON reale!
r(x,y)=sqrt(x**2+y**2)
#
J$(x,y)=sum[k=0:65](-1)**k/(2.** (2*k+n)*k!*(k+n)!)*r(x,y)**(2*k+n)
#
splot J$(x,y) w pm3d

```

(†) Realizzazioni grafiche 3-dim delle funzioni $r \mapsto J_0, J_1, N_0, N_1$ sono mostrate anche nelle pagine iniziali del math-notebook:

Rappresentazioni in serie di potenze delle Funzioni di Bessel in \mathbb{R} .

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set time          # opzionale
# set title "Grafico di z = N0(r)"
unset key
set xyplane at 0
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-20.1:20.1]; set yrange [-20.1:20.1]; set zrange [-1.8:1.5]
set xtics 4.0; set ytics 4.0; set ztics 1.0
set samples 100, 100; set isosamples 65, 65
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox
set pm3d depthorder hidden3d
set view 71, 20
set size square 1.0, 1.0
#
r(x,y)=sqrt(x**2+y**2)
€=0.5772156649
#
N0(x,y) = 2/pi * \
( \
    log(r(x,y)/2) + € \
    + sum[k=1:65](-1)**k/((2.**k*k!)*(log(r(x,y)/2) + € \
    - sum[p=1:k]1./p)*r(x,y)**(2*k) \
) \
)
#
splot N0(x,y) w pm3d

```

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set time          # opzionale
# prima di eseguire la routine, specificare l'ordine intero $ > 0 nel simbolo N$(x,y)
# set title "Grafico di z = N$(r)"
unset key; set xyplane at 0
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-20.1:20.1]; set yrange [-20.1:20.1]; set zrange [-2.1:1.5]
set xtics 4.; set ytics 4.; set ztics 1.
set samples 100, 100; set isosamples 65, 65
set palette defined (0 "#606c88", 3 "#3f4c6b", 7 "#99ff99", 11 "#ffff00", 16 "#dd5500", 20 "#f00000")
# colorbox personalizzata
unset colorbox
set pm3d depthorder hidden3d
set view 58, 19
set size square 1.0, 1.0 # parametri b, h (v. p. 2)
#
n=$ # formato INTERO (senza punto), NON reale!
r(x,y)=sqrt(x**2+y**2)
€=0.5772156649 # costante di Euler-Mascheroni (approx.)
#
N$(x,y)=2/pi * \
( \
    1./n!*(log(r(x,y)/2) + € - 1./2*sum[p=1:n]1./p)*(r(x,y)/2)**n \
    + (sum[k=1:65](-1)**k/(2.**k*(k+n)!*(k+n)!*(log(r(x,y)/2) + € \
    - (sum[p=1:k]1./p) - 1./2*sum[p=1:n]1./p)*(r(x,y)**(2*k+n)) \
    - 1./2*sum[p=0:n-1](n-p-1)!/(2.**k*(k+n)!*(k+n)!*(log(r(x,y)/2) + € \
    - (sum[p=1:k]1./p) - 1./2*sum[p=1:n]1./p)*(r(x,y)**(2*k+n)) \
) \
)
#
splot N$(x,y) w pm3d

```

Osservazioni

1. Nella scrittura di formule e di espressioni di funzioni soggette all'elaborazione numerica di GNUPLOT, è necessario porre attenzione al *formato delle costanti intere* esplicitate introdotte: esse vanno scritte *sempre* in formato *reale*, i.e., *seguite dal punto decimale* (e.g., $-4.$, non semplicemente -4), fatta eccezione dei *fattoriali* e degli *indici correnti* delle sommatorie.

Infatti, si osservino le differenze – dalle conseguenze *drammatiche* nei calcoli più complicati! – tra i risultati, sia pure restituiti da GNUPLOT in formato reale,

```
print sum[n=1:10]-1+5/n**3
-5.0
```

risultato *errato*, corrispondente alla sola *parte intera* del risultato corretto;

```
print sum[n=1:10]-1+5./n**3
-4.01234007162903
```

risultato approssimato ma *corretto*: la somma di un numero *intero* e di un numero *reale* è un numero *reale*.

2. GNUPLOT interpreta l'intera stringa che segue il comando $\text{sum}[n=a:b]$ ($\equiv \sum_{n=a}^b$) come il termine generale u_n di una somma. Se tale somma è l'ultimo comando da eseguire, GNUPLOT riconosce u_n correttamente, altrimenti, la somma deve essere posta *tra parentesi*: $(\text{sum}[n=a:b]u_n)$. Questo può essere necessario in presenza sia di altri addendi *estranei* a u_n e in posizione *successiva* alla somma sia, soprattutto, di somme *nidificate* in u_n stesso, somme che devono essere calcolate *per prime* perché, a loro volta, hanno n come valore *massimo* (variabile!) del proprio indice corrente j :

$$\text{sum}[n=a:b](\underbrace{\dots (\text{sum}[j=k:n]v_j) \dots}_{u_n}).$$

Qui sopra, evidentemente, gli indici correnti sono $j = k, k+1, \dots, n$ (interno) e $n = a, a+1, \dots, b$ (esterno).

Come esempli, si analizzino attentamente le scritture precedenti delle espansioni generali delle *Funzioni di Bessel*.

3. Per la costruzione di *palettes* personalizzate, è conveniente disporre di un *color-picker* per la generazione corretta dei valori cromatici, sia RGB che HEX, da inserire, con la sintassi appropriata (si vedano gli esempi precedenti), nell'argomento del comando `set palette defined (...)`. Molti di questi *tools* sono portabili e si scaricano gratuitamente dal Web (e.g., *ColorCop*, *Just Color Picker*, etc.).

■■■

Raccolta di scripts 2-dim

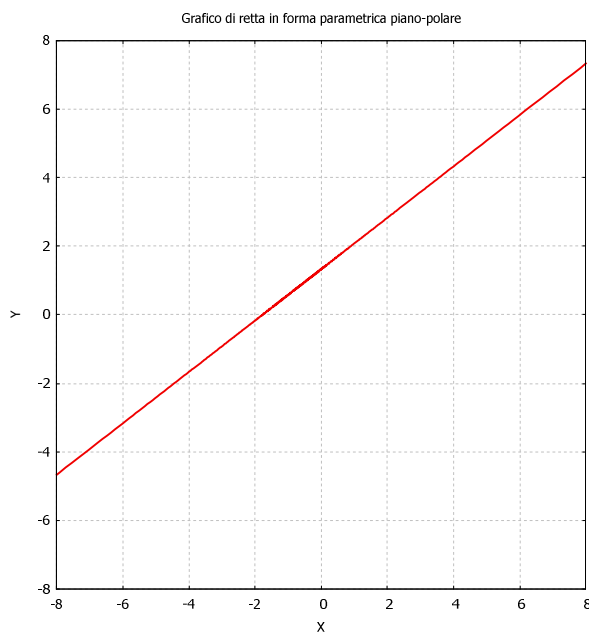
1. Grafico di retta in rappresentazione parametrica piano-polare (forma cartesiana esplicita: $y = mx + q$, con $q \neq 0$)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico di retta in forma parametrica piano-polare"
unset key
set grid # opzionale
set xlabel "X"; set ylabel "Y"
set size square 1., 1. # parametri b, h (v. p. 2)
set samples 1000
set parametric
set trange [-pi/2.:pi/2.]; set xrange [-8.:8.]; set yrange [-8.:8.]
# Assegnazione della coppia di valori {m,q} seguente in formato REALE-decimale (utente):
m=0.75
q=4./3
rho(t)=q/(sin(t)-m*cos(t)) # distanza tra il punto variabile sulla retta e l'origine (polo)
x(t)=rho(t)*cos(t) # equazione piano-polare della retta generica, {x(t),y(t)}, con q≠0
y(t)=rho(t)*sin(t)
#
plot x(t), y(t) lc rgb "#f00000" lw 2 # il colore e lo spessore personalizzati del grafico vanno definiti \
                                        dopo la y-componente parametrica (di CIASCUN grafico se i \
                                        grafici da tracciare contemporaneamente sono più di uno)

unset parametric

```



Osservazione

Si confronti questo script con quello *parametrico-vettoriale* dell'Esempio 5.3, P. 6.

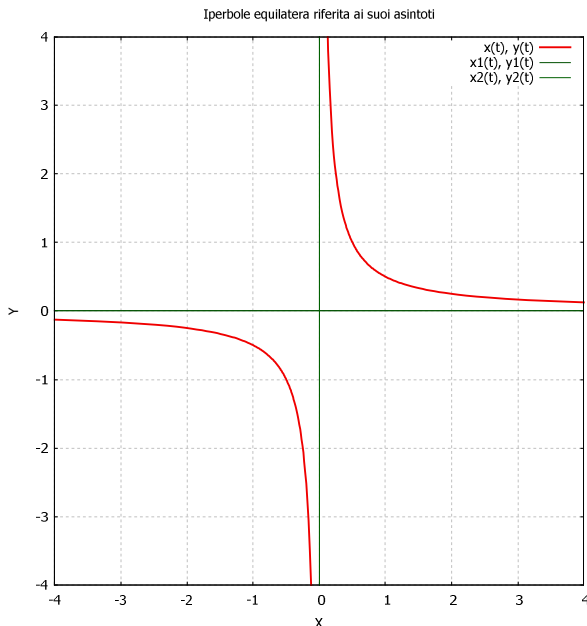
■

2. Iperbole equilatera riferita ai suoi asintoti (coordinate piano-polari)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Iperbole equilatera riferita ai suoi asintoti"
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala
#
set parametric
set trange [0.:2.*pi]; set xrange [-4.*a:4.*a]; set yrange [-4.*a:4.*a]
#
rho(t)=a/sqrt(sin(2.*t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare dell'iperbole equilatera, {x(t),y(t)}
# asintoti:
x1(t)=a*tan(t)
y1(t)=0.                   # equazione piano-polare dell'asintoto ORIZZONTALE (l'asse X), {x1(t),y1(t)}
#
x2(t)=0.
y2(t)=a/tan(t)             # equazione piano-polare dell'asintoto VERTICALE (l'asse Y), {x2(t),y2(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#005000", x2(t), y2(t) lc rgb "#005000"
#
unset parametric

```

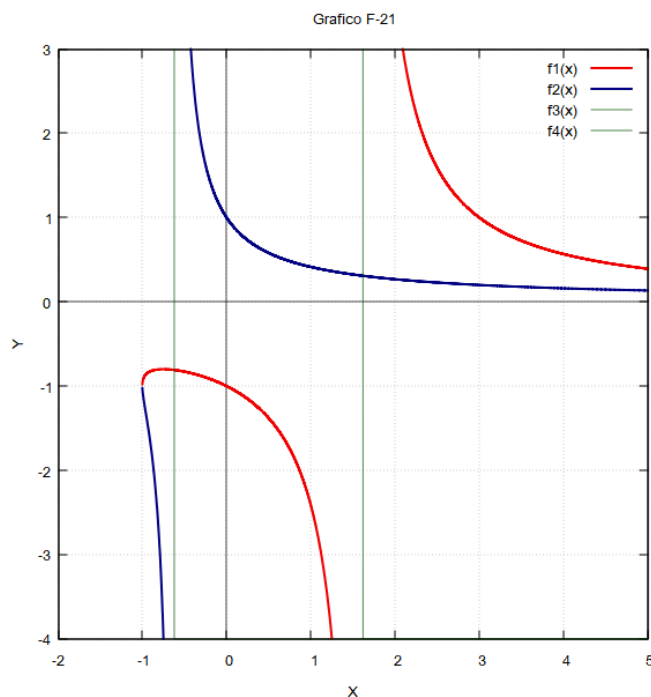


3. Grafici delle due funzioni algebriche reali definite implicitamente dall'equazione cartesiana: $(xy-1)^2 - (x+1)y^2 = 0$

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Grafico F-21"
set grid
set xlabel "X"; set ylabel "Y"
set samples 10000 # campionamento elevato necessario per il raffinamento della traccia \
                  in un intorno infinitesimo del punto di raccordo tra i grafici
set size square 1., 1.; set xrange [-2.:5.] # parametri b, h (v. p. 2)
set yrange [-4.:3.] # OBBLIGATORIO! ((utente) dipende da M, v. sotto)
# In coordinate cartesiane, un asintoto verticale, di equazione x=h (h in formato reale) può essere \
# simulato dalla funzione sgn con il comando della forma f(x)=M*sgn(x-h), essendo M il numero di \
# valore assoluto MAGGIORE in yrange. In questo caso, si ha
M=-4.
#
set yzeroaxis lt 1 lw 0.7 lc rgb "#000000"; set xzeroaxis lt 1 lw 0.7 lc rgb "#000000" # definizione degli \
                                                                                          assi del sistema di riferimento rettangolare
f1(x)=1/(x-sqrt(x+1.)) # prima funzione esplicita
f2(x)=1/(x+sqrt(x+1.)) # seconda funzione esplicita
f3(x)=M*sgn(x-(1.-sqrt(5))/2) # equazione del primo asintoto verticale
f4(x)=M*sgn(x-(1.+sqrt(5))/2) # equazione del secondo asintoto verticale
#
plot f1(x) lc rgb "#f00000" lw 2, f2(x) lc rgb "#000081" lw 2, \
     f3(x) lc rgb "#005000" lw 0.7, \
     f4(x) lc rgb "#005000" lw 0.7
# gli assi di riferimento, i rami del grafico e gli asintoti sono definibili \
# singolarmente per spessore e per colore

```

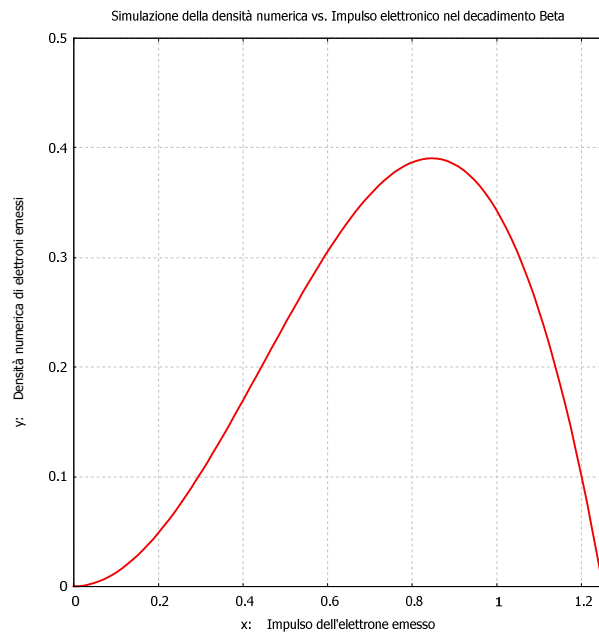


4. Simulazione della densità numerica vs. Impulso elettronico nel decadimento β

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Simulazione della densità numerica vs. Impulso elettronico nel decadimento Beta"
unset key
set grid
set xlabel "x: Impulso dell'elettrone emesso"; set ylabel "y: Densità numerica di elettroni emessi"
set xrange [0.:1.25]; set yrange [0.:0.5]
set size square 1., 1. # parametri b, h (v. p. 2)
#
k=1.5 # coefficiente di amplificazione del grafico (utente)
f(x)=k*x**2.*(sqrt(1.8125)-sqrt(x**2.+0.25))
#
plot f(x) lc rgb "#f00000" lw 2

```

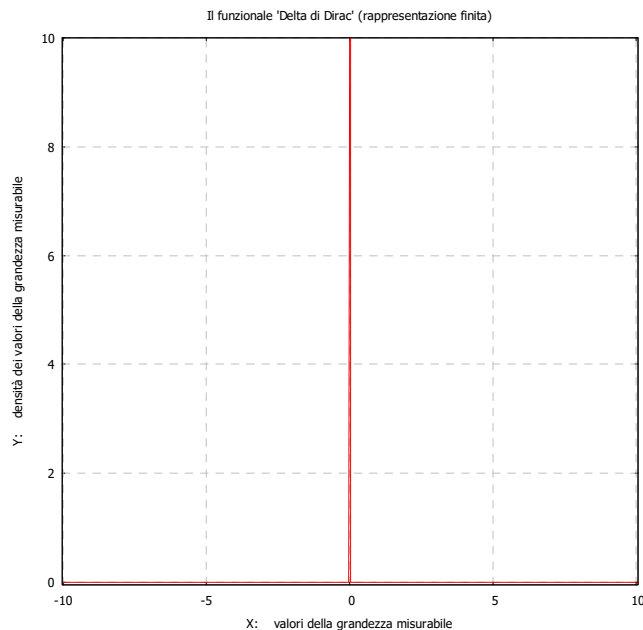


5. Rappresentazioni finite del funzionale 'Delta di Dirac'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Il funzionale 'Delta di Dirac' (rappresentazione finita)"
unset key
set grid
set xlabel "X:  valori della grandezza misurabile"
set ylabel "Y:  densità dei valori della grandezza misurabile"
set samples 1000
set xrange [-10.05:10.05]; set yrange [-0.05:10.]
set size square 1., 1. # parametri b, h (v. p. 2)
#
k=0.01
# attivare la forma finita scelta togliendo il blocco # . \
Le valutazioni grafiche non sono solo estetiche (quindi, soggettive) ma riflettono, anche, \
la percezione personale dell'autore sulla qualità del modello matematico approssimante.
# f(x)=(1/(k*sqrt(pi)))*exp(-(x/k)**2.) # eccellente
# f(x)=(1/(2.*k))*exp(-abs(x)/k) # ottima
# f(x)=(k/pi)/(x**2.+k**2.) # molto buona
# f(x)=(k/pi)*(sin(x/k)/x)**2. # discreta
# f(x)=sin(x/k)/(pi*x) # modesta
#
plot f(x) lc rgb "#f00000" # se assente, il comando lw 1 è di default

```

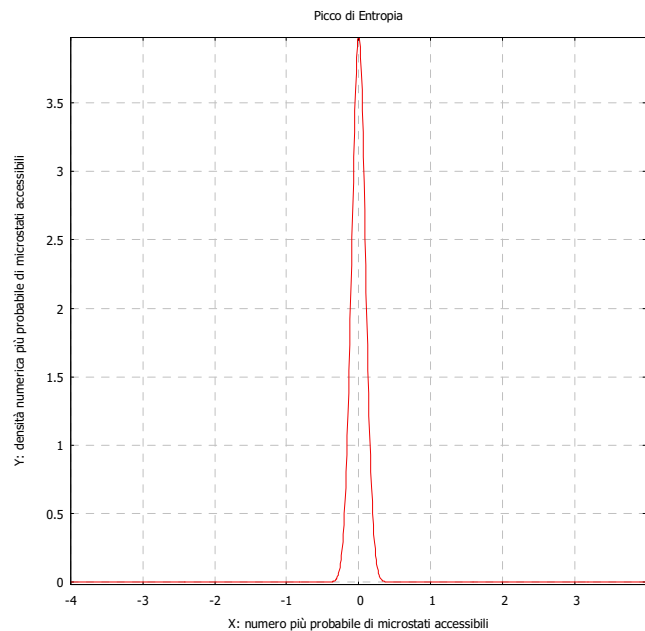


6. Picco di Entropia (MB-densità normale di microstati)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Picco di Entropia"
unset key
set grid
set xlabel "X: numero più probabile di microstati accessibili"
set ylabel "Y: densità numerica più probabile di microstati accessibili"
# set xtics 1.; set ytics 0.1
set samples 1000
set size square 1., 1. # parametri b, h (v. p. 2)
set xrange [-4.:4.]; set yrange [-0.02:3.98]
#
b=53.312
k=4. # coefficiente di amplificazione (utente)
f(x)=k*exp(-(b-0.5*log(abs(x)))*x**2.) # (†)
#
plot f(x) lc rgb "#f00000"

```



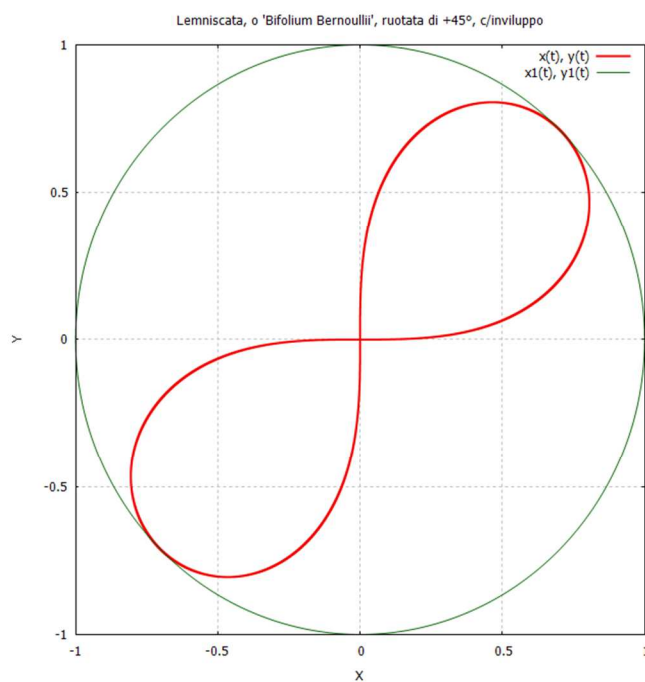
(†) Si veda il phys-notebook: Il Modello Statistico semi-classico del Gas Ideale, P.16-17.

7. Lemniscata, o 'Bifolium Bernoullii' (Jakob), ruotata di $\pi/4$

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Lemniscata, o 'Bifolium Bernoullii', ruotata di +45°, c/inviluppo"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 400000
set size square 1., 1.          # parametri b, h (v. p. 2)
a=1.                             # parametro di scala (:= rho_max)/sqrt(2)
set trange [0.:2.*pi]; set xrange [-a:a]; set yrange [-a:a]
#
rho(t)=a*sqrt(sin(2.*t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)              # equazione piano-polare della Lemniscata, {x(t),y(t)}
#
x1(t)=a*cos(t)
y1(t)=a*sin(t)                 # equazione piano-polare dell'inviluppo circolare, {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#005000"
#
unset parametric

```



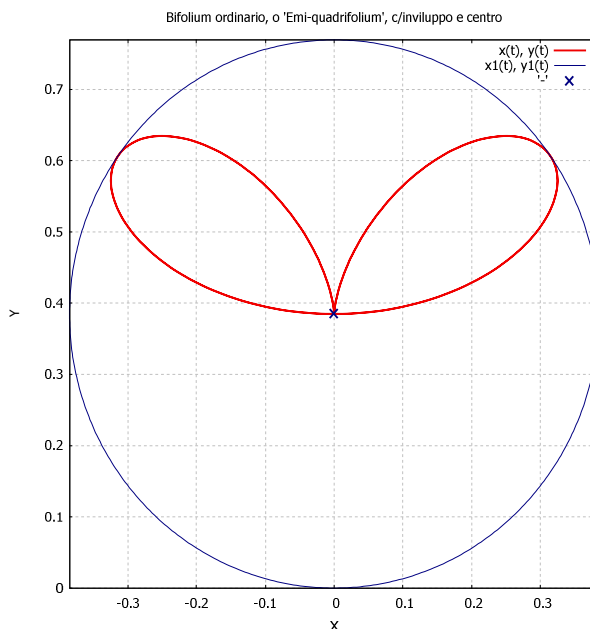
8. Bifolium ordinario o Emi-quadrifolium

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Bifolium ordinario, o 'Emi-quadrifolium', c/centro e inviluppo"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.
a=1.
set trange [0.:2.*pi]
set xrange [-2.*sqrt(3)/9*a:2*sqrt(3)/9*a]
set yrange [0.:4*a*sqrt(3)/9]
#
rho(t)=a*(cos(t))**2.*sin(t)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)+2.*(sqrt(3)/9.)*a
#
x1(t)=2*(sqrt(3)/9.)*a*cos(t)
y1(t)=2*(sqrt(3)/9.)*a*(sin(t)+1.)
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081", '-' w p ls 2 lc rgb "#000081" lw 2
0.      0.3849*a
#
e
#
unset parametric

```

parametri b, h (v. p. 2)
parametro di scala $(:= (3\sqrt{3})/2) \times \text{rho_max}$
$2\sqrt{3}/9 \times a = \text{rho_max}$
equazione piano-polare del 'Bifolium', $\{x(t), y(t)\}$
inviluppo circolare del 'Bifolium'
centro del 'Bifolium' ($y = 2\sqrt{3}/9 = 0.3849$ approx), \ coordinate tabulate in formato reale-decimale e \ posposte al comando 'plot' (v. script 9, p. 18)

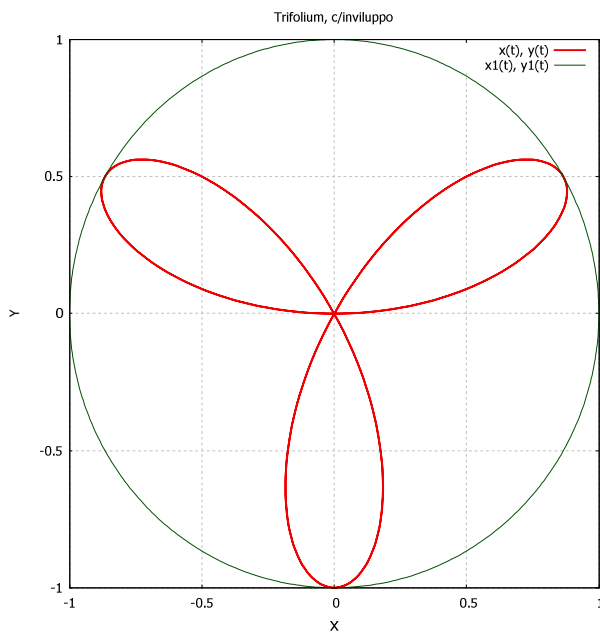


9. Trifolium

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Trifolium, c/inviluppo"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
#
a=1.                        # parametro di scala (raggio dell'involuppo circolare)
set trange [0.:2.*pi]; set xrange [-a:a]; set yrange [-a:a]
#
rho(t)=a*sin(3.*t)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare del 'Trifolium', {x(t),y(t)}
#
x1(t)=a*cos(t)
y1(t)=a*sin(t)             # involuppo circolare del 'Trifolium', {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#005000"
#
unset parametric

```

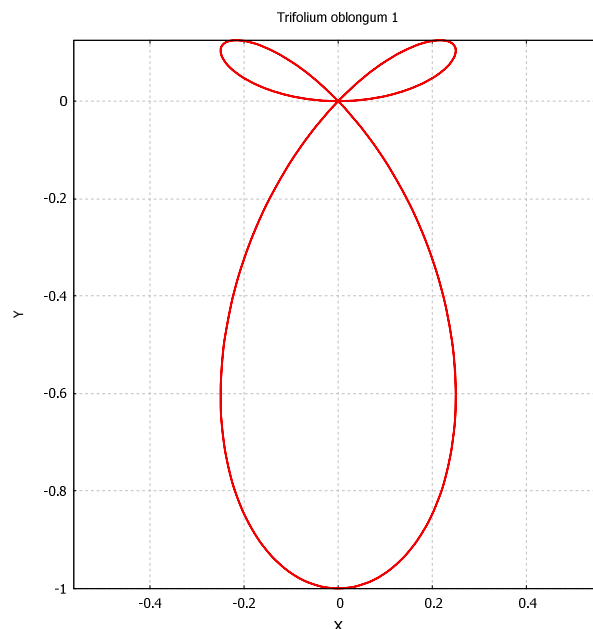


10. Trifolium oblongum (Trifoglio allungato)

Sono qui presentate tre realizzazioni elementari del 'Trifolium oblongum', i.e., quelle con una delle tre foglie ingigantita simmetricamente vs. il proprio asse. Le prime due realizzazioni, geometricamente congruenti, risultano ruotate di $\pi/2$ intorno al *nodo* (doppio), il *polo*, reciprocamente; la terza realizzazione è generata prendendo come *polo* il punto sull'asse di simmetria della macro-foglia che si trova alla distanza a (parametro di scala) dal suo punto di inversione.

La prima realizzazione possiede la rappresentazione cartesiana implicita $(x^2 + y^2)^2 + y(y^2 - x^2) = 0$:

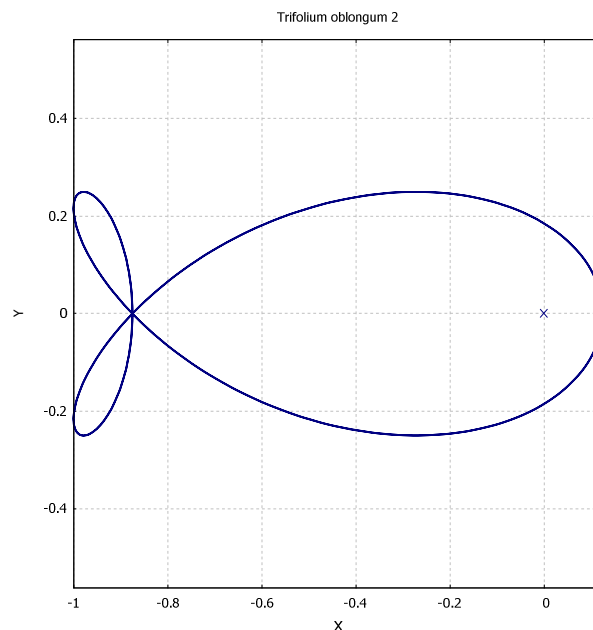
```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Trifolium oblongum 1"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (distanza tra il polo (nodo doppio), (0;0), e il punto (0;-1)), \
                             di minimo relativo (e di inversione) della curva
set xrange [-9.*a/16:9.*a/16]; set yrange [-a:a/8.]
#
rho(t)=a*sin(t)*cos(2*t)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare del 'Trifolium oblongum 1', {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2      # sono assegnati il colore e lo spessore della linea
unset parametric
```



```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Trifolium oblongum 2"      # (x = posizione dell'origine del riferimento cartesiano)
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1.0, 1.0      # parametri b, h (v. p. 2)
a=1.      # parametro di scala (distanza tra il polo (nodo doppio), (-7a/8;0), e il punto di \
          inversione della curva, (a/8;0)
set xrange [0.:2.*pi]; set xrange [-a:a/8.]; set yrange [-9.*a/16:9.*a/16]
#
rho(t)=a*cos(t)*cos(2.*t)
x(t)=rho(t)*cos(t)-7.*a/8
y(t)=rho(t)*sin(t)      # equazione piano-polare del 'Trifolium oblongum 2', {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#000081" lw 2, '-' w p ls 2 lc rgb "#000081" lw 1
# grafico c/visualizzazione dell'origine (punto isolato)
0.      0.      # coordinate cartesiane dell'origine, tabulate in formato reale-decimale e \
                posposte al comando 'plot'
e
unset parametric

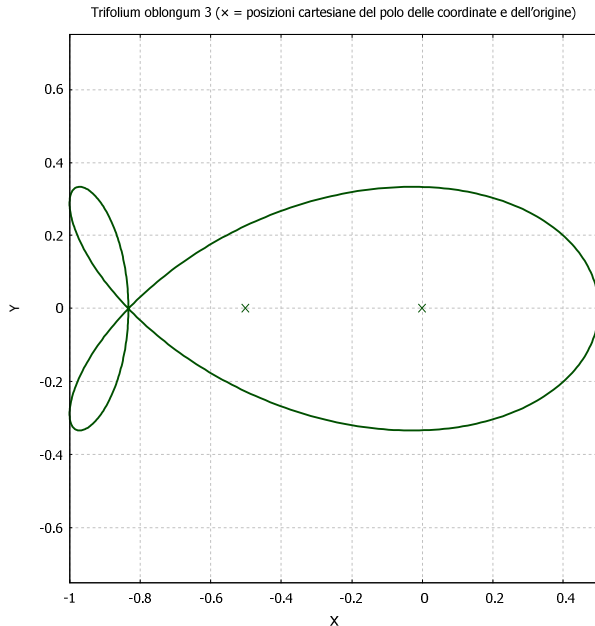
```



```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Trifolium oblongum 3 (x = posizioni cartesiane del polo delle coordinate e dell'origine)"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (distanza tra il polo, (-a/2;0), e il punto \
                             (a/2;0), di inversione della curva)
set xrange [0.:2.*pi]; set xrange [-a:a/2.]; set yrange [-3.*a/4:3.*a/4]
#
x(t)=(a/3.)*(2.*cos(t)+cos(2.*t)-3./2)
y(t)=(a/3.)*sin(2.*t)        # equazione piano-polare del 'Trifolium oblongum 3', {x(t),y(t)}, \
                             con il polo in (-a/2;0) e il nodo in (-5a/6;0)
#
plot x(t), y(t) lc rgb "#005000" lw 2, \
      '-' w p ls 2 lc rgb "#005000" lw 1, '-' w p ls 2 lc rgb "#005000" lw 1
# grafico c/visualizzazione del polo e dell'origine (punti isolati)
# coordinate cartesiane del polo, tabulate in formato reale-decimale e \
posposte al comando 'plot'
-0.5    0.
e
0.      0.
# coordinate cartesiane dell'origine, tabulate in formato reale-decimale e \
posposte al comando 'plot'
e
unset parametric

```

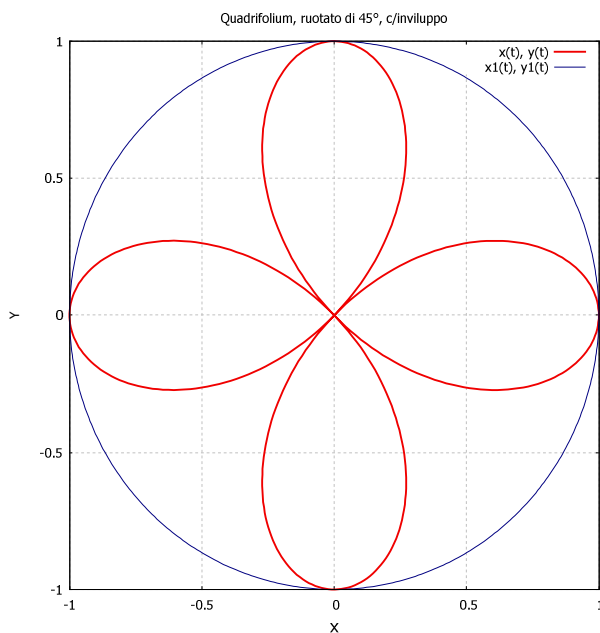


11. Quadrifolium (ruotato di $\pi/4$ rad)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Quadrifolium, ruotato di 45°, c/inviluppo"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (raggio dell'inviluppo circolare)
set trange [0.:2.*pi]; set xrange [-a:a]; set yrange [-a:a]
#
rho(t)=a*sin(2.*(t+pi/4.))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare del 'Quadrifolium', {x(t),y(t)}
#
x1(t)=a*cos(t)
y1(t)=a*sin(t)             # inviluppo circolare del 'Quadrifolium', {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

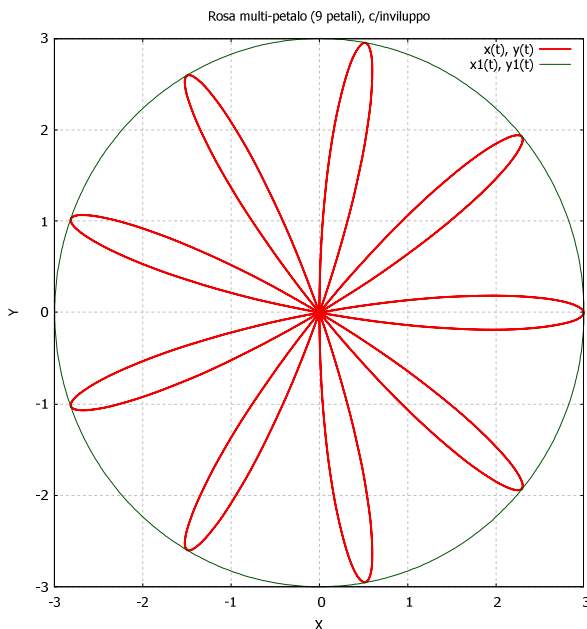


12. Rosa multi-petalo

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Rosa multi-petalo (9 petali), c/inviluppo"
# unset key
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
#
a=3.      # parametro di scala (raggio dell'involuppo circolare)
set parametric
set trange [0.:2.*pi]; set xrange [-a:a]; set yrange [-a:a]
      # Assegnato n > 1, intero in formato reale (e.g., n = 9.), \
      per n pari\dispari, escludere il rho(t) non-appropriato con l'interruttore '#'
n=9.
# rho(t)=a*sin(n*t)      # n pari:      viene generata una rosa con 2n petali
rho(t)=a*cos(n*t)      # n dispari:   viene generata una rosa con n petali
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)      # equazione piano-polare della 'Rosa multi-petalo', {x(t),y(t)}
#
x1(t)=a*cos(t)
y1(t)=a*sin(t)          # involuppo circolare della 'Rosa multi-petalo', {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#005000"
#
unset parametric

```



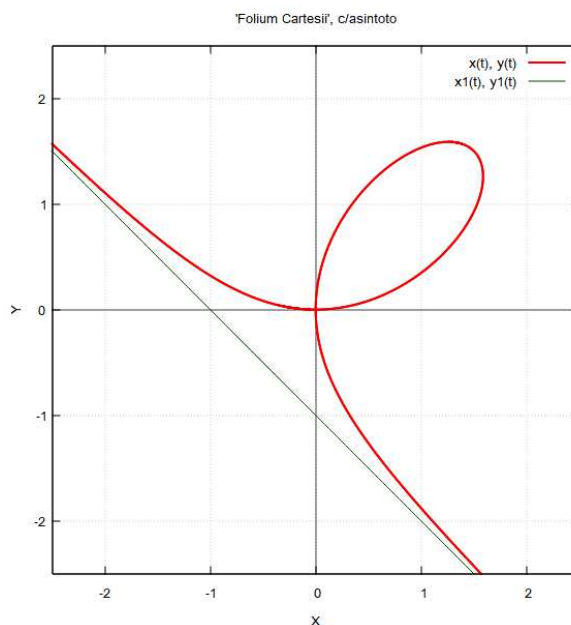
13. 'Folium Cartesii' ('Foglia' di Descartes), c/asintoto
(equazione cartesiana simmetrica in forma implicita: $x^3 + y^3 - 3axy = 0$)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Folium Cartesii', c/asintoto"
# unset key
set grid
set yzeroaxis lt 1 lw 0.7 lc rgb "#000000"; set xzeroaxis lt 1 lw 0.7 lc rgb "#000000"
set xlabel "X"; set ylabel "Y"
set samples 5000
set size square 1., 1. # parametri b, h (v. p. 2)
a=1. # parametro di scala (distanza tra l'origine (il polo nodale) \
    e il punto (0;q) di intersezione tra l'asintoto e l'asse Y)

set parametric
set trange [0.:2.*pi]; set xrange [-5./2*a:5./2*a]; set yrange [-5./2*a:5./2*a]
#
rho(t)=3./2*a*sin(2.*t)/((cos(t))**3.+(sin(t))**3.)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t) # equazione piano-polare del 'Folium Cartesii', {x(t),y(t)}, con t = phi
#
rho1(t)=-a/(cos(t)+sin(t)) # m = -1, q = -a (v. equazione piano-polare di una retta con q ≠ 0, \
    Esempio 1, p. 32)
x1(t)=rho1(t)*cos(t) # asintoto (obliquo), {x1(t),y1(t)}
y1(t)=rho1(t)*sin(t)
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#005000"
#
unset parametric

```

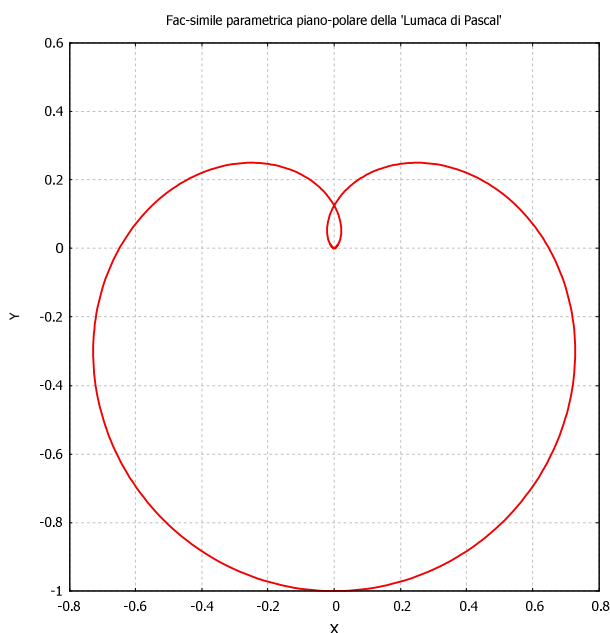


14. Fac-simile parametrica piano-polare della 'Lumaca di B. Pascal'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Fac-simile parametrica piano-polare della 'Lumaca di Pascal'"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (distanza max tra il vertice del cappio, (0;0), e i punti della curva)
set trange [0.:3.*pi]; set xrange [-0.8*a:0.8*a]; set yrange [-1.*a:0.6*a]
#
rho(t)=a*(sin(t/3.))**3.
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della curva, {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2
#
unset parametric

```

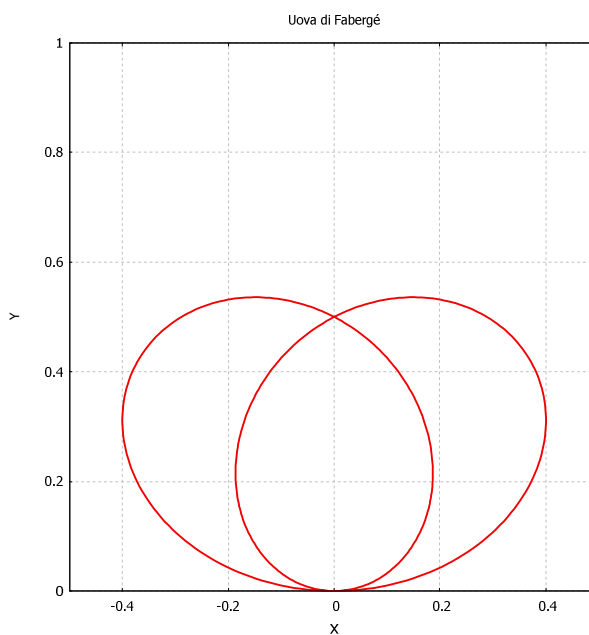


15. Uova di Fabergé

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Uova di Fabergé"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.      # parametro di scala (2×distanza dell'origine (polo) dal punto di intersezione tra gli ovali)
set trange [0.:2.*pi]; set xrange [-0.5*a:0.5*a]; set yrange [0.:a]
#
rho(t)=a*sin(t)/(2.-cos(t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della curva, {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2
#
unset parametric

```

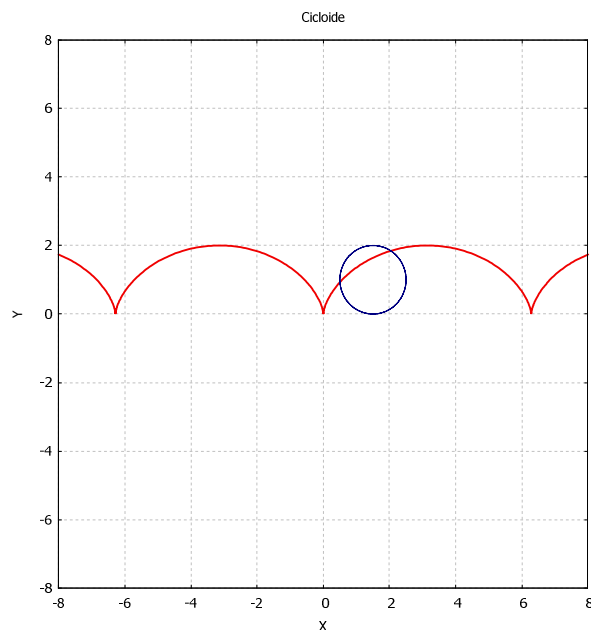


16. Cicloide

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Cicloide"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (raggio della circonferenza generatrice)
#
# # Attivare il valore appropriato di 'b' vs. 'a' (in formato reale) togliendone il blocco '#'. \
# 'b' è la distanza tra il punto variabile sulla cicloide e il centro della circonferenza generatrice. \
# Se b>a, la cicloide è 'allungata', se b=a, la cicloide è 'ordinaria', se b<a, la cicloide è 'accorciata'.
#
# # b=1.3*a
#
# b=a
#
# b=0.7*a
#
x(t)=a*t-b*sin(t)
y(t)=a-b*cos(t)             # equazione piano-polare della Cicloide, {x(t),y(t)}
#
x1(t)=a*(cos(t)+1.5)
y1(t)=a*(sin(t)+1.)         # equazione piano-polare della circonferenza generatrice, {x1(t),y1(t)}, \
                             vista con centro istantaneo in (1.5;1)
#
plot [-3.*pi*a:3.*pi*a] [-8.*a:8*a] [-8.*a:8*a] x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

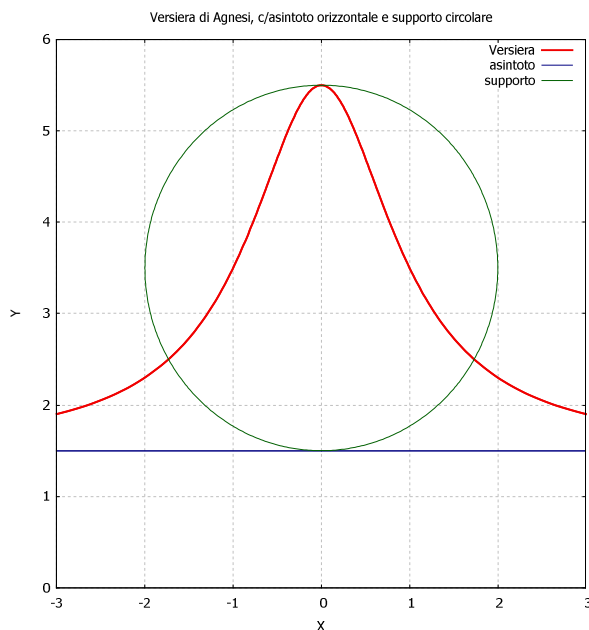


17. La 'Versiera' di M. G. Agnesi, c/asintoto e supporto

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Versiera di Agnesi, c/asintoto orizzontale e supporto circolare"
# unset key
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.      # parametro di scala (diametro della circonferenza unitaria di supporto)
b=1.5     # termine di traslazione verticale del grafico (utente)
k=4.     # coefficiente di 'aggiustamento' verticale del picco a distribuzioni di dati sperimentali (utente)
set parametric
x(t)=a/tan(t)
y(t)=k*a*(sin(t))**2.+b      # equazione piano-polare della Versiera, {x(t),y(t)}
x1(t)=a/2.*tan(t)
y1(t)=b                      # asintoto (orizzontale), {x1(t),y1(t)}
x2(t)=k*a/2.*cos(t)
y2(t)=k*a/2.*(sin(t)+1.)+b   # supporto circolare, {x2(t),y2(t)}
#
plot [-pi:pi] [-3.*a:3.*a] [0.:6.*a] x(t), y(t) lc rgb "#f00000" lw 2 title "Versiera", \
    x1(t), y1(t) lc rgb "#000081" title "asintoto", x2(t), y2(t) lc rgb "#005000" title "supporto"
#
unset parametric

```



Nota

In Fisica, la 'Versiera' trova applicazione nella rappresentazione del picco di frequenza armonica EM di assorbimento radiativo (quasi-) monocromatico e di emissione successiva, alla stessa frequenza, dell'energia assorbita da sistemi atomici e molecolari in equilibrio statistico. In regime di risonanza (o quasi-), corrispondente a frequenze di oscillazione 'in fase' di tutto il sistema, questa energia è quella *teorica* dei salti quantici di assorbimento e di emissione eseguiti dal sistema. A una versiera, si riducono le distribuzioni di Cauchy-Lorentz e di Breit-Wigner (relativistico-speciale).

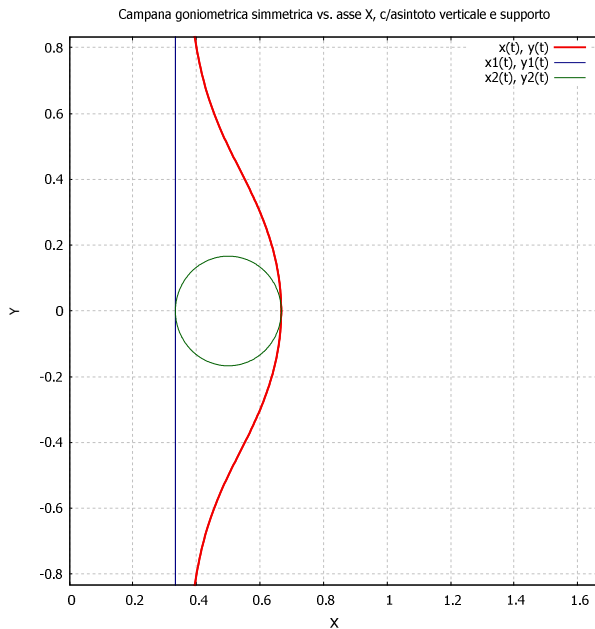
■

18. Campana goniometrica simmetrica vs. asse X, c/asintoto e supporto

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Campana goniometrica simmetrica vs. asse X, c/asintoto verticale e supporto"
# unset key
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (6×raggio del supporto circolare)
set parametric
set trange [-pi:pi]; set xrange [0.:5./3*a]; set yrange [-5./6*a:5./6*a]
#
rho(t)=a/3.*(1./(cos(t))+cos(t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della campana, {x(t),y(t)}
#
x1(t)=a/3.
y1(t)=a/(3.*tan(t))         # asintoto verticale, {x1(t),y1(t)}
#
x2(t)=a/6.*(3.+cos(t))
y2(t)=a/6.*sin(t)          # equazione del supporto circolare, {x2(t),y2(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081", x2(t), y2(t) lc rgb "#005000"
#
unset parametric

```



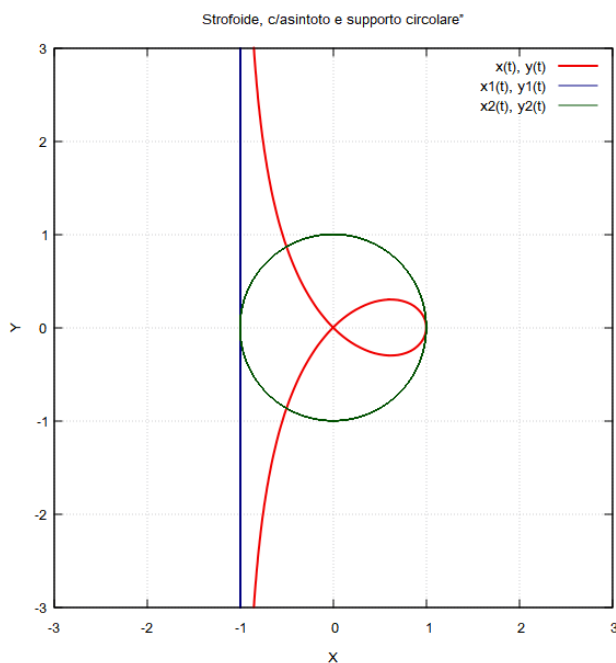
19. Strofoide, c/asintoto e supporto circolare

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Strofoide, c/asintoto e supporto circolare"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 5000
set size square 1., 1.          # parametri b, h (v. p. 2)
a=1.                            # parametro di scala (distanze nodo-asintoto e nodo-vertice del cappio)
set trange [-100.:100.]; set xrange [-3.*a:3.*a]; set yrange [-3.*a:3.*a]
#
x(t)=a*(1.-t**2)/(1.+t**2.)
y(t)=a*t*(1.-t**2)/(1.+t**2.)  # equazione piano-polare della Strofoide, {x(t),y(t)}

x1(t)=-a
y1(t)=a/tan(t)                # asintoto (verticale), {x1(t),y1(t)}
#
x2(t)=a*cos(t)
y2(t)=a*sin(t)                # equazione del supporto circolare, {x2(t),y2(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081", x2(t), y2(t) lc rgb "#005000"
#
unset parametric

```

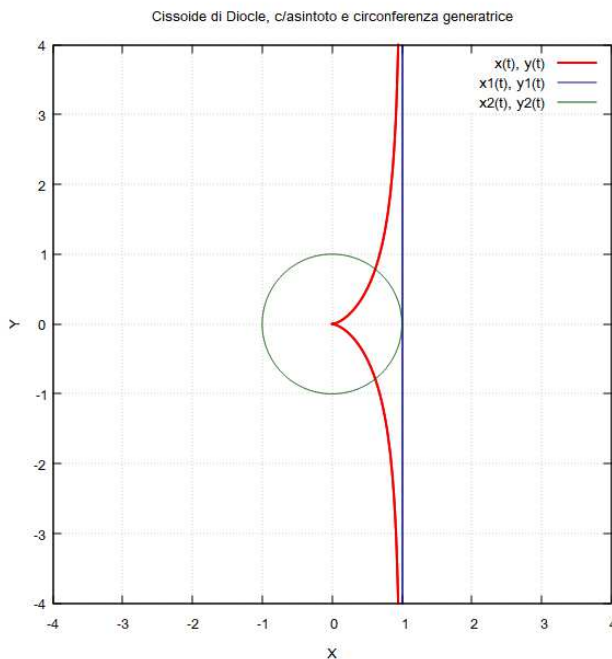


20. Cissoide di Diocle, c/asintoto e circonferenza generatrice

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Cissoide di Diocle, c/asintoto e circonferenza generatrice"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (raggio della circonferenza generatrice)
set trange [-pi:pi]; set xrange [-4.*a:4.*a]; set yrange [-4.*a:4.*a]
#
rho(t)=a*sin(t)*tan(t)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della Cissoide, {x(t),y(t)}
#
x1(t)=a
y1(t)=a/tan(t)              # asintoto (verticale), {x1(t),y1(t)}
#
x2(t)=a*cos(t)
y2(t)=a*sin(t)              # circonferenza generatrice, {x2(t),y2(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081", x2(t), y2(t) lc rgb "#005000"
#
unset parametric

```



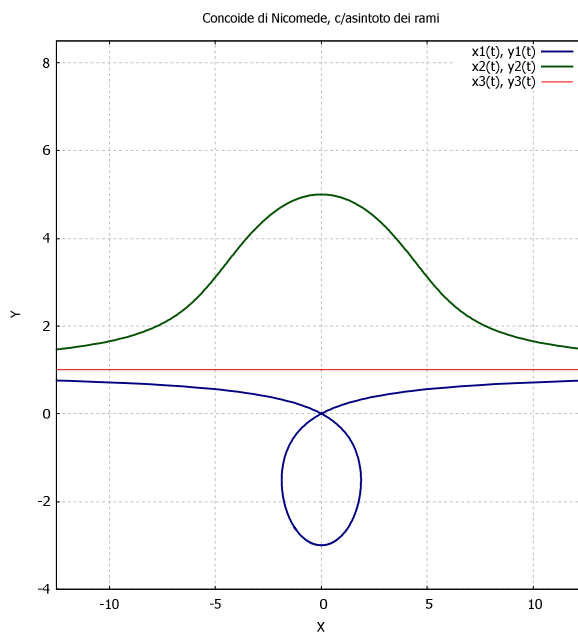
21. Concoide di Nicomede, c/asintoto dei rami

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Concoide di Nicomede, c/asintoto dei rami"
# unset key
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
set parametric
set trange [0.:pi]
#
a=1.      # parametro di scala (distanza tra i punti (0;0) e (0;a))
k=4.

      # Assegnata (utente) la coppia {a,k} in formato reale, attivare, \
      con l'interruttore '#', la coppia {xrange,yrange} appropriata
# set xrange [-4.5*a:4.5*a]; set yrange [0.:2.5*a]      # opzioni per 0<k<1
set xrange [-12.5*a:12.5*a]; set yrange [-4.*a:8.5*a]      # opzioni per k>1
#
      # algoritmi generali (i rami {x1(t),y1(t)} e {x2(t),y2(t)} sono distinti cromaticamente):
rho1(t)=a*(1./sin(t)-k)
x1(t)=rho1(t)*cos(t)
y1(t)=rho1(t)*sin(t)
#
rho2(t)=a*(1./sin(t)+k)
x2(t)=rho2(t)*cos(t)
y2(t)=rho2(t)*sin(t)
#
      # il caso k=1 è risolvibile con l'uno o l'altro algoritmo precedenti
x3(t)=a*tan(t)
y3(t)=a      # asintoto (orizzontale) dei rami della Concoide, {x3(t),y3(t)}
#
plot x1(t), y1(t) lc rgb "#000081" lw 2, x2(t), y2(t) lc rgb "#005000" lw 2, x3(t), y3(t) lc rgb "#f00000"
#
unset parametric

```

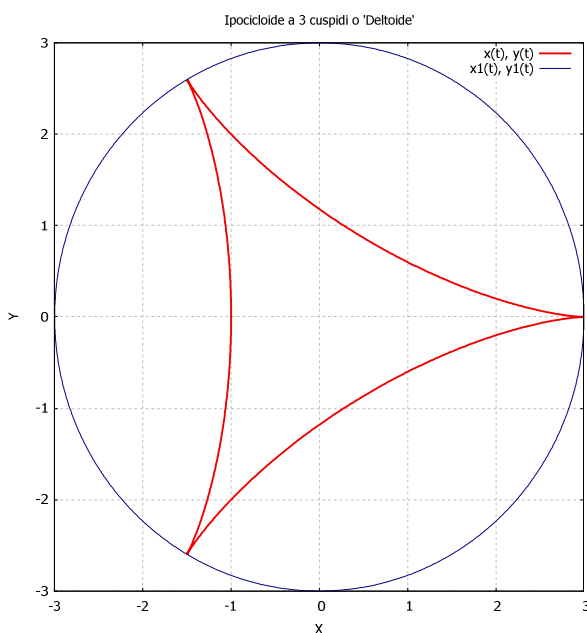


22. Ipocicloide a 3 cuspidi o 'Deltoide'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Ipocicloide a 3 cuspidi o 'Deltoide'"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 1000
set size square 1., 1.          # parametri b, h (v. p. 2)
a=1.          # parametro di scala (raggio della circonferenza generatrice)
set trange [0.:2.*pi]; set xrange [-3.*a:3.*a]; set yrange [-3.*a:3.*a]
#
rho(t)=a
x(t)=2.*a*cos(t)+a*cos(2.*t)
y(t)=2.*a*sin(t)-a*sin(2.*t)      # equazione piano-polare della 'Deltoide', {x(t),y(t)}
rho1(t)=3.*a
x1(t)=rho1(t)*cos(t)
y1(t)=rho1(t)*sin(t)            # involuppo circolare della 'Deltoide', {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

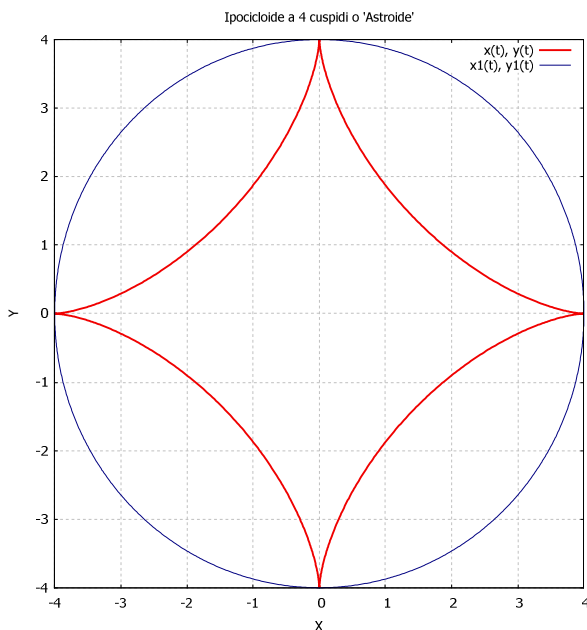


23. Ipocicloide a 4 cuspidi o 'Astroide' (involuppo di ellissi)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Ipocicloide a 4 cuspidi o 'Astroide'"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 1000
set size square 1., 1.          # parametri b, h (v. p. 2)
a=1.                            # parametro di scala (raggio della circonferenza generatrice)
set trange [0.:2.*pi]; set xrange [-4.*a:4.*a]; set yrange [-4.*a:4.*a]
#
rho(t)=a
x(t)=(4.*rho(t)*cos(t)**2.)*cos(t)
y(t)=(4.*rho(t)*sin(t)**2.)*sin(t)      # equazione piano-polare della 'Astroide', {x(t),y(t)}
#
rho1(t)=4.*a
x1(t)=rho1(t)*cos(t)
y1(t)=rho1(t)*sin(t)                # involuppo circolare della 'Astroide', {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

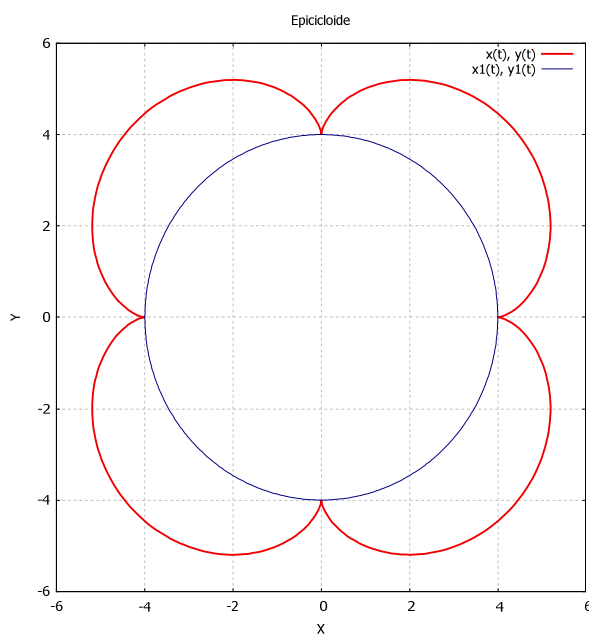


24. Epicicloide

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Epicicloide"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (raggio della circonferenza generatrice)
b=4.                        # raggio del supporto circolare fisso (utente; deve essere: b = a o b > a)
set trange [0.:2.*pi]; set xrange [-6.*a:6.*a]; set yrange [-6.*a:6.*a]
#
rho(t)=a
x(t)=(a+b)*cos(t)-a*cos((b/a+1.)*t)
y(t)=(a+b)*sin(t)-a*sin((b/a+1.)*t)      # equazione della curva algebrica, {x(t),y(t)}
#
rho1(t)=b
x1(t)=b*cos(t)
y1(t)=b*sin(t)                          # equazione del supporto circolare fisso, {x1(t),y1(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

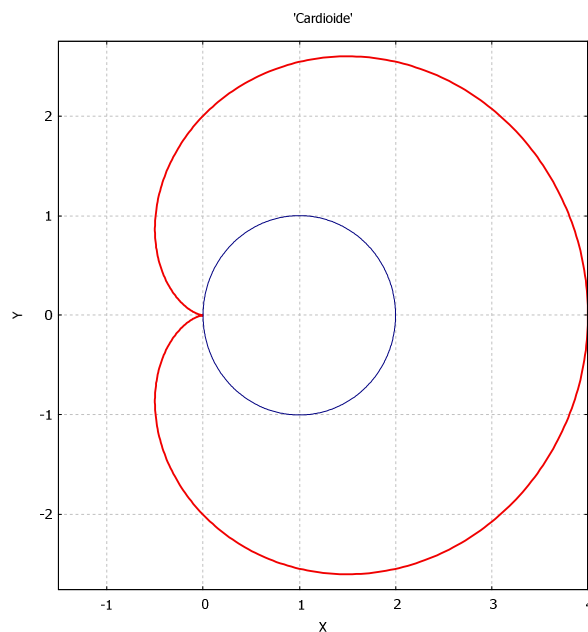


25. 'Cardioide' (o Epicicloide a una sola cuspid)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Cardioide'"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (diametro della circonferenza generatrice)
set trange [0.:2.*pi]; set xrange [-3./2*a:4.*a]; set yrange [-11./4*a:11./4*a]
#
rho(t)=2.*a*(1.+cos(t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della Cardioide, {x(t),y(t)}
#
x1(t)=a*(cos(t)+1.)
y1(t)=a*sin(t)
#
# equazione piano-polare della circonferenza generatrice, {x1(t),y1(t)}, \
# vista con centro istantaneo in (1;0)
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081"
#
unset parametric

```

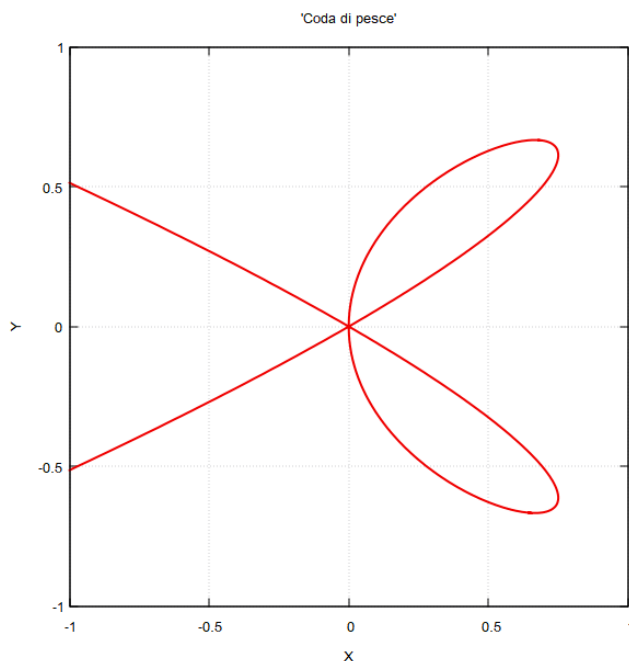


26. 'Coda di pesce' (curva quartica simmetrica vs. l'asse X,
di equazione cartesiana implicita: $x^3 - 3y^2(x - y^2) = 0$).

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Coda di pesce'"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 5000
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (utente)
set trange [-10.*a:10.*a]   # t := x/y (=cot(phi), phi è la coordinata angolare piano-polare)
set xrange [-a:a]; set yrange [-a:a]
#
x(t)=(1./3)*a*t**2.*(3.-t**2.)
y(t)=(1./3)*a*t*(3.-t**2.)   # equazione piano-polare della curva, {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2
#
unset parametric

```



27. 'Angel's wings' 2-dim (curva algebrica irrazionale in forma cartesiana implicita:

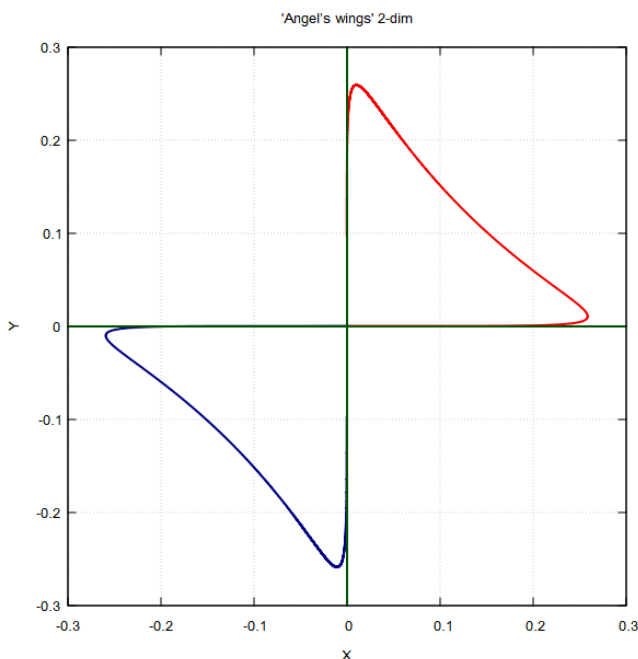
$$xy - (|x|^{1/2} + |y|^{1/2})^{12} = 0,$$

a simmetria sia assiale, vs. la bisettrice $y=x$, che centrale, vs. il punto (0;0))

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Angel's wings' 2-dim"
unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 50000
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala
set trange [0.:100.*a]      # t:=sqrt(y/x) (=sqrt(tan(phi))), phi è la coordinata angolare piano-polare)
set xrange [-0.3*a:0.3*a]; set yrange [-0.3*a:0.3*a]
#
x1(t)=a*t**(1./2)/(1.+t)**3.
y1(t)=a*t**(5./2)/(1.+t)**3.      # equazione del ramo nel 1.o quadrante, {x1(t),y1(t)}
#
x2(t)=-a*t**(1./2)/(1.+t)**3.
y2(t)=-a*t**(5./2)/(1.+t)**3.    # equazione del ramo nel 3.o quadrante, {x2(t),y2(t)}
#
x3(t)=a*tan(t)
y3(t)=0.                        # tangente di flesso orizzontale (l'asse X), {x3(t),y3(t)}
#
x4(t)=0.
y4(t)=a/tan(t)                  # tangente di flesso verticale (l'asse Y), {x4(t),y4(t)}
#
plot x1(t), y1(t) lc rgb "#f00000" lw 2, x2(t), y2(t) lc rgb "#000081" lw 2, \
     x3(t), y3(t) lc rgb "#005000", x4(t), y4(t) lc rgb "#005000"      # cf/c versione 3-dim, script 29, p. 97
#
unset parametric

```

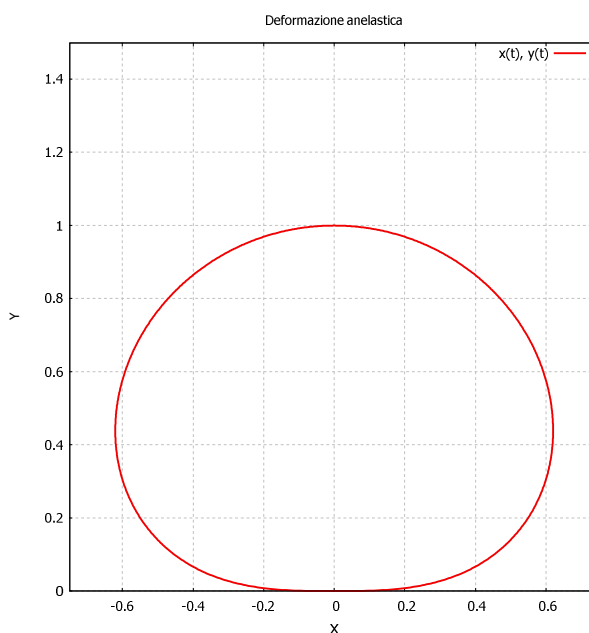


28. 'Deformazione anelastica'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Deformazione anelastica"
# unset key
set grid
set parametric
set xlabel "X"; set ylabel "Y"
set samples 1000
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (distanza tra (0;0) e il massimo della curva)
set trange [0.:pi]; set xrange [-0.75*a:0.75*a]; set yrange [0.:1.5*a]
#
rho(t)=sqrt(sin(t))
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione piano-polare della curva, {x(t),y(t)}
#
plot x(t), y(t) lc rgb "#f00000" lw 2
#
unset parametric

```



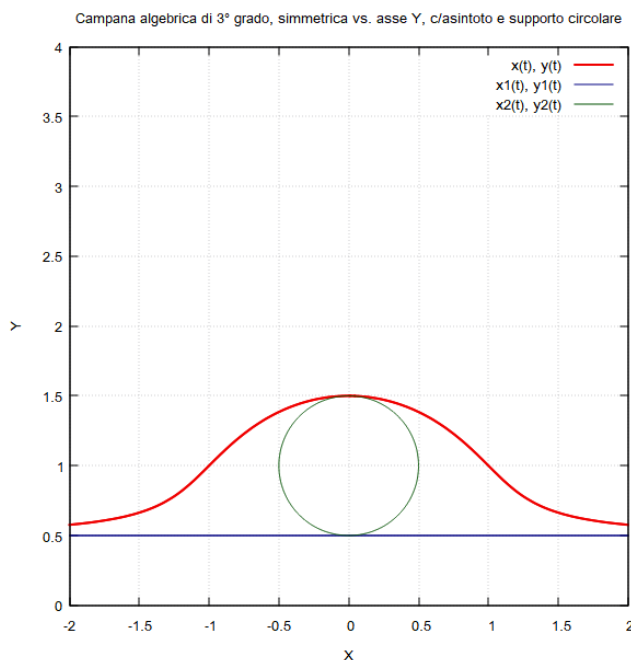
29. Campana algebrica simmetrica di 3° grado, simmetrica vs. l'asse Y, c/asintoto e supporto circolare, di equazione cartesiana in forma implicita:

$$4y(x^2 + y^2) - a^2(3y + a) = 0$$

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Campana algebrica di 3° grado, simmetrica vs. asse Y, c/asintoto e supporto circolare"
# unset key
set grid
set xlabel "X"; set ylabel "Y"
set samples 500
set size square 1., 1.      # parametri b, h (v. p. 2)
a=1.                        # parametro di scala (diametro del supporto circolare)
set parametric
set trange [-pi:pi]; set xrange [-2.*a:2.*a]; set yrange [0.:4.*a]
# Per un uso conveniente delle coordinate polari (t;rho(t)), si trasli rigidamente, di vettore (0, -a/2), \
# il vecchio sistema di riferimento cartesiano nel nuovo. L'origine nuova – il punto (0;-a/2) vs. il \
# sistema di riferimento vecchio – è scelta come polo (traslato) delle coordinate (polari).
rho(t)=a/2.*(2.*(sin(t))**2.+1.)/sin(t)
x(t)=rho(t)*cos(t)
y(t)=rho(t)*sin(t)          # equazione della curva algebrica, {x(t),y(t)}
#
x1(t)=a*tan(t)
y1(t)=a/2.                 # asintoto orizzontale, {x1(t),y1(t)} (l'asse X del vecchio sistema di riferimento!):
#
x2(t)=a/2.*cos(t)
y2(t)=a/2.*(sin(t)+2.)     # supporto circolare della Campana algebrica {x2(t),y2(t)} (cf/c scripts 20 e 19)
#
plot x(t), y(t) lc rgb "#f00000" lw 2, x1(t), y1(t) lc rgb "#000081", x2(t), y2(t) lc rgb "#005000"
#
unset parametric

```

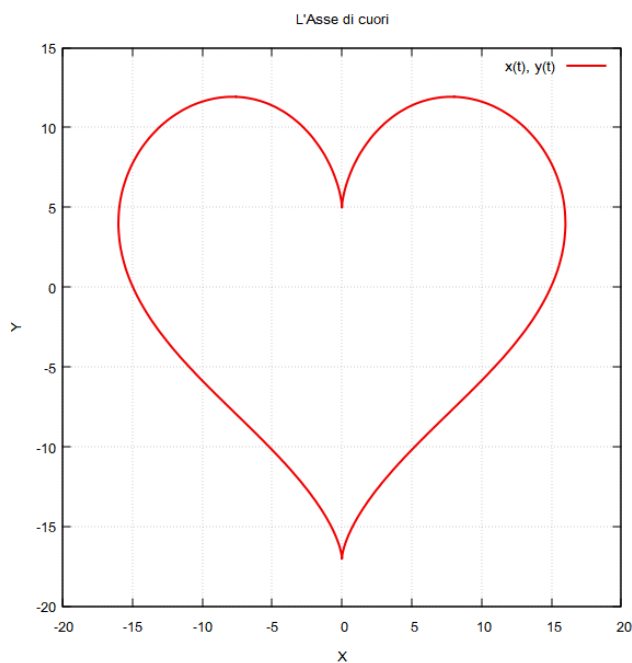


30. 'L'asse di cuori'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "L'Asse di cuori"
set xlabel "X"; set ylabel "Y"
set samples 500
set parametric
set grid
set size square 1., 1.      # parametri b, h (v. p. 2)
#
x(t)=16.*(sin(t))**3.
y(t)=13.*cos(t)-5.*cos(2.*t)-2.*cos(3.*t)-cos(4.*t)
plot [0.:2.*pi] [-20.:20.] [-20.:15.] x(t), y(t) lc rgb "#f00000" lw 2
unset parametric

```



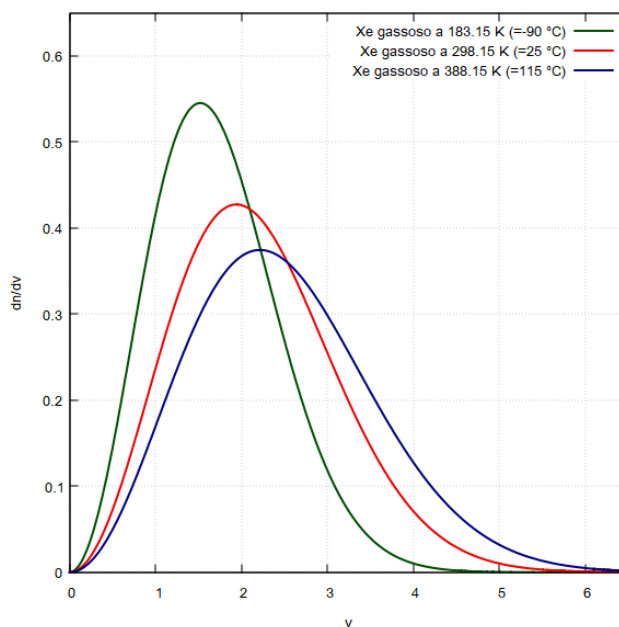
31. Densità molare di MB-probabilità della norma della velocità di atomi di Xe gassoso in equilibrio statistico a 3 temperature diverse (MB=Maxwell-Boltzmann)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "MB-densità di probabilità molare della norma della velocità di atomi di Xe gassoso in equilibrio
statistico a temperature crescenti, T1 < T2 < T3"
# unset key
set grid
set xlabel "v"; set ylabel "dn/dv"      # x=v=norma della velocità;   y = dn/dx =dn/dv = concentrazione
set samples 500
set size square 1., 1.                # parametri b, h (v. p. 2)
set xrange [0.:6.5]; set yrange [0.:0.65]
#
k=1.3806488*10**-23.                  # costante di Boltzmann [in J/K]
m=2.180*10**-21                      # massa dell'atomo di Xe [in kg]
T1=183.15                            # temperatura [in K]
T2=298.15                            #
T3=388.15                            #
f1(x)=1/sqrt(pi/2.)*(m/(k*T1))**(3./2)*x**2*exp(-m*x**2/(2.*k*T1))
f2(x)=1/sqrt(pi/2.)*(m/(k*T2))**(3./2)*x**2.*exp(-m*x**2/(2.*k*T2))
f3(x)=1/sqrt(pi/2.)*(m/(k*T3))**(3./2)*x**2.*exp(-m*x**2/(2.*k*T3))
#
plot f1(x) lc "#005000" lw 2 title "Xe gassoso a 183.15 K (= -90 °C)", \
      f2(x) lc "#f00000" lw 2 title "Xe gassoso a 298.15 K (=25 °C)", \
      f3(x) lc "#000081" lw 2 title "Xe gassoso a 388.15 K (=115 °C)"

```

MB-densità di probabilità molare della norma della velocità di atomi di Xe gassoso in equilibrio statistico a temperature crescenti, T1 < T2 < T3



32. Grafico della Funzione universale di Densità di Probabilità di Landau-Vavilov

La perdita media totale di energia, $-\langle dE/dx \rangle$, di un fascio di particelle cariche elementari, relativistiche o non-, dovuta alle collisioni casuali durante l'attraversamento disordinato (*straggling*) di un bersaglio rivelatore di spessore x *sottile* o *moderatamente sottile*, è descritta, *qualitativamente*, dalla funzione statistica universale di *densità-di-probabilità di Landau-Vavilov* (LV) per valori ≥ -3.5 della *lunghezza (estrapolata) di radiazione* λ ,

$$\phi_{LV}(\lambda) = \frac{1}{\pi} \int_0^{+\infty} e^{-u(\ln u + \lambda)} \sin(\pi u) du, \quad (1)$$

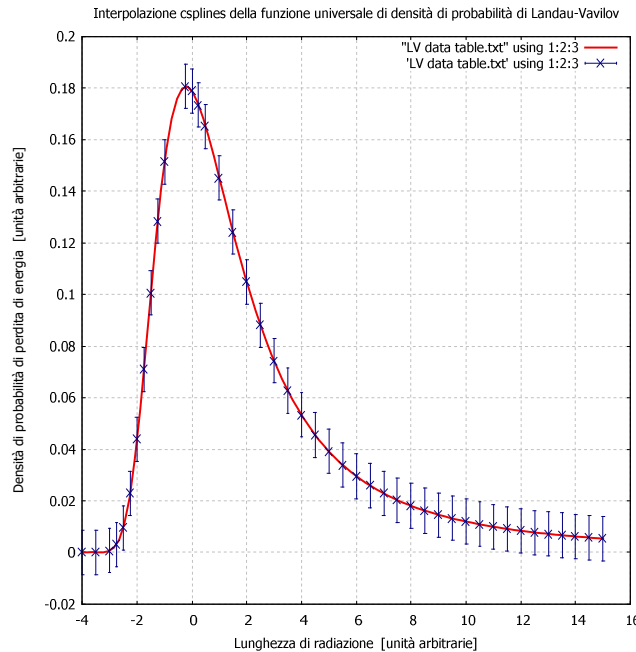
con $\langle \lambda \rangle \approx -0.225$ e $\lambda_{\max} \approx -0.223$. Il parametro λ dipende dalla micro-struttura del mezzo attraversato.

Non essendo nota alcuna forma esatta generale di $\phi_{LV}(\lambda)$, è allora necessario predisporre un file TXT di terne di valori tabulati, $\{x, y, z\} \equiv \{\lambda, \phi_{LV}(\lambda), \sigma\}$, da interpolare, e.g., con **csplines**. Qui, σ è la *deviazione standard* riferita al set di valori $\{\lambda, \phi_{LV}(\lambda)\}$. Con $\hat{\sigma}$, la *deviazione standard dal valore medio* dello stesso set, si procede in modo identico.

Per un campione di valori di $\lambda \in [-4, 15]$, uno script possibile, eseguito con GNUPLOT integrato in MAXIMA™, è

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set grid
# unset key
set title "Interpolazione csplines della funzione universale di densità di probabilità di Landau-Vavilov"
set xlabel "Lunghezza di radiazione [unità arbitrarie]"
set ylabel "Densità di probabilità di perdita di energia [unità arbitrarie]"
set size square 1., 1. # parametri b, h (v. p. 2)
plot "LV data table.txt" using 1:2:3 smooth csplines lc rgb "#f00000" lw 2, \
    'LV data table.txt' using 1:2:3 w yerrorbars lc rgb "#000081" lw 1
```

Le integrazioni *numeriche*, $\forall \lambda$ scelto, sono state eseguite con MAXIMA.



Chiaramente, la curva tracciata da csplines con un numero N sufficiente di valori di λ è *ideale*, essendo ricavata da dati sperimentali (*isolati*) di $\phi_{LV}(\lambda)$. I valori dedotti dei parametri di stima **stan_dev** (σ_y) e **stan_dev_dev** ($\hat{\sigma}_y$) (si veda, e.g., il phys-notebook PDF: **phys-crumb**s, p. 5-6) dipendono unicamente da N , dalla *precisione numerica* e dall'*efficienza* di csplines con il modello teorico (1). Qui di seguito, è applicato **stan_dev_dev**, più *raffinato*. Il grafico risultante è uno *standard di confronto* per altre distribuzioni di dati *sperimentali* e le loro analisi d'errore.

```

#      Campionamento della Funzione universale Densità di probabilità di Landau-Vavilov (LV)
#
#      Il valore y=0 del 1.o dato è assegnato arbitrariamente (valore 'vero': y = 0.2832...) per bloccare la \
#      variazione drastica locale della funzione di LV quando x < - 3.5
#
#      x              y              stan_dev_dev      ( stan_dev = 0.057)
#
-4.      0.      0.0086
-3.5     0.0001290044576  0.0086
-3.      0.000673441158  0.0086
-2.75    0.003052447858  0.0086
-2.5     0.009636888560  0.0086
-2.25    0.02296846503  0.0086
-2.      0.04398547908  0.0086
-1.75    0.07105209589  0.0086
-1.5     0.1005507520  0.0086
-1.25    0.1283984947  0.0086
-1.      0.1513919113  0.0086
-0.223   0.1806556320  0.0086
0.      0.1788541607  0.0086
0.25     0.1733196899  0.0086
0.5      0.1652322754  0.0086
1        0.1452066371  0.0086
1.5      0.1242210940  0.0086
2.      0.1049129894  0.0086
2.5      0.08822420092  0.0086
3.      0.07424765460  0.0086
3.5      0.06271978868  0.0086
4.      0.05326865722  0.0086
4.5      0.04552484015  0.0086
5.      0.03916341958  0.0086
5.5      0.03391419485  0.0086
6.      0.02955871241  0.0086
6.5      0.02592281135  0.0086
7.      0.02286847952  0.0086
7.5      0.02028649488  0.0086
8.      0.01809028394  0.0086
8.5      0.01621099451  0.0086
9.      0.01459362363  0.0086
9.5      0.01319400673  0.0086
10.     0.01197648738  0.0086
10.5    0.01091211543  0.0086
11      0.009977251885  0.0086
11.5    0.009152485625  0.0086
12.     0.008421789058  0.0086
12.5    0.007771857010  0.0086
13.     0.007191586647  0.0086
13.5    0.006671666247  0.0086
14.     0.006204248381  0.0086
14.5    0.005782688828  0.0086
15.     0.005401336928  0.0086

```

Infine, trovata la cartella contenente il file TXT di questa tabella con il tasto di comando **Change directory**, si esegue immediatamente lo script (v. P. 18-19).

■

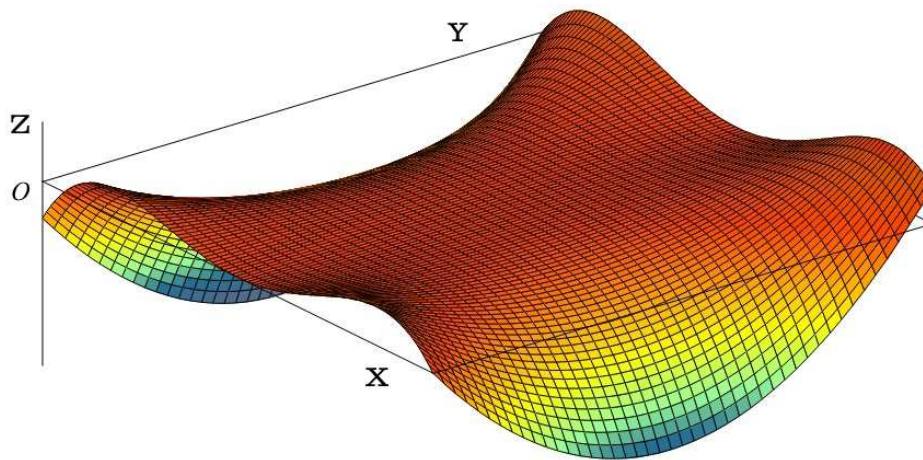
Raccolta di scripts 3-dim

1. Il 'Tappeto Volante'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Il 'Tappeto Volante' (sella di funzione)"
unset key # nasconde l'espressione matematica del grafico
set xyplane at 0.
# set xlabel "X"; set ylabel "Y"; set zlabel "Z"
# unset border # nasconde gli assi X, Y, Z e il piano X×Y del dominio (visualizzati di default)
unset xtics # nasconde le tacche di misura sull'asse X (visualizzate di default)
unset ytics # " " " Y "
unset ztics # " " " Z "
set samples 100, 90; set isosamples 50, 45
# set xrange [-25.:25.]; set yrange [-25.:25.]; set zrange [-14000.:6000.]
set size square 1., 1. # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set view 45, 52, 1, 1
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
unset colorbox
#
z(x,y)=x**2.*(x+y)*(y-x+2.75)
#
splot z(x,y) w pm3d

```



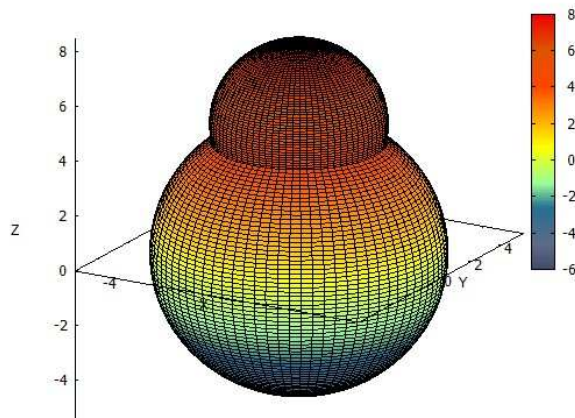
2. Intersezione tra superfici sferiche

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Intersezione tra superfici sferiche"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:pi]; set vrange [0:2.*pi]
set xrange [-5.5:5.5]; set yrange [-5.5:5.5]; set zrange [-5.5:8.5]
set samples 100, 100; set isosamples 80, 80
set size square 0.46, 1. # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
a=5. # raggio della 1.a sfera (centro in (0;0;0))
b=3. # raggio della 2.a sfera
h=4.7 # distanza del centro della 2.a sfera dall'origine
set xtics 2.; set ytics 2.; set ztics 2.
set view 75, 30, 1, 1
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
# unset colorbox
#
splot a*sin(u)*cos(v), a*sin(u)*sin(v), a*cos(u) w pm3d, \
      b*sin(u)*cos(v), b*sin(u)*sin(v), h+b*cos(u) w pm3d
#
unset parametric

```

Intersezione tra superfici sferiche



3. Studio grafico di funzione vincolata

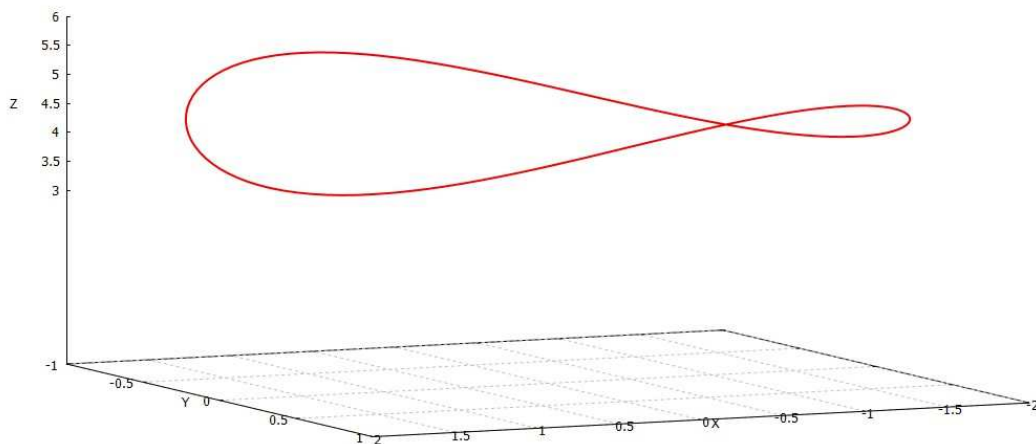
```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Studio grafico di funzione vincolata"
unset key
set grid
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 120, 120; set isosamples 100, 100
set size square 1., 1.      # parametri b, h (v. p. 2)
set view 77, 155, 1, 1
a=1.                        # parametro di scala
set parametric
set vrange [0.:2.*pi]       # parametro di linea vincolare
set xrange [-2.*a:2.*a]; set yrange [-a:a]; set zrange [3.*a:6.*a]
#
x(u,v)=2.*cos(v)            # vincolo sulla variabile x
y(u,v)=sin(v)              # vincolo sulla variabile y
z(u,v)=4.*(cos(v))**2.-cos(v)*sin(v)+5.*(sin(v))**2. # restrizione della funzione alla linea vincolare \
                           (questa proietta un'ellisse sul piano X×Y)

splot x(u,v), y(u,v), z(u,v) lc rgb "#f00000" lw 2
#
unset parametric

```

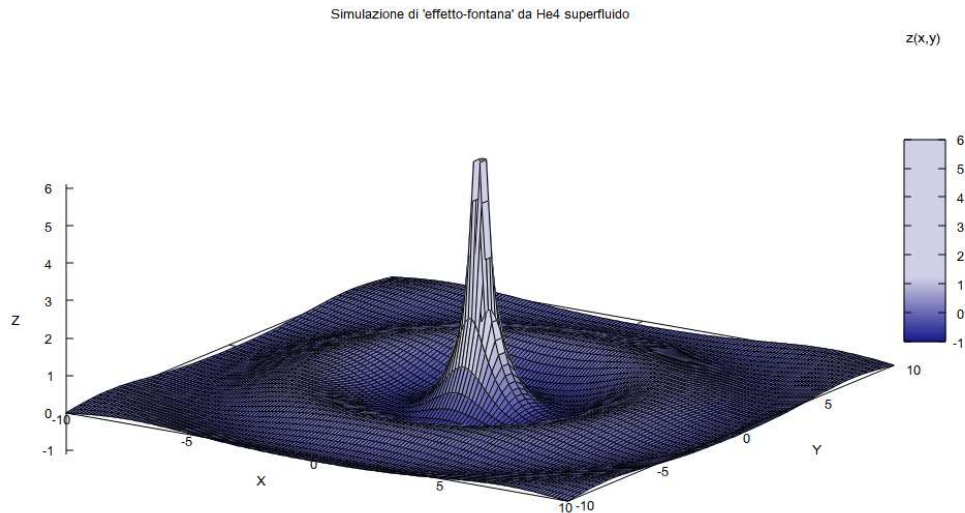
Studio grafico di funzione vincolata



4. Simulazione di effetto 'fontana' dall' He^4 superfluido

[superficie cartesiana $z = (3/2)\cos((x^2 + y^2)^{1/2})/(x^2 + y^2)^{1/2}$]

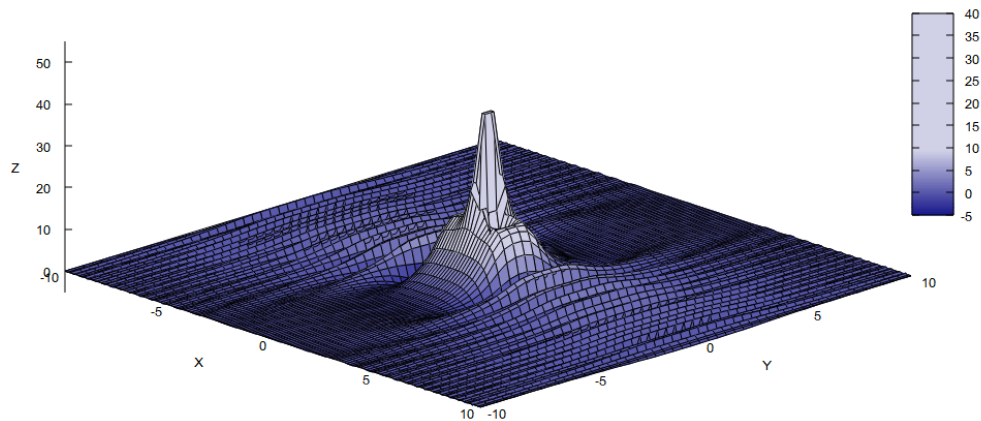
```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Simulazione di 'effetto-fontana' da He4 superfluido"
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 150, 150; set isosamples 60, 50
set xrange [-10.:10.]; set yrange [-10.:10]; set zrange [-1.1:6.1]
set size square 1.0, 1.0
set pm3d depthorder hidden3d
set view 59, 33, 1, 1
set palette defined (0 "#17178b", 7 "#d0d0e7", 22 "#d0d0e7")
# unset colorbox
#
z(x,y)=(3./2)*cos(sqrt(x**2.+y**2.))/sqrt(x**2.+y**2.)
splot z(x,y) w pm3d
```



5. Simulazione di un picco d'onda solitonica in un canale di He⁴ superfluido
 [superficie cartesiana $z = 15 \sin((x^4 + y^2)^{1/2}) / (x^2 + y^4)^{1/2}$]

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Simulazione di un picco d'onda solitonica in un canale di He4 superfluido"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 150, 150; set isosamples 60, 50
set xrange [-10.:10.]; set yrange [-10.:10.]; set zrange [-5.:55.]
set size square 1.,1. # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set view 53, 46, 1, 1
set palette defined (0 "#17178b", 7 "#d0d0e7", 22 "#d0d0e7")
# unset colorbox
#
z(x,y)=15.*sin(sqrt(x**4.+y**2.))/sqrt(x**2.+y**4.)
#
splot z(x,y) w pm3d
```

Simulazione di un picco d'onda solitonica in un canale di He⁴ superfluido



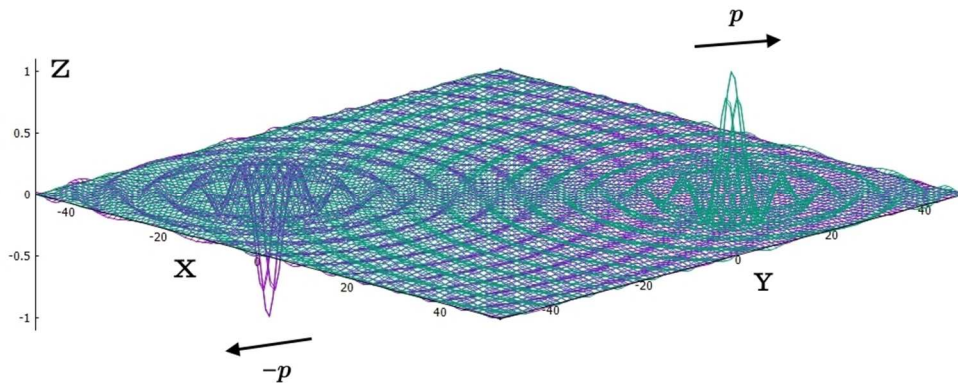
6. Simulazione di creazione di coppia fermione-antifermione (CM-riferimento)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Simulazione di creazione di coppia fermione-antifermione \
dall'annichilazione del bosone virtuale mediatore di interazione"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 120, 120; set isosamples 85, 85
set xrange [-50:50]; set yrange [-50:50]; set zrange [-1.1:1.1]
set view 57, 45, 1, 1
set size square 1., 1.
#
# parametri b, h (v. p. 2)
# onde-particelle:
z1(x,y)=-sin(sqrt((x+25.)**2.+(y+25.)**2.))/sqrt((x+25.)**2.+(y+25.)**2.)
z2(x,y)=sin(sqrt((x-25.)**2.+(y-25.)**2.))/sqrt((x-25.)**2.+(y-25.)**2.)
#
# modello: j0(r) (Bessel sferica)
#
splot z1(x,y), z2(x,y)

```

Simulazione di creazione di coppia fermione-antifermione dall'annichilazione del bosone virtuale mediatore di interazione

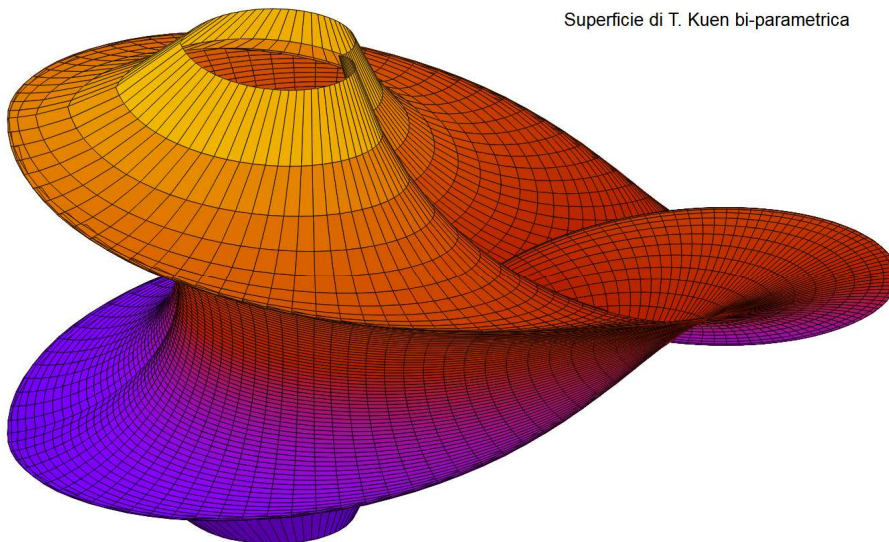


7. La Superficie di T. Kuen bi-parametrica (non-orientabile)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set label 1 "Superficie di T. Kuen bi-parametrica" at screen 0.5, 0.9
set label 1 font "frscript, 20"
set style line 3 linetype -1 linewidth 0.5
set hidden3d
set ticslevel 0.
unset xtics; unset ytics; unset ztics
unset border; unset colorbox; unset key
set lmargin at screen 0.1      # margine di sinistra
set bmargin at screen 0.1     # margine in basso
set rmargin at screen 0.9     # margine a destra
set tmargin at screen 0.9    # margine in alto
set parametric
set dummy u, v
set urange [-4.5:4.5]; set vrange [0.05:pi-0.05]
set samples 150, 150; set isosamples 100, 100
set view 57, 0, 1.35, 1.08
set pm3d depthorder hidden3d
a=1.0                        # parametro arbitrario (utente)
b=2.0                        # parametro arbitrario (utente)
#
x(u,v)=2.*a*(cos(u)+u*sin(u))*sin(v)/(1.+u**2.*(sin(v))**2.)
y(u,v)=2.*a*(sin(u)-u*cos(u))*sin(v)/(1.+u**2.*(sin(v))**2.)
z(u,v)=a*log(tan(v/2.))+b*cos(v)/(1.+u**2.*(sin(v))**2.)
#
splot x(u,v), y(u,v), z(u,v) w pm3d

```



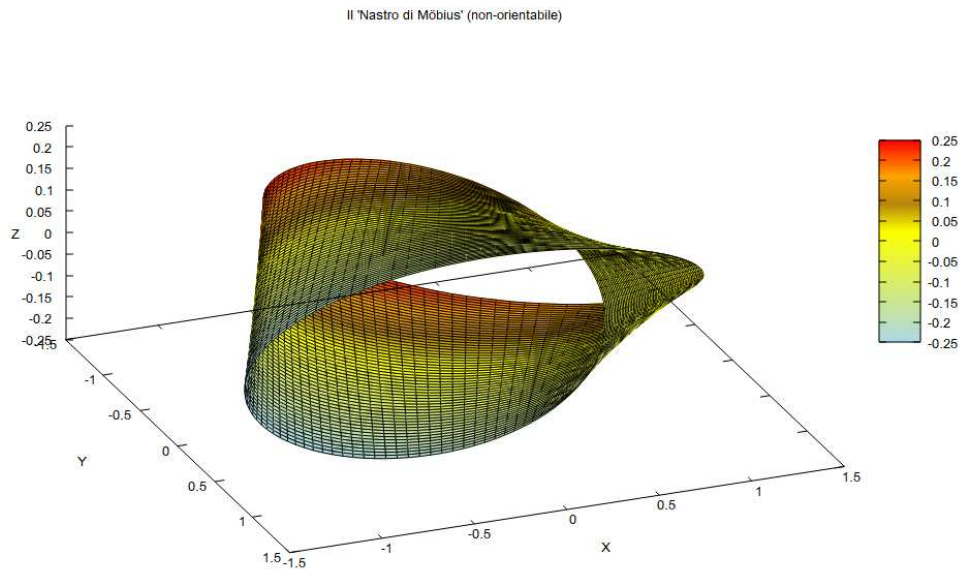
Superficie di T. Kuen bi-parametrica

8. Il 'Nastro di Möbius' (superficie non-orientabile)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Il 'Nastro di Möbius' (non-orientabile)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set ticslevel 0.
set samples 100, 100; set isosamples 60, 60
set parametric
set urange [0.:2.*pi]; set vrange [-0.5:0.5]
x(u,v)=(1.+(v/2.)*cos(u/2.))*cos(u)
y(u,v)=(1.+(v/2.)*cos(u/2.))*sin(u)
z(u,v)=(v/2.)*sin(u/2.)
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# unset colorbox
set view 137, 22, 1, 1
set size square 1., 1.      # parametri b, h (v. p. 2)
#
splot x(u,v), y(u,v), z(u,v) w pm3d
#
unset parametric

```

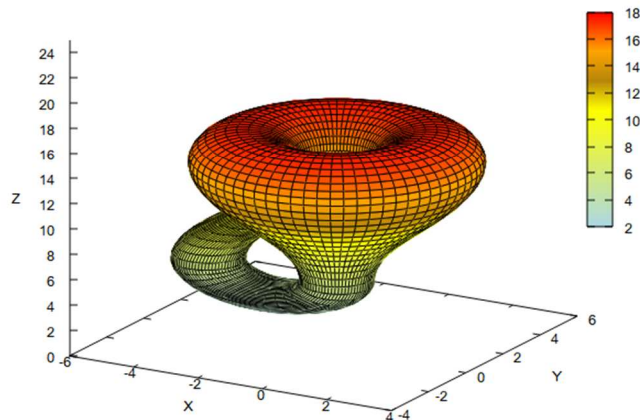


9. La 'Bottiglia di Klein' (superficie non-orientabile)

Script impegnativo e molto delicato riguardo al condizionamento delle variabili cartesiane $x(u, v)$, $y(u, v)$, $z(u, v)$.
Se necessario, consultare l'Help attentamente.

```
set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "La 'Bottiglia di Klein' (non-orientabile)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set tics 2.
set parametric
set urange[0.:2.*pi]; set vrange[0.:4.*pi]
set xrange[-6.:4.]; set yrange[-4.:6.]; set zrange [0.:25.]
set samples 70, 70; set isosamples 60, 60
set view 71, 30, 1, 1
set size square 0.6, 1. # parametri b, h (v. p. 2)
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# unset colorbox
set pm3d depthorder hidden3d
x(u,v)= v < pi ? (2.5-1.5*cos(v))*cos(u): \
        v < 2.*pi ? (2.5-1.5*cos(v))*cos(u): \
        v < 3.*pi ? -2.+(2.+cos(u))*cos(v): -2.+2.*cos(v)-cos(u)
y(u,v)= v < pi ? (2.5-1.5*cos(v))*sin(u): \
        v < 2.*pi ? (2.5-1.5*cos(v))*sin(u): \
        v < 3.*pi ? sin(u): sin(u)
z(u,v)= v < pi ? -2.5*sin(v): v < 2.*pi ? 3.*v-3.*pi: \
        v < 3.*pi ? (2.+cos(u))*sin(v)+3.*pi: -3.*v+12.*pi
k=15.
#
splot x(u,v), y(u,v), k-z(u,v) w pm3d
#
unset parametric
```

La 'Bottiglia di Klein' (non-orientabile)



10. La Superficie di Stewart (non-orientabile, in coordinate sferiche)

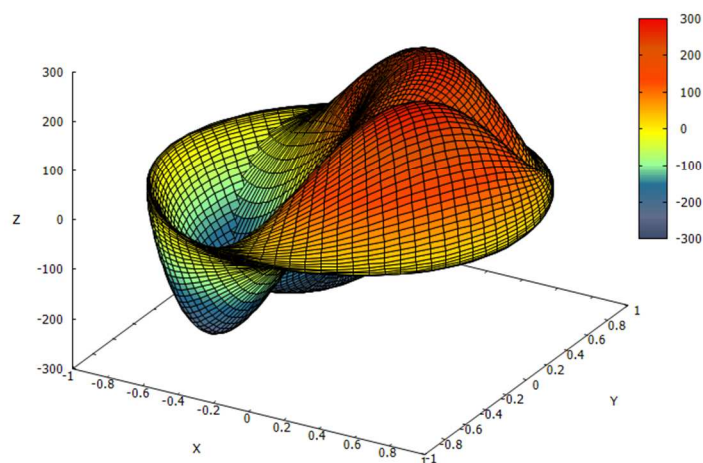
```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "La Superficie di Stewart (non-orientabile, in coordinate sferiche)"
# (da uno studio di I. Stewart sulla Bottiglia di Klein (1991))

unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set urange [0.:pi]; set vrange [0.:2.*pi]
set samples 100, 100; set isosamples 80,80
set ticslevel 0.
set parametric
x(u,v)=sin(u)*cos(v)
y(u,v)=sin(u)*sin(v)
z(u,v)=cos(u)
set view 60, 30, 1, 1
set size square 0.6, 1. # parametri b, h (v. p. 2)
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
# unset colorbox
set pm3d depthorder hidden3d
#
r=2.5
g(u,v)=(r**2.+2.*r*sin(u)*sin(v)-1.)*((r**2.-2.*r*sin(u)*sin(v)-1.)**2. \
-8.*(r*cos(u))**2.)+8.*r**2.*sin(2.*u)*cos(v)*(r**2.-2.*r*sin(u)*sin(v)-1.)
#
splot x(u,v), y(u,v), g(u,v)*z(u,v) w pm3d
#
unset parametric

```

La Superficie di Stewart (non-orientabile, in coordinate sferiche)

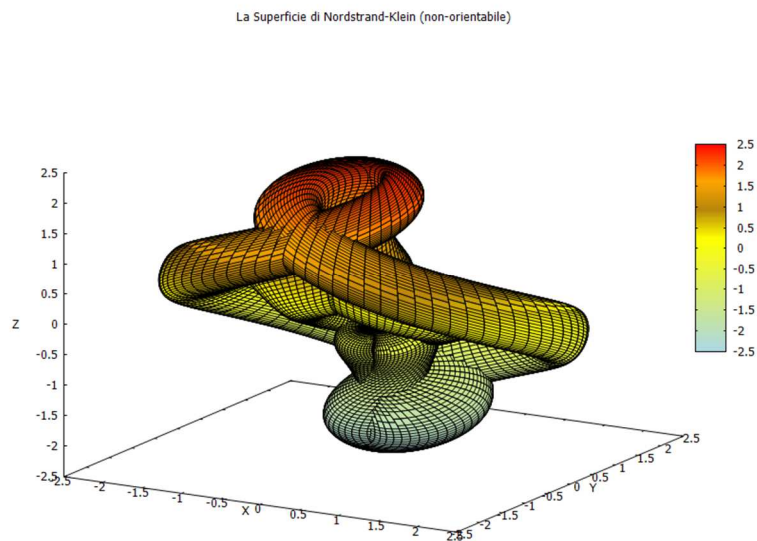


11. La Superficie di Nordstrand-Klein (non-orientabile)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "La Superficie di Nordstrand-Klein (non-orientabile)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set ticslevel 0.
set samples 150, 150; set isosamples 120, 120
set size square 0.7, 1. # parametri b, h (v. p. 2)
set parametric
set urange [0.:4.*pi] # angolo azimutale (phi) vs. il piano X×Y
set vrange [0.:4.*pi] # posizione sulla superficie
x(u,v)=(cos(u/2.)*(cos(v)+sqrt(2.))+0.5*sin(u/2.)*sin(2.*v))*cos(u)
y(u,v)=(cos(u/2.)*(cos(v)+sqrt(2.))+0.5*sin(u/2.)*sin(2.*v))*sin(u)
z(u,v)=-(cos(v)+sqrt(2.))*sin(u/2.)+0.5*cos(u/2.)*sin(2.*v)
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# unset colorbox
set view 70, 30, 1, 1
#
splot x(u,v), y(u,v), z(u,v) w pm3d
#
unset parametric

```



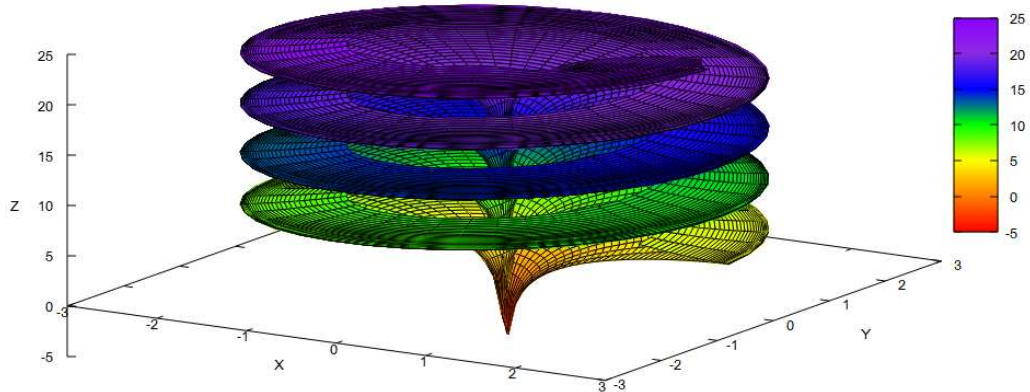
12. Superficie di Dini ('Calla della Guyana')

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie di Dini (Calla della Guyana)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:8*pi]; set vrange [0.01:2.5]
    # u=phi (coordinata cilindrica azimutale; il numero delle rivoluzioni di u è 4 (8*pi=4*(2*pi)) \
    # v=rho (coordinata cilindrica radiale)
set xrange[-3.:3.]; set yrange[-3.:3.]; set zrange [-5.:25.]
set size square 1., 1. # parametri b, h (v. p. 2)
set samples 150, 150; set isosamples 80, 80
set view 65, 32
a=2.5 # v-raggio max del grafico (cf/c vrange)
b=0.8 # passo elicoidale
c=5. # spostamento z-assiale del grafico
set palette defined (0 "#ff0000", 3 "#ff7f00", 6 "#ffff00", 9 "#00ff00", \
    12 "#0000ff", 15 "#8a2be2", 18 "#8f00ff")
set pm3d depthorder hidden3d
#
splot a*cos(u)*sin(v), a*sin(u)*sin(v), a*(cos(v)+log(tan(v/2)))+b*u+c w pm3d
#
unset parametric

```

Superficie di Dini (Calla della Guyana)

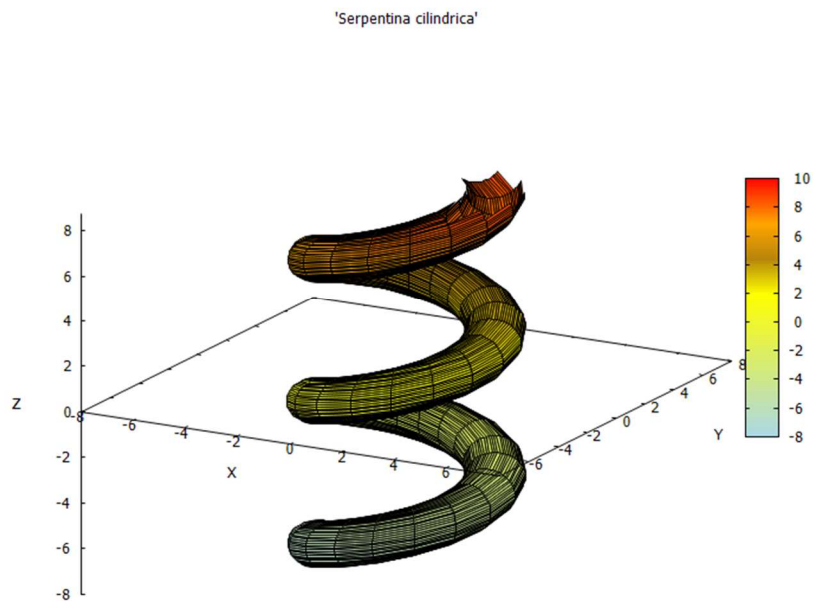


13. La superficie 'Serpentina cilindrica'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Serpentina cilindrica'"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [-3.*pi:6.*pi]; set vrange [0.:2.*pi]
set view 71, 29
set size square 0.6, 1. # parametri b, h (v. p. 2)
set samples 80, 80; set isosamples 70, 70
set xrange [-8.:8.]
set yrange [-8.:8.]
set zrange [-8.:8.755]
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
set pm3d depthorder hidden3d
#
splot (cos(v)+3.)*cos(u), (cos(v)+3.)*sin(u), sin(v)+u+2. w pm3d
#
unset parametric

```

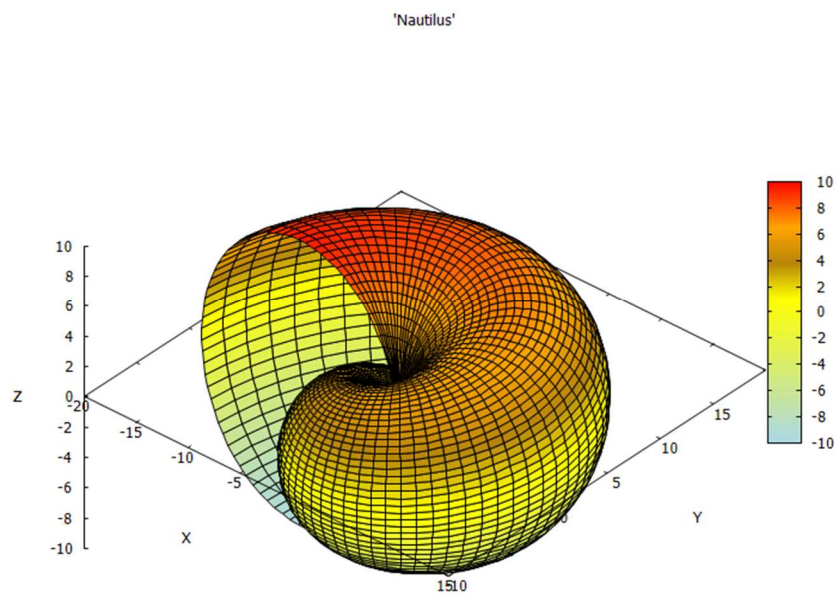


14. 'Nautilus' (spirale conica [1])

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "'Nautilus'"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:3.*pi]; set vrange [0.:2.*pi]
set view 48, 41
set size square 0.6, 1.      # parametri b, h (v. p. 2)
set samples 100, 100; set isosamples 80, 80
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
# unset colorbox
set pm3d depthorder hidden3d
#
splot u*(cos(v)+1.)*cos(u), u*(cos(v)+1.)*sin(u), u*sin(v) w pm3d
#
unset parametric

```

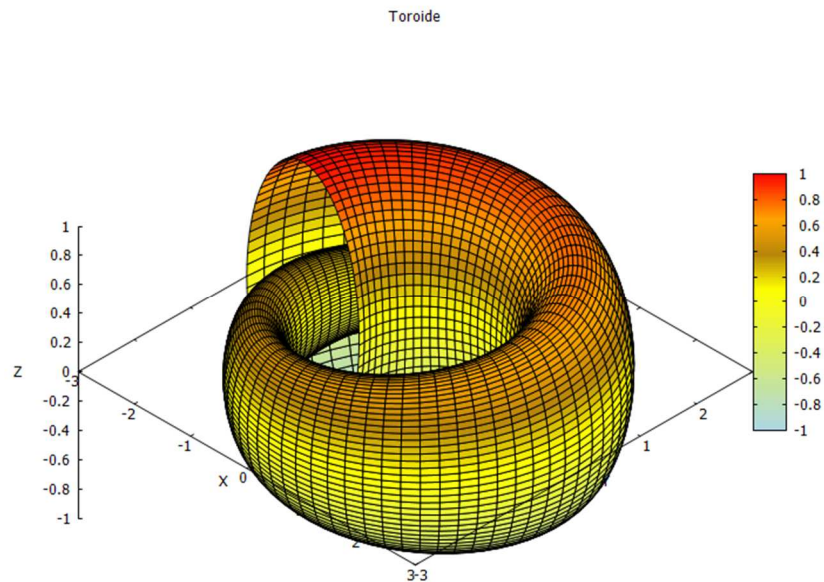


15. Toroide (spirale conica [2])

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Toroide"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:3.*pi]; set vrange [0.:2.*pi]
set view 47, 45
set size square 0.6, 1.      # parametri b, h (v. p. 2)
set samples 100, 100; set isosamples 80, 80
# unset colorbox
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
set pm3d depthorder hidden3d
#
splot (u/(3.*pi)*cos(v)+2.)*cos(u), (u/(3.*pi)*cos(v)+2.)*sin(u), u*sin(v)/(3.*pi) w pm3d
#
unset parametric

```

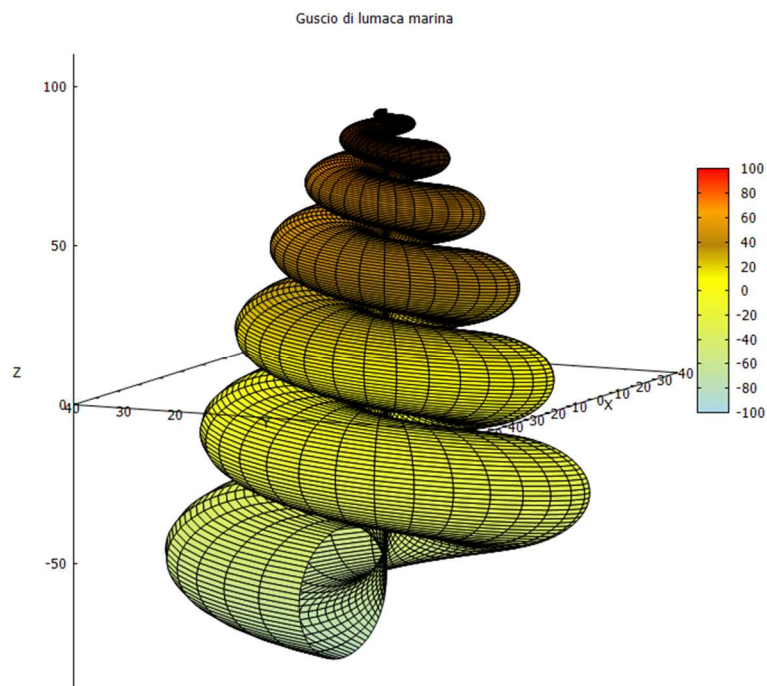


16. Il 'Guscio di lumaca marina'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Guscio di lumaca marina"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:13.*pi]; set vrange [0.:2.*pi]; set zrange [-90.:110.]
set view 80, 295, 1, 1.7
set size square 0.6, 1. # parametri b, h (v. p. 2)
set samples 200, 50; set isosamples 300, 75
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
set pm3d depthorder hidden3d
b=0.5
k=175.
#
splot b*u*(cos(v)+1.)*cos(u), b*u*(cos(v)+1.)*sin(u), b*(u*sin(v)-((u+3.)/8.*pi)**2.+k) w pm3d
#
unset parametric

```

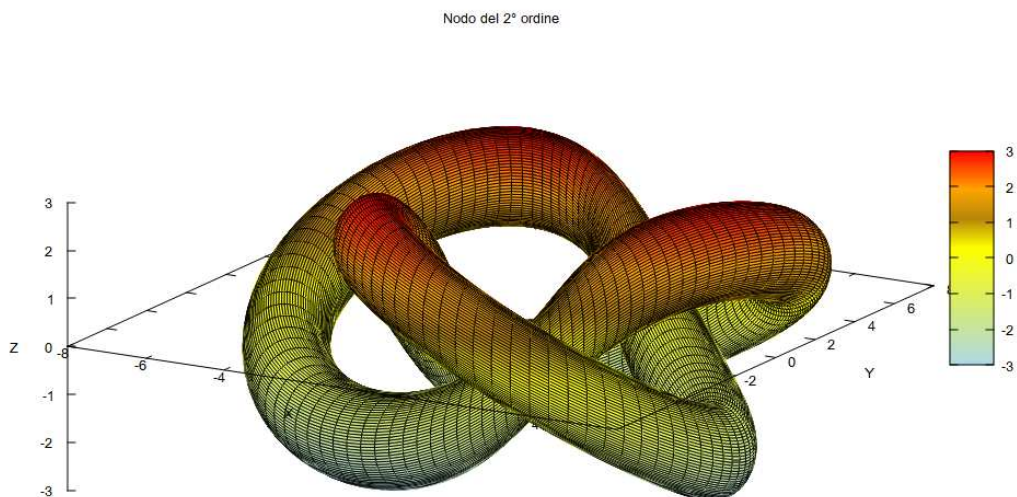


17. 'Nodo del 2° ordine'

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Nodo del 2° ordine"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [-2.*pi:2.*pi]; set vrange [0.:2.*pi]
set view 60, 30
set size square 1., 1.      # parametri b, h (v. p. 2)
set samples 200, 100; set isosamples 100, 100
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
set pm3d depthorder hidden3d
#
splot cos(u)*cos(v)+3.*(1.5+0.5*sin(1.5*u))*cos(u), \
      sin(u)*cos(v)+3.*(1.5+0.5*sin(1.5*u))*sin(u), sin(v)+2.*cos(1.5*u) w pm3d
#
unset parametric

```

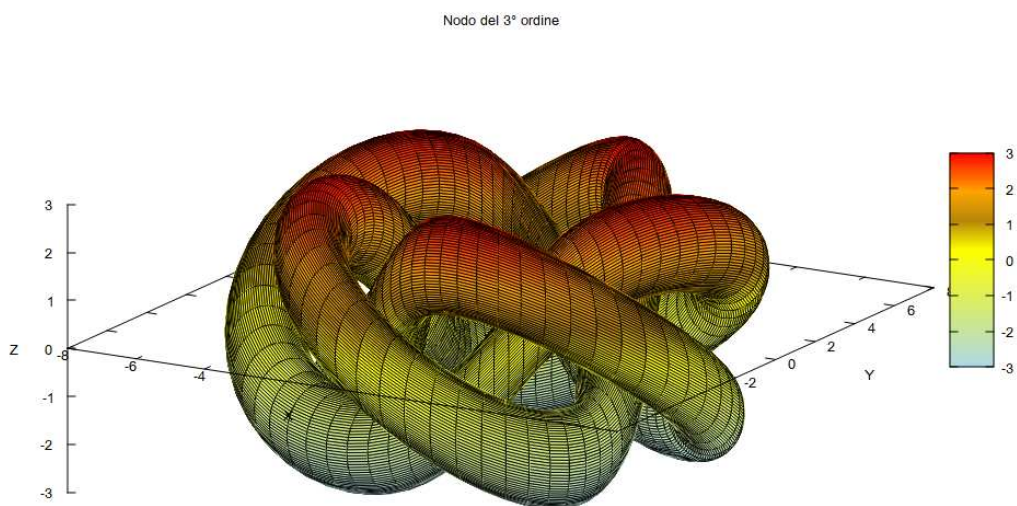


18. 'Nodo del 3° ordine'

```

set term qt enhanced size 1056, 624 font "Arial Unicode MS, 10"
set title "Nodo del 3° ordine"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [-3.*pi:3.*pi]; set vrange [0.:2.*pi]
set samples 200, 100; set isosamples 100, 100
set view 60, 30
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
set pm3d depthorder hidden3d
set size square 1., 1.      # parametri b, h (v. p. 2)
#
splot cos(u)*cos(v)+3*(1.5+0.5*sin(u*5/3))*cos(u), \
      sin(u)*cos(v)+3*(1.5+0.5*sin(u*5/3))*sin(u), sin(v)+2*cos(u*5/3) w pm3d
#
unset parametric

```

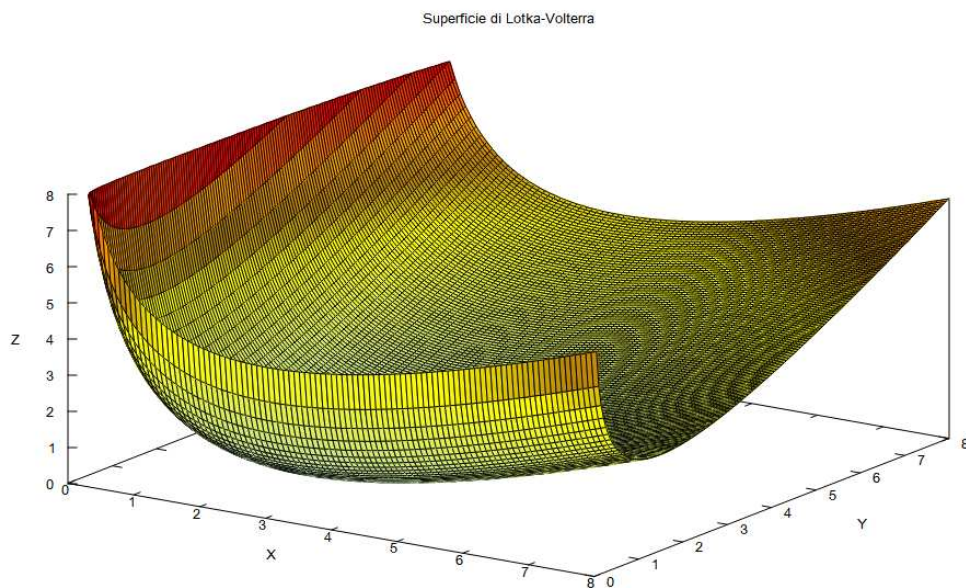


19. Superficie di Lotka-Volterra (†)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie di Lotka-Volterra"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xtics 1.; set ytics 1.; set ztics 1.
set xrange [0.:8.]; set yrange [0.:8.]; set zrange [0.:8.]
set samples 100, 100; set isosamples 150, 150
set pm3d depthorder hidden3d
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
unset colorbox
set view 60, 34, 1, 1
# assegnazione della quaterna parametrica ordinata {a, b, p, q} (utente):
a=1.5
b=1.
p=3.
q=1.
z(x,y)=-p*log(x)+q*x-a*log(y)+b*y
#
splot z(x,y) w pm3d

```



(†) Per un'analisi essenziale della superficie di Lotka-Volterra, si vedano, e.g.,

Pagani, C. D. - Salsa, S., *Analisi Matematica*, vol. 2, P. 199, 289, 291, Zanichelli (-Masson), (1991; RIST. 1998);
 Salsa, S., Squellati, A., *Forme ed Equazioni Differenziali - Appunti ed Esercizi*, P. 252-258, Masson (1990)

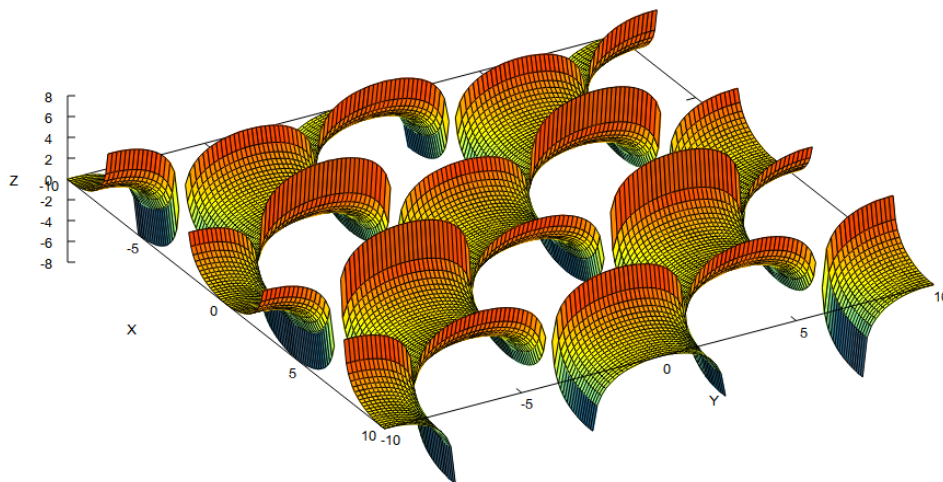
20. Superficie di Scherk (H. F., 1798-1885)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie periodica minima di Scherk"
unset key # nasconde l'espressione matematica del grafico
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
# unset border # nasconde gli assi X, Y, Z e il piano X×Y del dominio (visualizzati di default)
# unset xtics; unset ytics; unset ztics # nasconde le tacche di misura sugli assi X, Y, Z (visualizzate di default)
set samples 150, 150; set isosamples 125, 125
set xrange [-10.:10.]; set yrange [-10.:10.]; set zrange [-8.:8.] # valori-range cartesiani di default
set size square 1., 1. # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set view 30, 60, 1, 1
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
unset colorbox
a=pi/2. # parametro di scala (valore opzionale reale > 0)
#
z(x,y)=a*log(cos(y/a)/cos(x/a))
#
splot z(x,y) w pm3d

```

Superficie periodica minima di Scherk



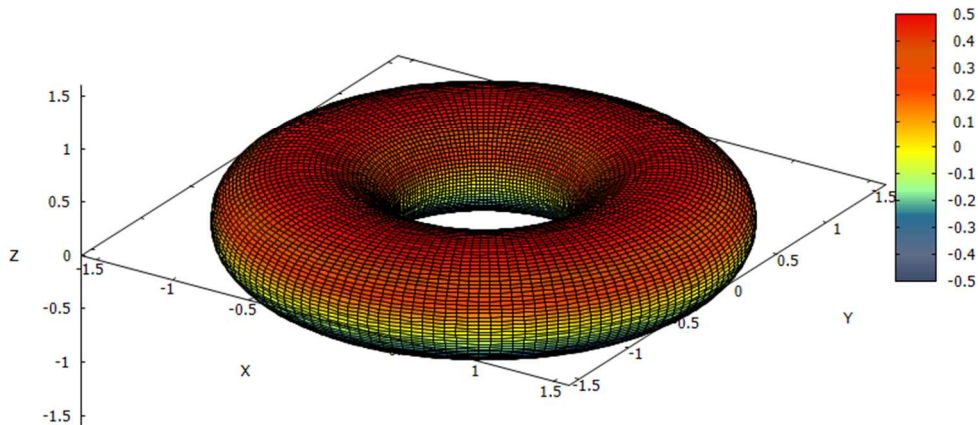
21. Superficie torica oblata

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie torica oblata (rapporto di aspetto r:R=1:2)"
unset key
set xyplane at 0.
set parametric
set urange [0.:2*pi]; set vrange [0.:2*pi]
set samples 150, 150; set isosamples 100, 100
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
set size square 0.7, 1. # parametri b, h (v. p. 2)
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xtics 0.5; set ytics 0.5; set ztics 0.5
set xrange [-1.6:1.6]; set yrange [-1.6:1.6]; set zrange [-1.6:1.6]
set view 55,33,1,1
set pm3d depthorder hidden3d
# unset colorbox
r=0.5 # raggio interno del solido
R=1. # raggio medio del solido dall'asse (esterno) di simmetria passante per il centro del foro
#
splot (R+r*cos(v))*cos(u), (R+r*cos(v))*sin(u), r*sin(v) w pm3d
#
unset parametric

```

Superficie torica oblata (rapporto di aspetto r:R=1:2)



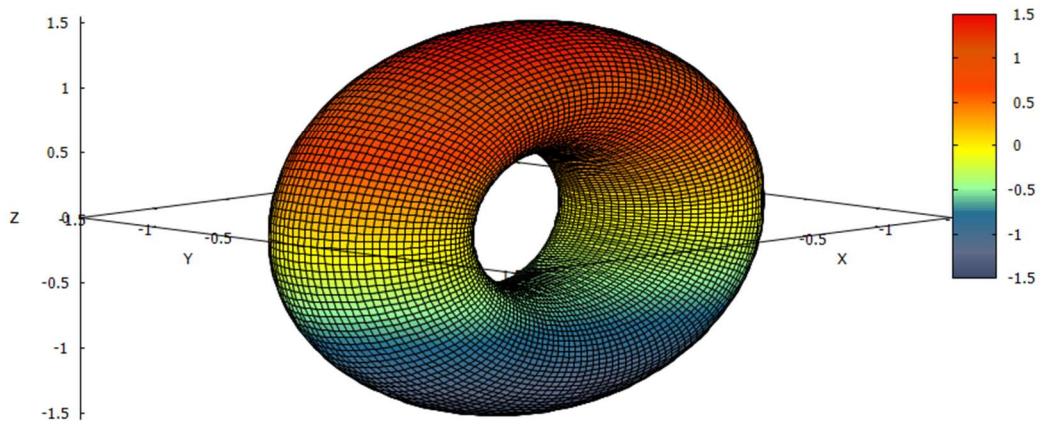
22. Superficie torica prolata

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
unset key
set title "Superficie torica prolata (rapporto di aspetto r:R=1:2)"
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xtics 0.5; set ytics 0.5; set ztics 0.5
set parametric
set urange [0.:2.*pi]; set vrange [0.:2.*pi]
set xrange [-1.5:1.5]; set yrange [-1.5:1.5]; set zrange [-1.55:1.55]
set samples 150, 150; set isosamples 80, 80
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
set size square 0.75, 1. # parametri b, h (v. p. 2)
set view 79,135,1,1
set pm3d depthorder hidden3d
# unset colorbox
r=0.5
R=1.
#
splot (R+r*cos(v))*cos(u), r*sin(v), (R+r*cos(v))*sin(u) w pm3d
#
unset parametric

```

Superficie torica prolata (rapporto di aspetto r:R=1:2)



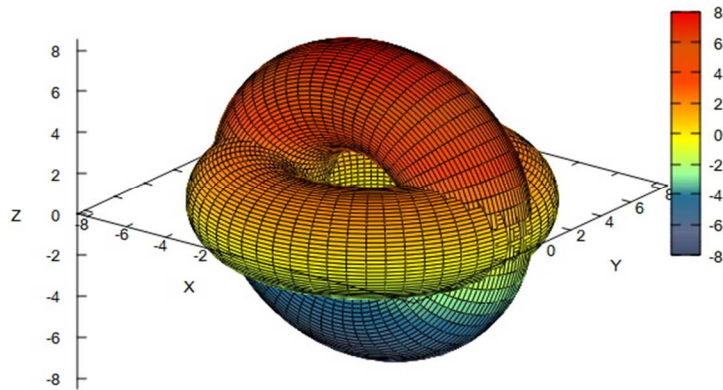
23. Superfici toriche 'siamesi' [1]

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superfici toriche siamesi [1] (rapporto di aspetto r:R=1:2 per entrambe)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set samples 80, 80; set isosamples 60, 60
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
                    10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
set parametric
set urange [0.:2.*pi]; set vrange [0.:2.*pi]
set xrange [-8.5:8.5]; set yrange [-8.5:8.5]; set zrange [-8.5:8.5]
R=5.
r=2.5
set view 67, 37, 1, 1
set pm3d depthorder hidden3d
# unset colorbox
set size square 0.6, 1.      # parametri b, h (v. p. 2)
#
splot (R+r*cos(u))*cos(v), (R+r*cos(u))*sin(v), r*sin(u) w pm3d, \
      (R+r*cos(u))*cos(v), r*sin(u), (R+r*cos(u))*sin(v) w pm3d
#
unset parametric

```

Superfici toriche siamesi [1] (rapporto di aspetto r:R=1:2 per entrambe)



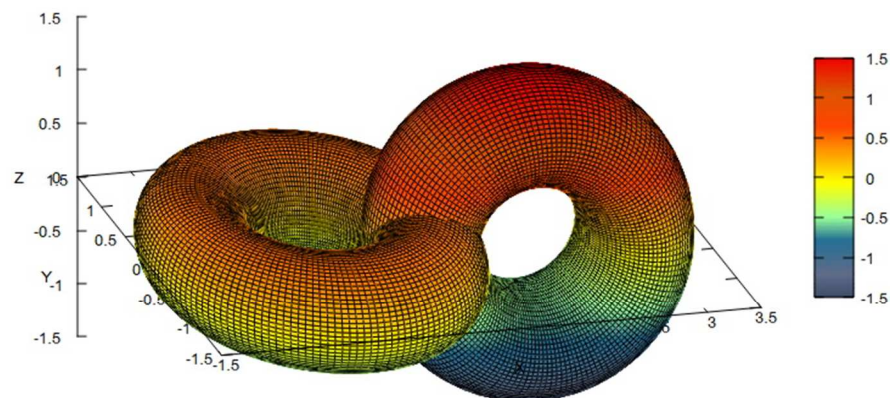
24. Superfici toriche 'siamesi' [2]

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superfici toriche siamesi [2] (rapporto di aspetto r:R=1:2 per entrambe)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
# unset border # nasconde gli assi X, Y, Z e il piano X×Y del dominio (visualizzati di default)
# unset xtics; unset ytics; unset ztics # nasconde le tacche di misura sugli assi X, Y, Z (visualizzate di default)
set samples 150, 150; set isosamples 80, 80
set size square 0.8, 1.0 # parametri b, h (v. p. 2)
set parametric
set urange [0.:2*pi]; set vrange [0.:2*pi]
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
set pm3d depthorder hidden3d
# unset colorbox
set view 60, 345, 1, 1
R=1.
r=0.5
b=2.
#
splot (R+r*cos(v))*cos(u), (R+r*cos(v))*sin(u), r*sin(v) w pm3d, \
b+(R+r*cos(v))*cos(u), r*sin(v), (R+r*cos(v))*sin(u) w pm3d
#
unset parametric

```

Superfici toriche siamesi [2] (rapporto di aspetto r:R=1:2 per entrambe)



25. Superfici toriche concatenate

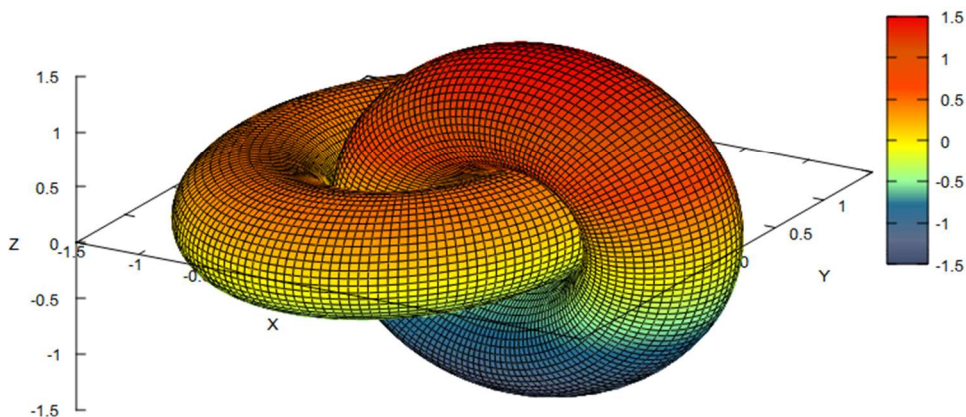
```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superfici toriche concatenate (rapporto di aspetto r:R=1:2 per entrambe)"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set urange [0.:2.*pi]; set vrange [0.:2.*pi]
set samples 150, 150; set isosamples 80, 50
set size square 0.55, 0.9 # parametri b, h (v. p. 2)
set xtics 0.5; set ytics 0.5; set ztics 0.5
# opzione tacche di misura sull'asse X e loro passo

set view 60, 30, 1, 1
set size square 0.8, 1.0 # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
# unset colorbox
#
splot (1.+0.5*cos(v))*cos(u), (1.+0.5*cos(v))*sin(u), 0.5*sin(v) w pm3d, \
1.+(1.+0.5*cos(v))*cos(u), 0.5*sin(v), (1.+0.5*cos(v))*sin(u) w pm3d
#
unset parametric

```

Superfici toriche concatenate (rapporto di aspetto r:R=1:2 per entrambe)



Nota

cf/c grafico a P. 13.

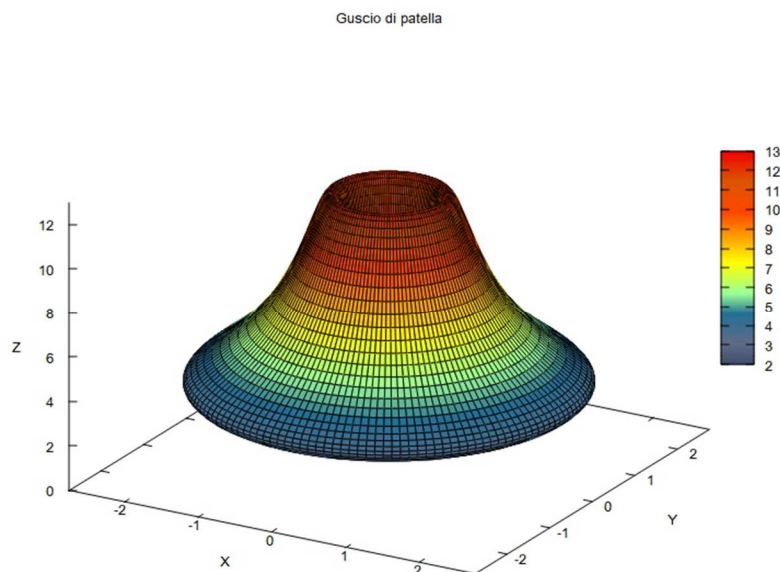
■

26. Guscio di patella

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Guscio di patella"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-2.75:2.75]; set yrange [-2.75:2.75]; set zrange [0.:13.]
set parametric
set urange [-pi:pi]; set vrange [-pi:pi]
set samples 150, 150; set isosamples 100, 100
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
set size square 0.75, 1. # parametri b, h (v. p. 2)
set view 60, 30, 1, 1
set pm3d depthorder hidden3d
# unset colorbox
#
splot cos(u)/(sin(v)+sqrt(2.)), sin(u)/(sin(v)+sqrt(2.)), 5./(cos(v)+sqrt(2.)) w pm3d
#
unset parametric

```

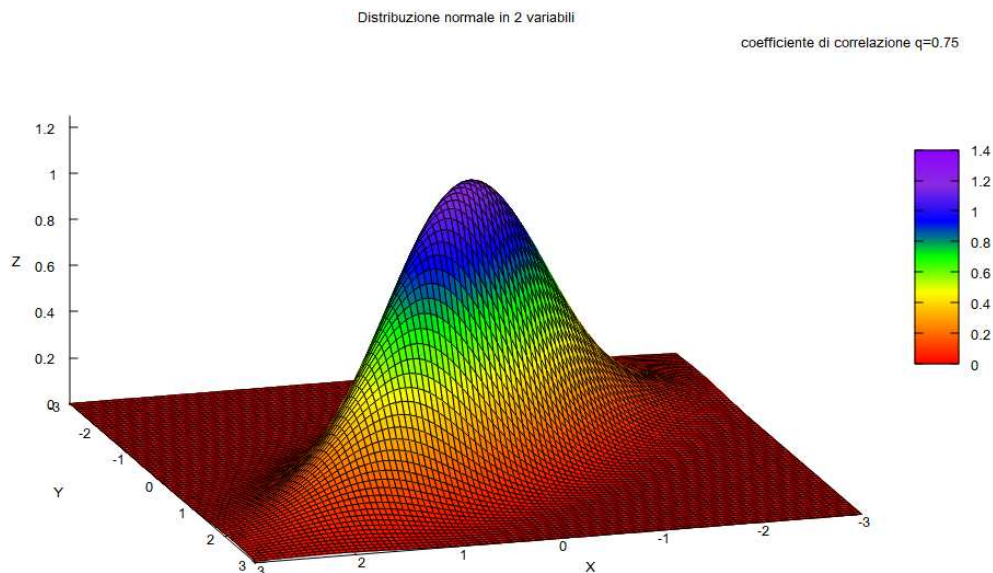


27. Distribuzione normale in 2 variabili

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Distribuzione normale in 2 variabili"
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-3.:3.]; set yrange [-3.:3.]; set zrange [0.:1.25]
set samples 100, 100; set isosamples 80, 80
set pm3d depthorder hidden3d
set size square 1., 1.      # parametri b, h (v. p. 2)
set view 60, 163
set palette defined (0 "#ff0000", 3 "#ff7f00", 6 "#ffff00", 9 "#00ff00", \
                    12 "#0000ff", 15 "#8a2be2", 18 "#8f00ff")
#
q=0.75                      # coefficiente di correlazione; vincolo: 0<q<1
z(x,y)=5*exp(-0.5*(x**2-2*q*x*y+y**2)/(1-q**2))/(2*pi*sqrt(1-q**2))  # amplificazione=5
#
splot z(x,y) w pm3d title "coefficiente di correlazione q=0.75"

```

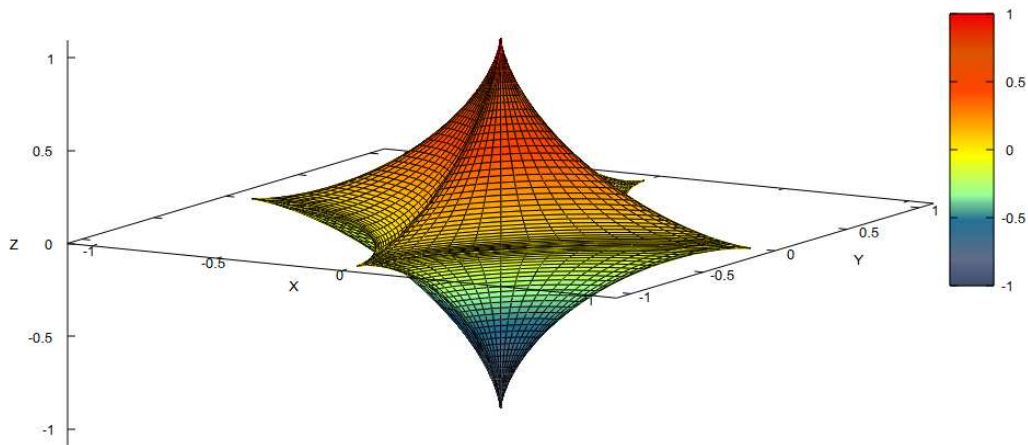


28. Ottaedro iperbolico (coordinate sferiche)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Ottaedro Iperbolico in coordinate sferiche"
unset key # nasconde, da qui in poi, l'espressione matematica di qualsiasi grafico
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set parametric
set size square 1., 1.2
set urange [0.:pi]; set vrange [0.:2*pi]
set xrange [-1.1:1.1]; set yrange [-1.1:1.1]; set zrange [-1.1:1.1]
set samples 100, 100; set isosamples 80, 80
set pm3d depthorder hidden3d
#
a=1.
x(u,v)=(a*sin(u)*cos(v))**3.
y(u,v)=(a*sin(u)*sin(v))**3.
z(u,v)=(a*cos(u))**3.
#
set view 75, 30, 1, 1
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
# unset colorbox
#
splot x(u,v), y(u,v), z(u,v) w pm3d
unset parametric

```

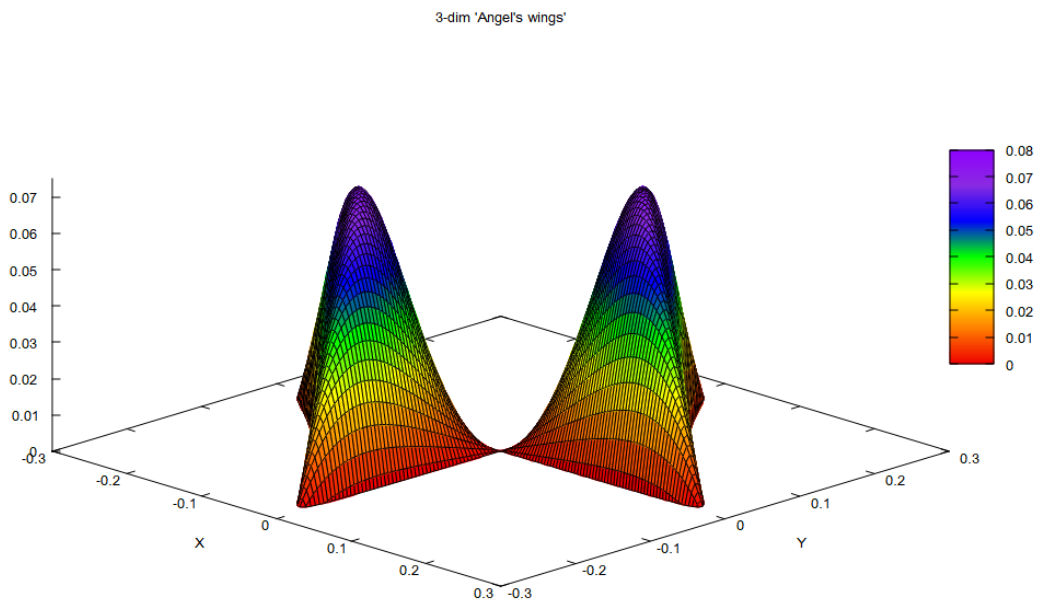


29. 3-dim 'Angel's wings'

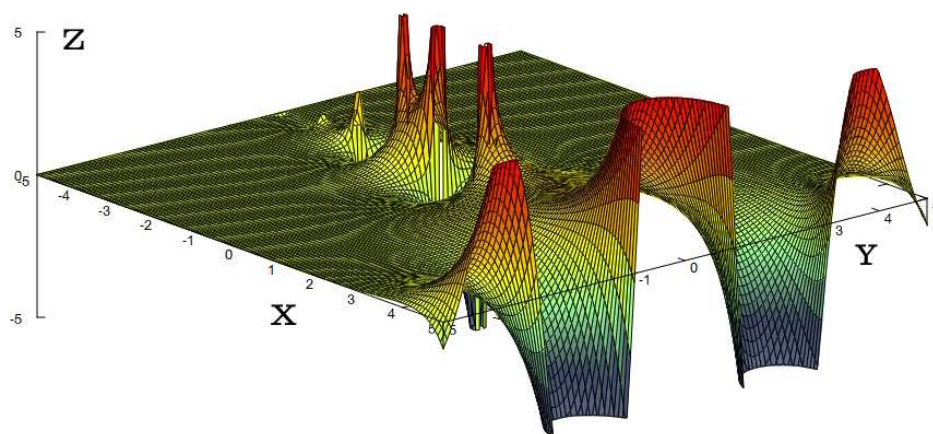
```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "3-dim 'Angel's wings'"
unset key
set xyplane at 0.
set xlabel "X"; set ylabel "Y"; unset zlabel      # asse-Z non marcato
set samples 150, 150; set isosamples 120, 120
set xrange [-0.3: 0.3]; set yrange [-0.3: 0.3]; set zrange [0.:0.075]
set size square 1., 1.      # parametri b, h (v. p. 2)
set pm3d depthorder hidden3d
set view 55, 45, 1, 1
set palette defined (0 "#f00000", 3 "#ff7f00", 6 "#ffff00", 9 "#00ff00", \
12 "#0000ff", 15 "#8a2be2", 18 "#8f00ff")
# unset colorbox
#
a=12.      # parametro di amplificazione (utente)
z(x,y)=a*((x*y-((abs(x))**(1./2)+(abs(y))**(1./2))**12.))
# (cf/c versione 2-dim, script 27, p. 61)
splot z(x,y) w pm3d

```



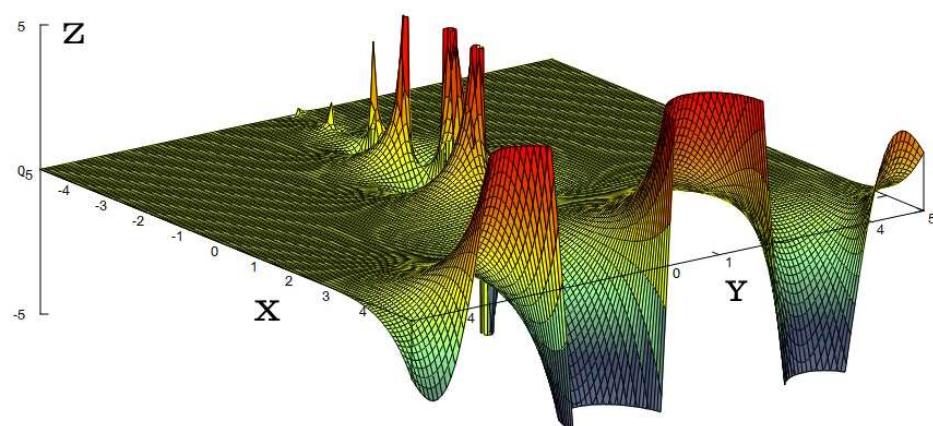
30. Grafico 3-dim della funzione $(x;y) \mapsto \operatorname{Re}(\Gamma(w))$, con $w = x + iy$ (v. PP. 26-27)



$$z = \operatorname{Re}(\Gamma(w))$$

■

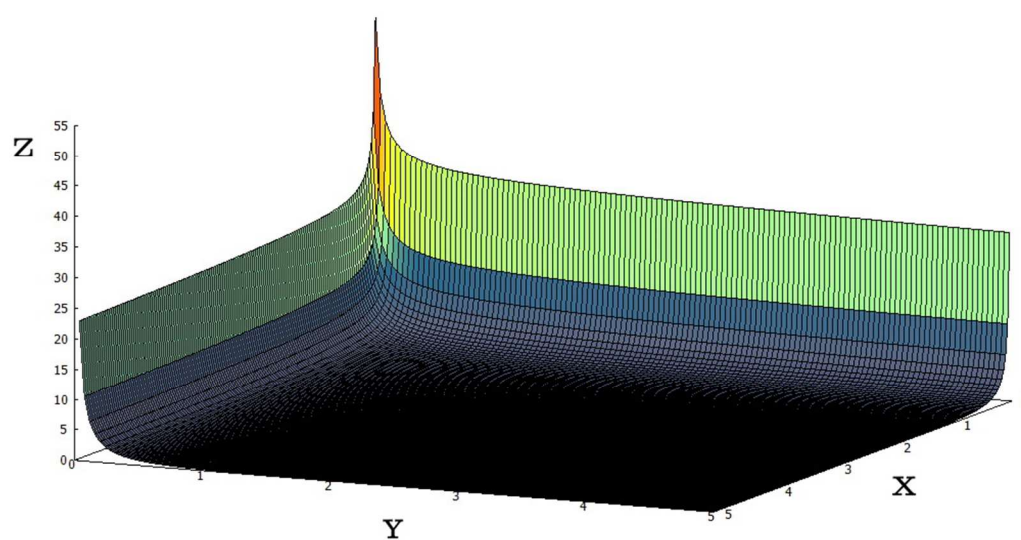
31. Grafico 3-dim della funzione $(x;y) \mapsto \operatorname{Im}(\Gamma(w))$, con $w = x + iy$ (v. P. 26-27)



$$z = \operatorname{Im}(\Gamma(w))$$

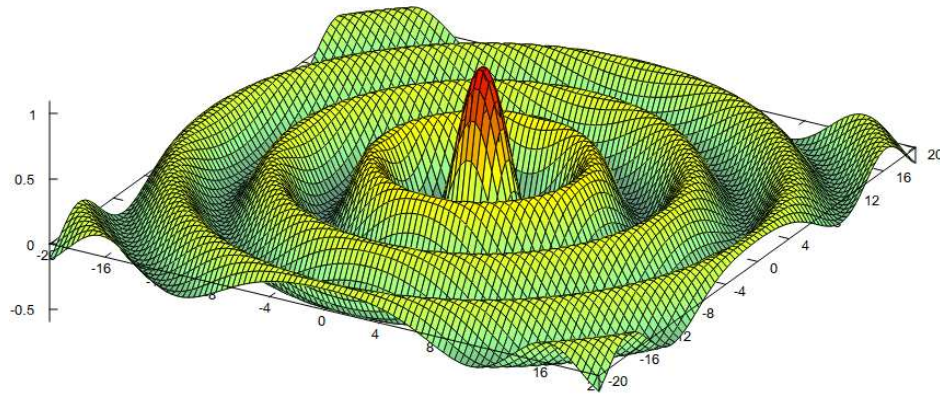
■

32. La Funzione Beta di Euler, $(x,y) \mapsto z := B(x,y)$, con $x, y > 0$ (v. P. 27)



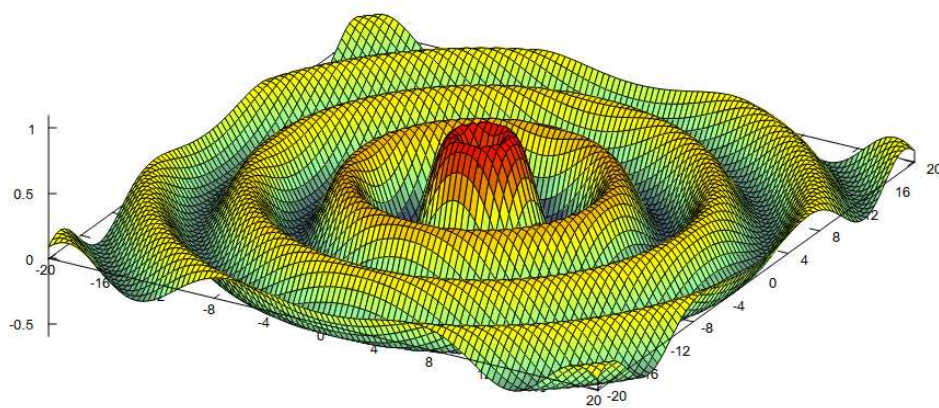
■

33. La Funzione 3-dim di Bessel $(x;y) \mapsto J_0(\rho)$ (v. P. 28)



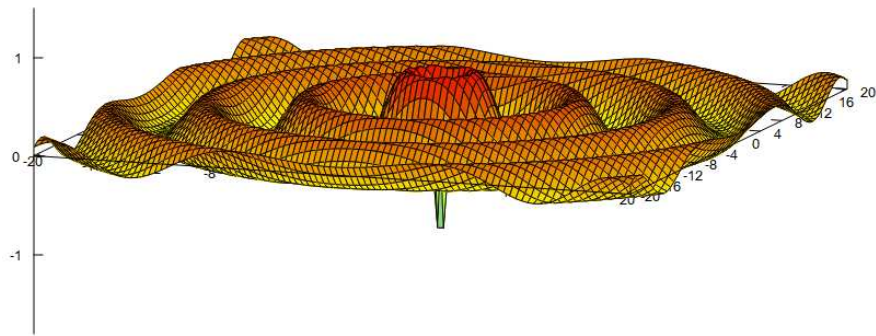
■

34. La Funzione 3-dim di Bessel $(x;y) \mapsto J_1(\rho)$ (v. P. 28)



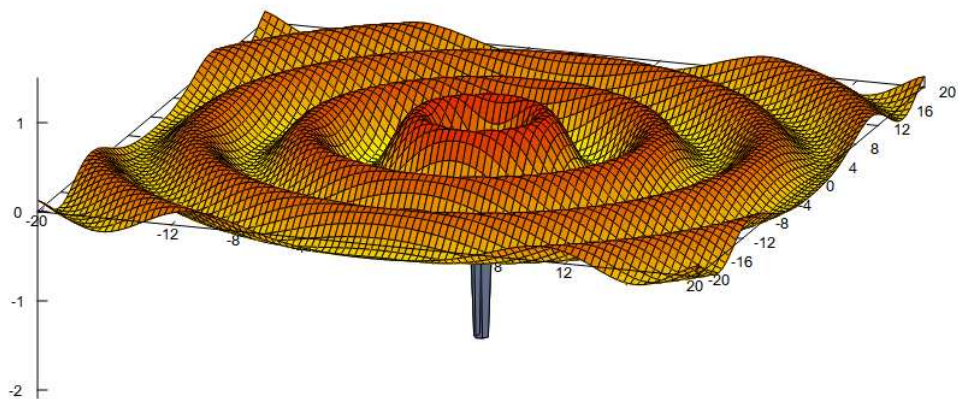
■

35. La Funzione 3-dim di Bessel $(x;y) \mapsto N_0(\rho)$ (v. P. 29)



■

36. La Funzione 3-dim di Bessel $(x;y) \mapsto N_1(\rho)$ (v. P. 29)



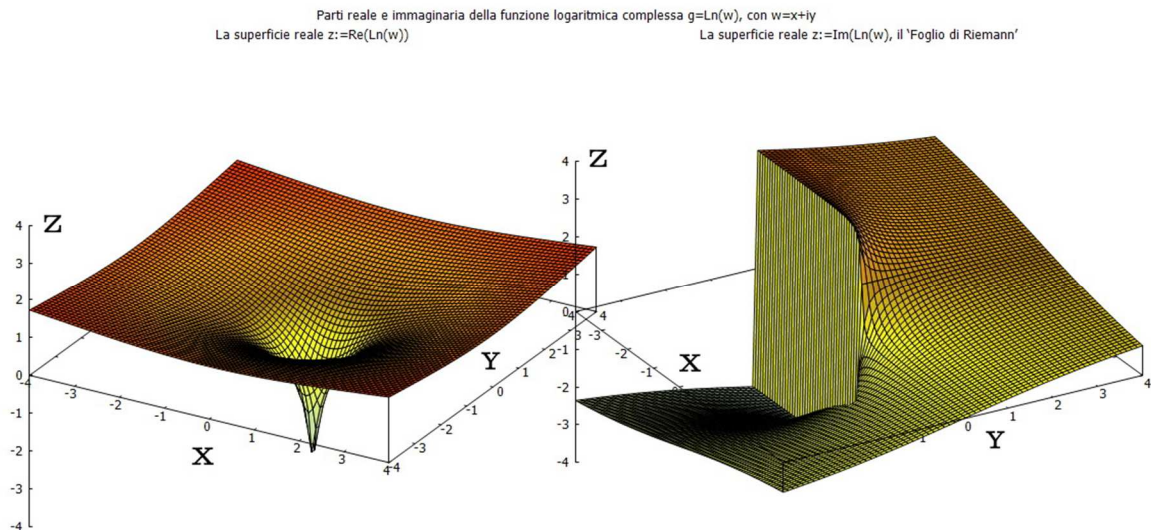
■

37. La funzione logaritmica complessa: $g = \text{Ln}(w)$, con $w = x + iy$

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set multiplot
set multiplot title "Parti reale e immaginaria della funzione logaritmica complessa  $g = \text{Ln}(w)$ , con  $w = x + iy$ "
#
set title "La superficie reale  $z := \text{Re}(\text{Ln}(w))$ "
unset key
set xyplane at 0.; #; set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-4.:4.]; set yrange [-4.:4.]; set zrange [-4.:4.]
set samples 85, 85; set isosamples 70, 70
set pm3d depthorder hidden3d
set size square 0.6, 1. # parametri b, h (v. p. 2)
set palette defined (0 "#add8e6", 12 "#ffff00", 15 "#b8860b", 18 "#ffa500", 22 "#ff0000")
unset colorbox
set view 60, 30, 1., 1.
#
Re(x,y)=(1./2)*log(x**2.+y**2.)
splot Re(x,y) w pm3d
#
set title "La superficie reale  $z := \text{Im}(\text{Ln}(w))$ , il 'Foglio di Riemann'"
set origin 0.43, 0.
# Si noti che i comandi dichiarativi per il primo grafico, assenti per il secondo, \
# si intendono confermati anche per questo; altrimenti, vanno inseriti esplicitamente.
set view 60, 60, 1., 1.
#
k=0. # qualsiasi valore di k (in formato reale), uguale o diverso da 0, va specificato dall'utente
Im(x,y)=atan2(y,x)+2.*pi*k
splot Im(x,y) w pm3d
#
unset multiplot

```



Va sottolineato l'uso della funzione di libreria `atan2`, appropriata, $\forall \{x, y\} \neq \{0, 0\} \in \mathbb{R}^2$ (e.g., in $\Re z + i\Im z \in \mathbb{C}$), nella rappresentazione funzione/argomenti `atan2(y,x)` del rapporto y/x (cf/c.: https://en.wikipedia.org/wiki/Complex_logarithm).

Invece, la funzione `atan` restituisce il valore (*principale*) corretto dell'arco-tangente del rapporto tra due variabili, $\tan^{-1}(y/x)$, solo se $x > 0$ (denominatore *solo* positivo) (v. Help >> Index >> Indice >> expressions functions).

I raccordi in corrispondenza del *taglio di diramazione* del grafico (v.) risultano non tanto antiestetici quanto *analiticamente estranei* al grafico di ciò che dovrebbe apparire come una *pellicola sottile*, più che un corpo solido. Un 'bug' grafico nell'architettura di GNUPLOT 6.0.1.0 ?

■

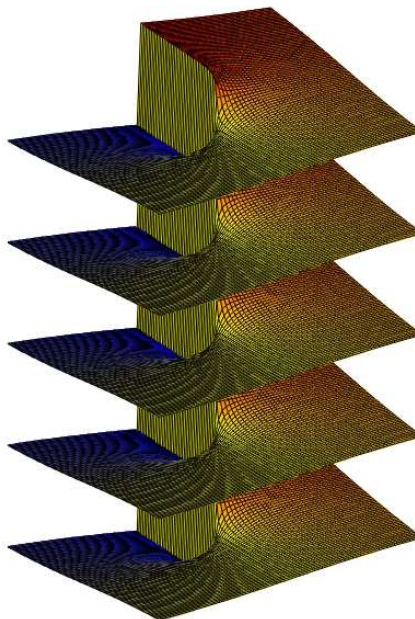
38. 'Fogli di Riemann' consecutivi, con $w = x + iy$ (cf/c script 37)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set multiplot
set multiplot title "Pila dei 'fogli di Riemann' di ordini k = -2, -1, 0, 1, 2"
unset key
set xyplane at 0.; set origin 0.23, -0.46; # set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-20.:20.]; set yrange [-20.:20.]; set zrange [-20.:20.]
# set xtics 1.; set ytics 1.; set ztics 2.
unset tics; unset border
set samples 85, 85; set isosamples 70, 70
set pm3d depthorder hidden3d
set palette defined (0 "#0000ff", 1 "#ffff00", 2 "#ff0000")
unset colorbox
set size square 0.5, 1.8          # parametri b, h (v. p. 2)
set view 80, 60, 1, 1
#
s1(x,y)=atan2(y,x)-4.*pi          # 'foglio di Riemann' di ordine k = -2
splot s1(x,y) w pm3d
#
s2(x,y)=atan2(y,x)-2.*pi          # 'foglio di Riemann' di ordine k = -1
splot s2(x,y) w pm3d
#
s3(x,y)=atan2(y,x)                # 'foglio di Riemann' di ordine k = 0 (foglio 'principale')
splot s3(x,y) w pm3d
#
s4(x,y)=atan2(y,x)+2.*pi          # 'foglio di Riemann' di ordine k = 1
splot s4(x,y) w pm3d
#
s5(x,y)=atan2(y,x)+4.*pi          # 'foglio di Riemann' di ordine k = 2
splot s5(x,y) w pm3d
#
unset multiplot

```

Pila dei 'fogli di Riemann' di ordini k = -2, -1, 0, 1, 2



Come per lo script 37, il 'rendering' grafico in corrispondenza dei tagli di diramazione (raccordi) sembra permanere come un'imperfezione nella gestione delle discontinuità essenziali da parte di GNUPLOT 6.0.1.0. Infatti, la visualizzazione del grafico non dovrebbe apparire quella di una 'gradinata' solida 3-dim ma quella di una pellicola sottile ad avvitamento infinito, di passo 2π (cf/c 'Elicoide lineare', p. 12).

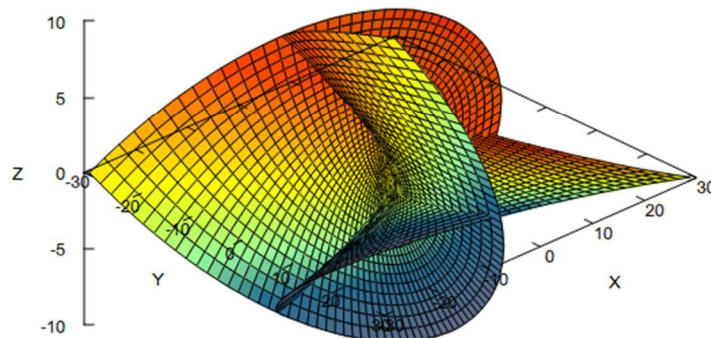
39. Superficie di A. Enneper (1855)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Superficie di Enneper"
unset key # nasconde l'espressione matematica del grafico
set xyplane at 0.; set origin 0.2, 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set xrange [-30.:30.]; set yrange [-30.:30.]; set zrange [-10.:10.]
set samples 80, 80; set isosamples 60, 60
set size square 0.57, 0.9 # parametri b, h (v. p. 2)
set parametric
set pm3d depthorder hidden3d
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", 10 "#ffff00", \
15 "#ff4500", 18 "#dd5500", 21 "#f00000") # colorbox personalizzata
unset colorbox # opzionale
set view 123, 44, 1, 1.
x(u,v)=u*(1.-(u**2.)/3.+v**2.)/3. # x-componente di r(u,v)
y(u,v)=-v*(1.-(v**2.)/3.+u**2.)/3. # y-componente di r(u,v)
z(u,v)=(u**2.-v**2.)/3. # z-componente di r(u,v)
splot x(u,v), y(u,v), z(u,v) w pm3d
#
unset parametric

```

Superficie di Enneper



Nota

La *Superficie di Enneper*, a curvatura totale negativa $K_t = -4\pi$, soddisfa l'equazione definita implicitamente dal polinomio cartesiano di 9° grado in 3 variabili – alquanto ... pesante da gestire (cf/c § 7, p. 8) –

$$\begin{aligned}
 &64z^9 - 144(x^2 - y^2)z^6 \\
 &- 128z^7 - 432(x^2 + y^2)z^5 - 27(5(x^4 + y^4) + 26x^2y^2)z^3 \\
 &- 27(x^6 - y^6) - 240(x^2 - y^2)z^4 - 162(x^4 - y^4)z^2 + 81(x^2 - y^2)x^2y^2 \\
 &+ 64z^5 + 48(x^2 + y^2)z^3 + 9((x^2 - y^2)^2)z = 0.
 \end{aligned}$$

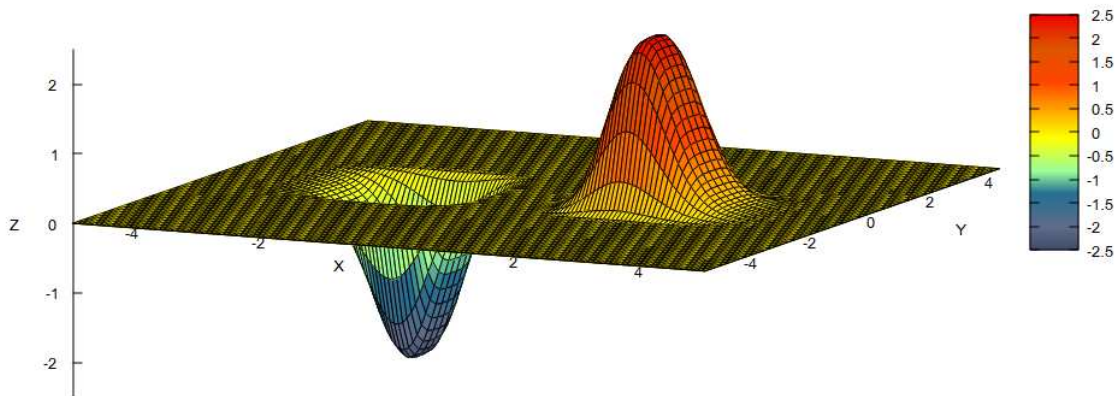
40. Impulso ondulatorio solitonico

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Impulso ondulatorio solitonico"
unset key
set xyplane at 0
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
# unset border
# nasconde gli assi X, Y, Z e il piano X×Y del dominio (visualizzati di default)
# unset xtics; unset ytics; unset ztics
# nasconde le tacche di misura sugli assi X, Y, Z (visualizzate di default)
set samples 150, 150
set isosamples 60, 50
set xrange [-5.:5.]; set yrange [-5.:5.]; set zrange [-2.5:2.5]
set size square 1.0,1.0
set pm3d depthorder hidden3d
set view 72, 25, 1, 1
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
# unset colorbox
#
z(x,y)=x**7*exp(-(x**2.+y**4.))
splot z(x,y) w pm3d

```

Impulso ondulatorio solitonico

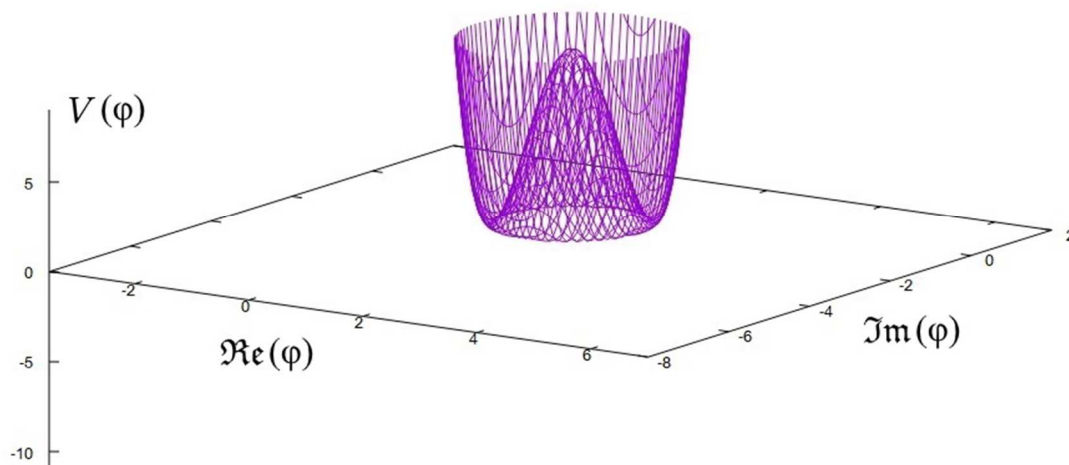


41. Potenziale di Higgs [1] (il 'cappello messicano')

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
# set title "Simulazione qualitativa del Potenziale di Higgs (il 'cappello messicano')"
unset key
set xyplane at 0.
unset xlabel; unset ylabel; unset zlabel
set xrange [-3.5:7.]; set yrange [-8.:2.]; set zrange [-11.:9.]
set samples 110, 110; set isosamples 90, 90
set view 67, 34, 1, 1
set size square 1., 1.
set pm3d depthorder hidden3d
#
a=5.; b=14.; c=8.4      # parametri di forma del potenziale (utente)
z(x,y)=a*(x**2+y**2)**2.-b*(x**2+y**2)+c
#
splot z(x,y)           # si noti l'assenza del comando 'w pm3d' \
                        e di qualsiasi altra assegnazione cromatica

```



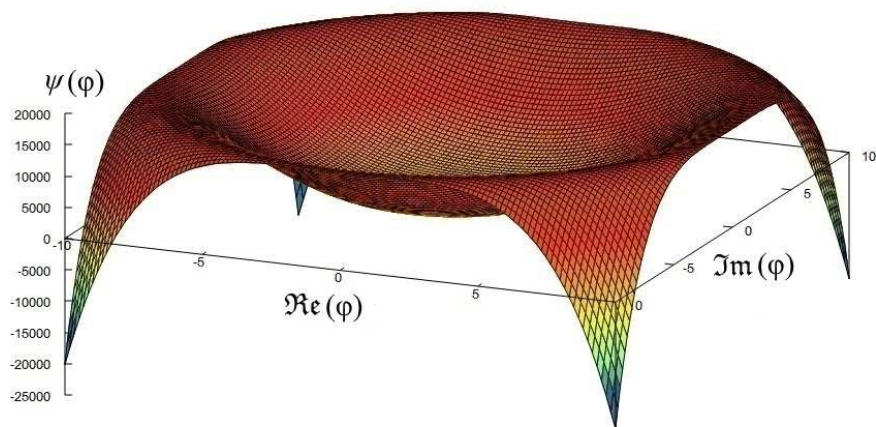
■

42. Potenziale di Higgs [2] (c\correzione di 'tunneling' quantistico)

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Correzione quantistica di 'tunneling' del Potenziale di Higgs"
unset key
set xyplane at 0.
unset xlabel; unset ylabel; unset zlabel
set samples 120, 120; set isosamples 100, 100
set pm3d depthorder hidden3d
set xrange [-10.:10.]; set yrange [-10.:10.]
set view 60, 23, 1, 1
set size square 1., 1.
set palette defined (0 "#3f4c6b", 2 "#606c88", 5 "#256f95", 7 "#99ff99", \
10 "#ffff00", 15 "#ff4500", 18 "#dd5500", 21 "#f00000")
unset colorbox
#
z(x,y)=-6.*(x**2+y**2)**2.*log(sqrt(x**2+y**2)/13)
#
splot z(x,y) w pm3d

```



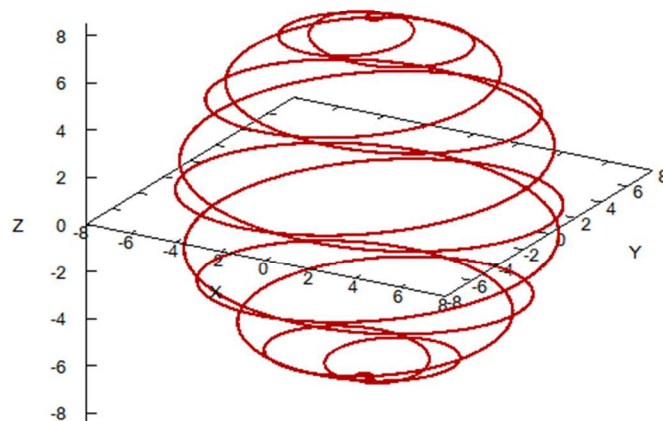
43. Elica sferica (parametrizzazione [1])

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Elica sferica 1"
unset key
set xyplane at 0.; set origin 0.23, 0.
set samples 5000
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set size square 0.53, 1.0      # parametri b, h (v. p. 2)
set view 70, 30, 1, 1
set parametric
#
# Costanti definite dall'utente:
r=7.5      # raggio del supporto sferico
a=1.       # componente polare (theta) della velocità angolare del punto mobile sulla linea
b=12.*a    # componente azimutale (phi) della velocità angolare del punto mobile (multipla di 'a')
set vrange [0.:20.*pi]      # 10 = numero di cicli della coordinata temporale 'v'
set xrange [-8.:8.]; set yrange [-8.:8.]; set zrange [-8.5:8.5]
x(v)=r*sin(a*v)*cos(b*v)
y(v)=r*sin(a*v)*sin(b*v)
z(v)=r*cos(a*v)      # passo dell'elica
#
splot x(v), y(v), z(v) lc rgb "#b20000" lw 2      # lw (=linewidth) 2: valore 2 dello spessore della linea (utente); \
                                                    lc (=linecolor): colore della linea (utente)
unset parametric

```

Elica sferica 1



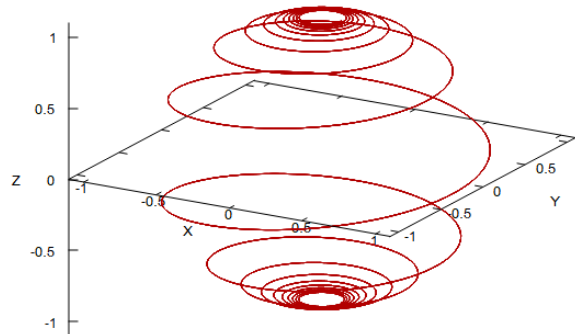
44. Elica sferica (parametrizzazione [2])

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Elica sferica 2"
unset key
set xyplane at 0.; set origin 0.23, 0.
set samples 5000
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set size square 0.6, 1. # parametri b, h (v. p. 2)
set view 70, 30, 1, 1
set parametric
set urange [-20.*pi:20.*pi] # coordinata temporale ciclica
set xrange [-1.1:1.1]; set yrange [-1.1:1.1]; set zrange [-1.1:1.1]
#
a=0.12 # parametro di scala (utente)
x(u)=cos(u)/(1+(a*u)**2)**(1./2)
y(u)=sin(u)/(1+(a*u)**2)**(1./2)
z(u)=a*u/(1+(a*u)**2)**(1./2) # passo dell'elica
#
splot x(u), y(u), z(u) lc rgb "#b20000" # qui, ovviamente, è lw=1 (default)
unset parametric

```

Elica sferica 2

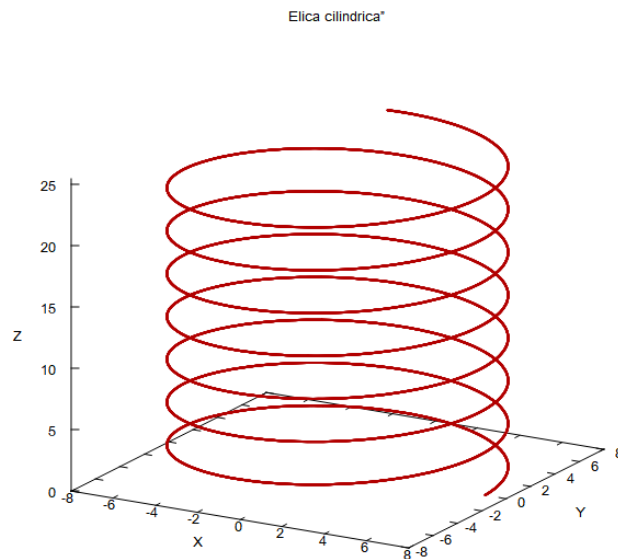


45. Elica cilindrica

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Elica cilindrica"
unset key
set samples 5000
set xyplane at 0.; set origin 0.23, 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set size square 0.63, 1. # parametri b, h (v. p. 2)
set view 70, 30, 1, 1
set parametric
    # costanti definite dall'utente:
h=3.5 # passo dell'elica
set urange [0.:8.*pi] # numero di cicli, 4, della coordinata temporale (qui, è 'u')
set xrange [-8.:8.]; set yrange [-8.:8.]; set zrange [0.:25.5]
r=7.
x(u)=r*cos(2*pi*u)
y(u)=r*sin(2*pi*u)
z(u)=h*u
#
splot x(u), y(u), z(u) lw 2 lc rgb "#b20000"
unset parametric

```



46. Elica conica

```

set term qt 0 size 1056, 624 font "Arial Unicode MS, 10" enhanced
set title "Elica conica"
unset key
set samples 5000
set xyplane at 0.; set origin 0.23, 0.
set xlabel "X"; set ylabel "Y"; set zlabel "Z"
set size square 0.63, 1. # parametri b, h (v. p. 2)
set view 70, 30, 1, 1
set parametric
    # le 3 costanti seguenti sono definite dall'utente:
a=1. # componente tangenziale all'elica della velocità del punto mobile
b=6.*a # componente azimutale (phi) della velocità angolare (proporzionale ad 'a')
h=0.3*a # componente z-assiale della velocità del punto mobile
set urange [0.:4*pi] # numero di cicli, 2, della variabile-tempo t (qui, è t = 'u')
set xrange [-13.:13.]; set yrange [-13.:13.]; set zrange [0.:4.]
x(u)=a*u*cos(b*u)
y(u)=a*u*sin(b*u)
z(u)=h*u # passo dell'elica
#
splot x(u), y(u), z(u) lc rgb "#b20000" lw 2
#
unset parametric

```

Elica conica

