

UTKAST TILL

INTRODUKTION TILL GNUPLOT

JOHAN WILD

©Johan Wild 2009

johan.wild@europaskolan.se

Får gärna användas i undervisning, kontakta i så fall författaren.

5 november 2009

Innehåll

1	Inledning	4
2	Att använda Gnuplot	4
3	Enkla grafer och användbara tips	4
4	Inställningar	5
4.1	Ändra intervall	5
4.2	Ändra kurvans egenskaper	5
4.3	Rubriker och axlar	6
4.4	Samla allt i script	7
5	Funktioner	8
6	Inläsning av data från fil	8
7	Flera grafer i en bild	9
8	Logaritmiska skalor	11
9	Att spara och skriva ut en graf	12
10	Kurvor på parameterform	14
11	Nivåytor	16
12	Parametriserade ytor	17

1 Inledning

Gnuplot är ett program som kan rita grafer till funktioner definierade antingen genom ett uttryck eller genom en datamängd som finns på en fil. Gnuplot har en lång historia som tar sin början på den tiden en datorskärm endast kunde visa text (eller ännu tidigare, jag är inte säker).

På den tiden kunde man dock ge skrivare instruktioner om att rita grafik. Därför utvecklades Gnuplot, som omvandlar en rad instruktioner (som alltså skrivs med vanlig text) till något som en skrivare förstår. Nuförtiden kan naturligtvis resultatet även visas på en datorskärm, där användaren även kan vrida och vända på grafer, zooma i dem och mycket mer.

Om man är van vid att använda moderna operativsystem upplevs kanske detta sätt att använda Gnuplot på som en aning omodernt. Må så vara, men begrundat att Gnuplot (och andra program av denna typ, till exempel Maxima) har utvecklats och förfinats under flera decennier, mycket längre än Windows har funnits. Gnuplot är även en de facto standard för att rita grafer i (natur-)vetenskapliga sammanhang, och många läromedel på universitetsnivå är fyllda med Gnuplot-grafer.

2 Att använda Gnuplot

När Gnuplot startas visas en text om copy-right, vilken version av Gnuplot man använder och var man skall vända sig om man upptäcker fel i Gnuplot. Det sista som står är:

```
gnuplot>
```

Här är det meningen att du skall skriva en instruktion till Gnuplot. Till exempel kan du skriva `plot x`. Då visas grafen till funktionen $y(x) = x$. Du kan även ge instruktioner som förändrar grafens utseende.

Det vanliga är att man samlar en rad instruktioner i en fil, som man sedan laddar in i Gnuplot. En sådan fil kallas för ett script. Script för Gnuplot skall ha filnamnstillägget `plt`.

Du kan använda vilken text-editor som helst för att skriva dina script.

3 Enkla grafer och användbara tips

Det kommando som säger åt Gnuplot att rita en graf heter `plot`. Efter kommandot `plot` skall man skriva den funktion man vill rita grafen till, uttryckt som funktion av x . Efter detta uttryck kan man fylla på med information om vilken titel funktionen skall ha, hur tjock kurvan skall vara, vilken färg den skall ha och mycket mer. Man kan utelämna detta

om man vill (vilket görs i detta avsnitt). Man kan ha flera uttryck efter varandra om man vill rita flera kurvor i samma graf.

Övning 3.1. Ge Gnuplot följande instruktioner.

```
plot x+3
plot 0,x**2
plot 0,x**3/100, 2*x+4, sqrt(x),pi
```

Potenser skrivs med dubbla asterisker, x^2 skrivs alltså `x**2`. Roten ur x skrivs `sqrt(x)`. Konstanten π heter `pi`. Normalt ritas inte en x -axel ut, men ett enkelt sätt att få en är att rita funktionen $y = 0$. Det görs genom att helt enkelt lägga till funktionen `0` som i de två sista exemplen ovan.

Du kan bläddra mellan senast inslagna funktioner med upp- och nedpil-tangenterna. Om du skriver `replot` ritas den senaste grafen.

4 Inställningar

4.1 Ändra intervall

Innan man ger Gnuplot instruktionen `plot` kan man ställa in allt möjligt för att få grafen som man vill ha den. Till exempel kan man bestämma vilka intervall på axlarna som skall ritas.

Övning 4.1. Skriv in följande kommandon.

```
reset
set xrange[-5:5]
set yrange[-3:5]
plot 0, x**3, 2*x+1
set xrange[-2:2]
replot
```

Det första kommandot, `reset`, återställer Gnuplot till utgångsläget. Sedan ställs intervallen för de båda axlarna in, varefter en graf ritas med `plot`. Vill man ändra på någon inställning går det bra. Kommandot `replot` ritas som sagt den senaste grafen, men den kommer nu att visas med den nya inställningen.

4.2 Ändra kurvans egenskaper

En enskild kurvas egenskaper kan ändras genom att du anger linjetyp och tjocklek för kurvan. Detta görs genom att lägga till orden `linetype` följt av ett tal som anger linjetypen, och ordet `linewidth` följt av ett tal som anger detta. Om du vill kan du använda förkortningarna `lt` och `lw` istället.

Du kan även ge en kurva en titel genom att lägga till ordet `title` och din titel inom citationstecken eller apostrofer. Vill du inte ha någon titel kan du ange `notitle`.

Övning 4.2. Skriv in följande kommando. Ändra sedan på linjetyper och linjebredder för att få en känsla för hur dessa skiljer sig åt.

```
plot x linetype 2 linewidth 4 title 'hej' ,2*x+3 lt 3 lw 1 notitle
```

4.3 Rubriker och axlar

Du kan ange rubrik för hela grafen och ge axlarna en förklaring med följande konstruktion.

```
set title 'Rubrik'
set xlabel 'X-axel'
set ylabel 'Y-axel'
```

Ofta blir en graf tydligare om det finns något slags ruttmönster som gör avläsning i grafen enklare. Man kan också speciellt markera koordinataxlarna. Kommandona för detta är `set grid`, `set xzeroaxis` och `set yzeroaxis`. Vill man ändra på hur tätt ruttmönstret blir skriver man `set xtics 0.25`. Talet anger hur långt det skall vara mellan två hjälplinjer. Man kan också sätta dit markeringar på axlarna mellan hjälplinjerna. Kommandot för det är `set mxtics 4`. Här anger talet hur många intervall som skall markeras mellan två hjälplinjer.

För både axlar och hjälplinjerna gäller att de kan få olika utseenden med `linetype` och `linewidth`

Övning 4.3. Skriv in följande kommandon. Experimentera sedan med de olika inställningarna så att du lär dig vad som är rimligt.

```
reset
set xrange[-2:3]
set yrange[-1:5]

set grid linewidth 1 linetype 3
set xtics 1
set ytics 2
set mxtics 2
set mytics 4

set xzeroaxis lw 2 lt 4
set yzeroaxis lw 2 lt 5

plot x**2 lw 3 notitle
```

För att spara arbete ändrar du på en sak i taget, till exempel ställer in ett nytt värde på `xtics` med `set xtics 0.5` och skriver `replot`.

Vill man inte längre visa hjälplinjerna kan man skriva `unset grid`. Generellt gäller att allt som kan sättas dit med `set` kan tas bort med `unset`.

4.4 Samla allt i script

Som du förstår blir det en hel del instruktioner. Det är alltså bra att samla dessa i ett script.

Övning 4.4. Skriv följande script och kör det i Gnuplot.

```
# Exemplifierar några inställningar.
# Johan Wild 2008-09-09

reset

set title 'Rubrik'
set xlabel 'X-axel'
set ylabel 'Y-axel'

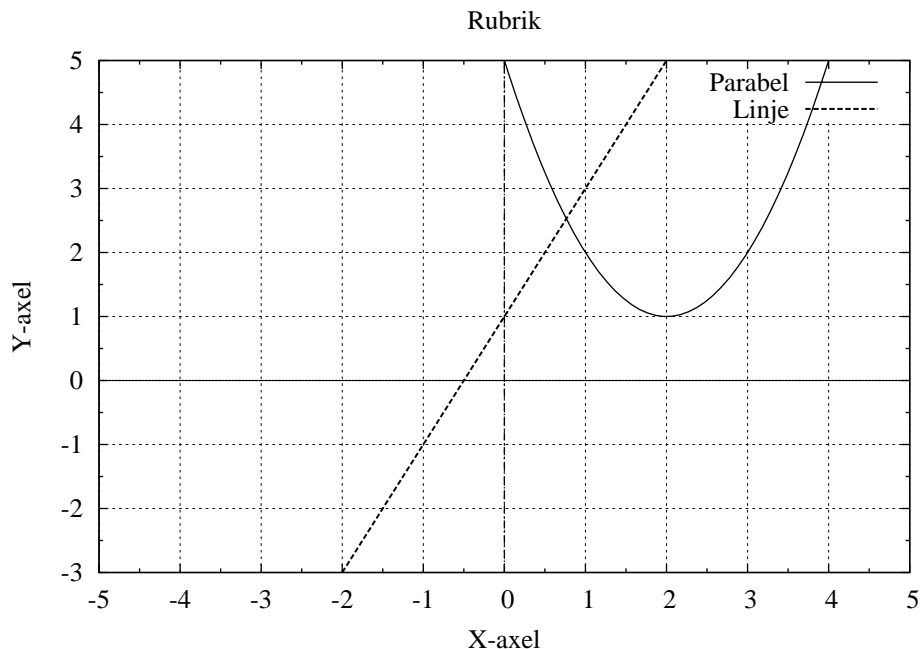
set xrange[-5:5] # Kommentar på en rad
set yrange[-3:5]

set grid linewidth 1 linetype 3

set xtics 1
set ytics 1
set mxtics 2
set mytics 2

set xzeroaxis lw 2 lt 4
set yzeroaxis lw 2 lt 5

plot 0 notitle,\
      (x-2)**2+1 lw 2 lt 1 title 'Parabel',\
      2*x+1 lw 3 lt 2 title 'Linje'
```



Med tecknet # anger man att resten av den raden inte skall läsas av Gnuplot. Det är brukligt att inleda script med en kommentar om vad syftet med scriptet är, vem som gjort det och när.

Tycker man att en rad blir för lång kan man ange tecknet \ som betyder att Gnuplot skall tolka det på raden nedanför som en förlängning på den

innevarande. Ofta används detta så att en kurva i grafen får en egen rad i scriptet.

5 Funktioner

Om man vill återanvända en funktion flera gånger, eller bara få ett mer lättläst script, kan det vara bra att definiera en funktion. När man definierar funktionen måste inte den oberoende variabeln heta x . Däremot använder plot sig av x vid anrop till funktioner från `plot`.

Man kan även definiera funktioner av flera variabler.

Övning 5.1. Gör ett script med följande innehåll eller skriv in instruktionerna direkt i Gnuplot.

```
f(q)=q+1
g(t)=t**2-3
plot f(x),f(x-3),g(x)
```

```
h(x,k)=k*x+3
plot h(x,2), h(x,3), h(x,-1)
```

6 Inläsning av data från fil

Antag att du har en fil med namnet `data.dat` som har ett innehåll av följande typ.

```
# Kommentar om vad data avser
# X      Y1      Y2
1.0      10.1    8.0
2.0      12.2    9.0
3.0      14.3    10.0
4.0      16.4    11.0
5.0      18.5    12.0
6.0      20.6    13.0
7.0      22.7    14.0
8.0      24.8    15.0
```

Dessa data kan nu visualiseras med följande konstruktion.

```
plot 'data.dat' using 1:2 with lines, \
     'data.dat' using 1:3 with points
```

I detta exempel används första kolumnen i `data.dat` som x -värden till två kurvor som tar sina y -värden från kolumn två (`using 1:2`) respektive tre (`using 1:3`). Det är också exemplifierat hur punkterna är återgivna med

linjer mellan punkterna (`with lines`) respektive med bara de enskilda punkterna (`with points`).

Har man ganska lite data kan man inkludera det i sitt script. Istället för filnamn skriver man `'-'`. Gnuplot läser då data från raderna under raden med `plot` till dess den stöter på ordet `end`.

```
set xrange[0:4]
set yrange[0:6]
plot '-' using 1:2 with points pointtype 4 pointsize 2 notitle
1 1
2 3
3 5
end
```

7 Flera grafer i en bild

Ibland vill man visa flera grafer i samma bild, till exempel kanske man vill förstora upp ett delområde i en graf, eller rita en hastighet-tid-graf under motsvarande sträcka-tid-graf för en kropp.

En graf i Gnuplot ritas normalt i ett område som upptas av en kvadrat med sidan 1. Nedre vänstra hörnet av grafen har koordinat $(0, 0)$ och övre högra hörnet har koordinat $(1, 1)$. Beroende på terminal så dras förvisso detta område ut så att bilden inte ser ut som en kvadrat, men hörnen har i alla fall dessa koordinater.

Genom att ändra på storlek och position för grafen kan man få plats med flera grafer i samma bild (de får överlappa varandra också). För att rita flera grafer i en bild måste man sätta Gnuplot i läget `multiplot`. Då kan man ha flera anrop till `plot` i samma skrift, som kommer att resultera i flera grafer på varandra.

Exemplet nedan kommer att resultera i två grafer, där den ena visar en uppförstoring i ett litet intervall.

```
# Script som visar exempel på multiplot
# Johan Wild 2008-09-15

unset multiplot
reset

set multiplot

set xrange[-10:10]
set yrange[-100:100]
set xtics 1
set ytics 20
set grid
set xzeroaxis lt 1
set yzeroaxis lt 1

plot (x-1)*(x-0.5)*(x+1) notitle

set size 0.3, 0.3
set origin 0.2, 0.6

set xrange[-2:2]
```

```

set yrange[-2:2]
set xtics 2.0
set ytics 2.0

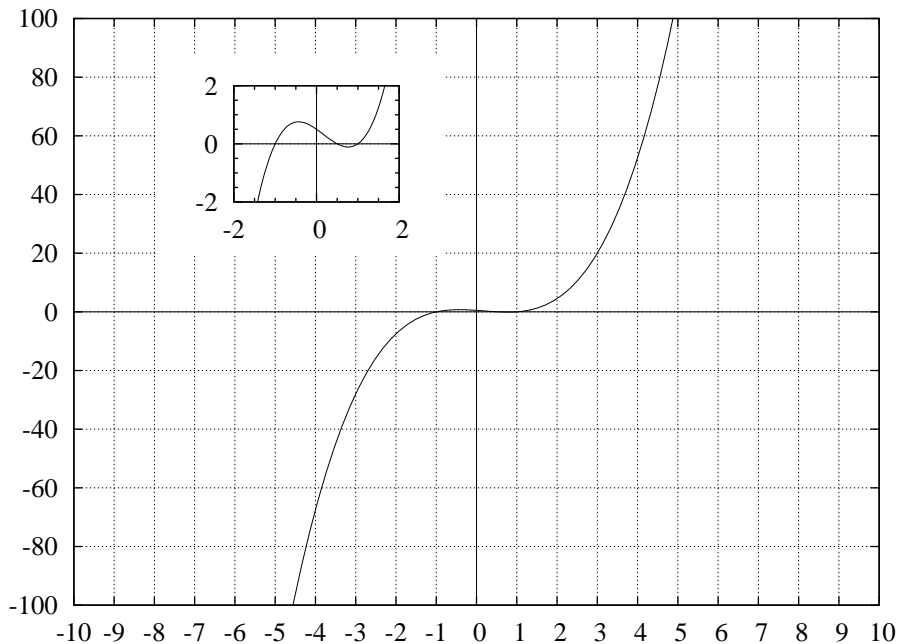
set mxtics 4
set mytics 4

clear

replot

unset multiplot

```



I detta script finns några detaljer att lägga märke till. Scriptet inleds med `unset multiplot`. Anledningen till det är att man inte kan göra vissa inställningar då Gnuplot är i läget `multiplot`. Om det blir i ett fel i ett script som sätter på detta läge innan felet uppstår, kommer Gnuplot förbli i detta läge.

Om nästa script som startas försöker ställa om något som inte går att ställa om i läget `multiplot` uppstår ett fel. Därför är det klokt att ställa om till normalläge med `unset multiplot` i början på scriptet. I just detta script görs förvisso inga sådana inställningar, men det är en bra vana att lägga till det.

Efter denna inledning sätts `multiplot` på och en graf ritas med vissa inställningar. Den första grafen ritas med standardinställningarna (eftersom inga inställningar görs), men inför den andra grafen ställs storlek och position om så att den upptar en lite mindre yta och placeras en bit upp till höger.

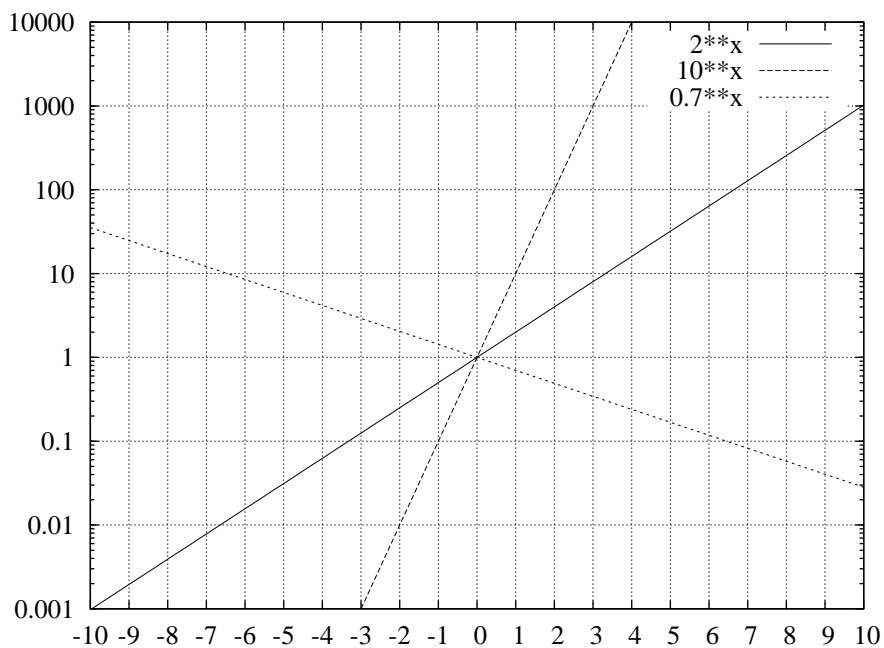
Övning 7.1. Gör ett script som använder `multiplot` för att generera flera grafer på samma bild.

8 Logaritmiska skalor

Exponentialfunktioner har egenskapen att dess graf spänner över värden som både ligger mycket nära noll (men ändå aldrig blir noll) till mycket stora värden.

För att åskådliggöra sådana funktioner kan det vara praktiskt att inte rita kurvan $y = f(x)$, utan $y = \log(f(x))$. Grafen blir lite mer lättläst för betraktaren om man ändå sätter ut det riktiga värdet på y -koordinaten på y -axeln. Gnuplot har en funktion för detta, vilket illustreras i scriptet nedan.

```
# Script som visar på användningen av logaritmiska skalor.  
# Johan Wild 2008-09-15  
  
reset  
  
set logscale y  
  
set xrange[-10:10]  
set yrange[0.001:10000]  
  
set xtics 1  
set ytics 10  
  
set mytics 10  
  
set grid  
  
plot 2**x , 10**x , 0.7**x
```



Övning 8.1. Skriv av scriptet ovan och modifiera det så att du får bättre känsla för exponentialfunktionen och logarimer. Till exempel kan du slå av `set logscale y` genom att ”kommentera bort” den raden genom att

sätta ett # i början på raden. Då måste du också ändra på `ytics`, annars blir det bara en gröt av y-axeln.

Notera också hur exponentialfunktioner med en bas som ligger mellan 0 och 1 kommer att bli en linje med negativ riktningskoefficient. De motsvarar också exponentiellt avtagande funktioner.

Testa också att ändra på värdet för `ytics`. Om du ändrar den till 8 kommer till exempel linjen $y = 2^x$ gå genom gridens linjer (varför?).

Varför går alla linjer genom samma punkt?

9 Att spara och skriva ut en graf

Detta är tyvärr en hel vetenskap i sig. I standardläge öppnas ett nytt fönster för varje anrop till plot som görs. Det enklaste sättet att skriva ut en graf är att klicka en gång i den ruta som finns längst upp till vänster i det fönstret (om man dubbelklickar stängs fönstret). Gör man detta visas en meny där man under Options kan välja Print.

Man kan också välja Copy to clipboard. Gör man det kan man infoga bilden i till exempel ett Word-dokument. Det blir tyvärr inte så bra kvalitet på den bilden. Vill man ha bättre kvalitet måste man trixa lite.

Gnuplot genererar grafen i något som kallas för en terminal. I standardläge är denna satt till `windows`. Genom att ändra terminal kan man få Gnuplot att spara grafen i många olika format.

Tyvärr beror grafens utseende lite på vilken terminal som används. I terminalen `windows` får man linjer av olika färg då man ändrar linjetyp. Gör man samma sak i en annan terminal kan man få streckade linjer av olika varianter istället. Även formateringen på kurvornas namn kan skilja sig åt.

Om du vill använda dina grafer i MS Word rekommenderar jag att du använder terminaltypen `postscript`. Nackdelen med denna är att vid versioner innan Office 2003 måste man skriva ut till en viss sorts skrivare (en skrivare som förstår postscript) för att det skall bli något vettigt resultat. Detta problem kan dock kringgås genom att man skriver ut till en virtuell skrivare som omvandlar till pdf (förutsatt att den virtuella skrivaren förstår postscript naturligtvis).

Efter att ha satt terminalen till `postscript` måste man ange vilket filnamn bilden skall få. Det gör man med `set output 'Filnamn.ps'`.

I terminalen `postscript` kan man använda grekiska bokstäver, och formatera text på diverse sätt. Det visas i exemplet nedan. Notera hur super- och subscript ange med `^` respektive `_`, som då x^2 skrivs som `x^2`. Det är också brukligt i matematiska texter att ha variabler i kursiv stil och namn på elementära funktioner i normal stil, som i $\sin(x)$.

De exempel som ges i scriptet får tjäna som hjälp för att förstå hur byte

av typsnitt och infogande av symboler går till.

Raden `set encoding iso_8859_1` syftar till att få Gnuplot att försöka hur de svenska tecknen (å, ä och ö) i scriptet skall tolkas.

Scriptet visar också hur man kan ange pilar och texter där man vill ha dem med `arrow` och `label`.

```
# Script som visar exempel på multiplot
# Johan Wild 2009-11-05

unset multiplot
reset
set encoding iso_8859_1

set terminal postscript eps enhanced 'Times-Roman' 20
set output 'Filnamn.ps'

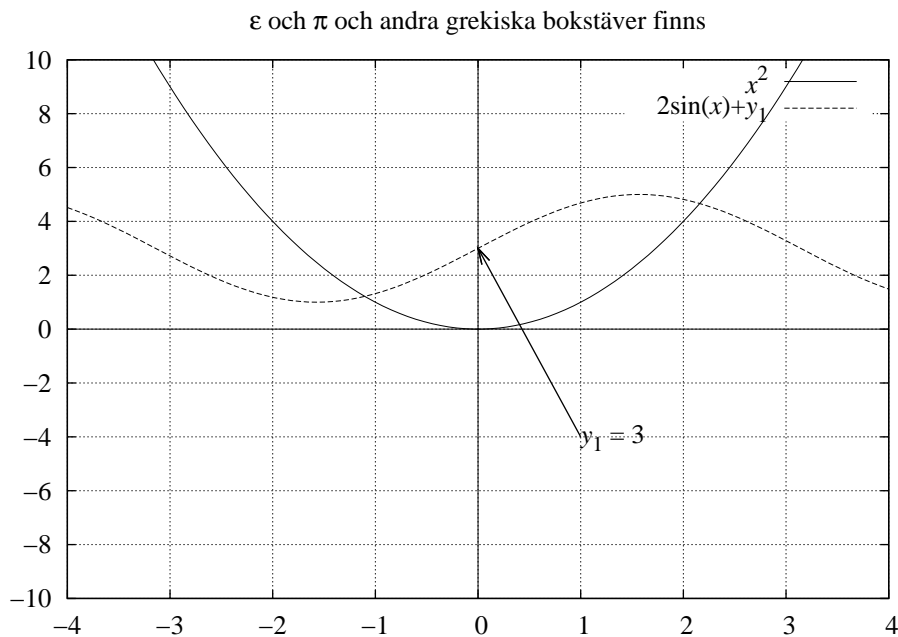
set title '{/Symbol e} och {/Symbol p} och andra grekiska bokstäver finns'

set xrange[-4:4]
set yrange[-10:10]
set xtics 1
set ytics 2
set grid
set xzeroaxis lt 1
set yzeroaxis lt 1

set arrow 1 from 1,-4 to 0,3
set label 1 '{/Times-Italic y}_1 = 3' at 1,-4

plot x**2 title '{/Times-Italic x}^2',\
      2*sin(x)+3 title '2sin({/Times-Italic x})+{/Times-Italic y}_1'

unset multiplot
```



Den mest komplicerade terminalen heter `latex`. Jag går inte in på hur man använder den, men det är i alla fall denna terminal som erbjuder alla

speciella matematiska symboler. Alla grafer i mina andra läromedel är gjorda med denna terminal.

Övning 9.1. Gör en graf och inkludera den i ett Word-dokument genom att lagra den på clip board och klistra in den i ett Word-dokument.

10 Kurvor på parameterform

När du lärde dig att rita grafen till en funktion $f(x)$ fick du lära dig att välja några värden på x , och räkna ut värdet för en funktion i dessa punkter. Sedan prickade du in punkterna $(x, f(x))$ i ett koordinatsystem.

Ett annat sätt att rita kurvor är att bestämma två funktioner som vardera får bestämma x - och y -koordinaterna för punkter. Ofta tänker man sig att något rör på sig i två dimensioner. Då beror både x - och y -koordinaten av tiden, och man brukar använda variabeln t som parameter.

Gnuplot kan rita kurvor på detta sätt också. Scriptet nedan är ett exempel på det. Kurvorna som ritas ges av

$$\begin{cases} x(t) = t^2 \\ y(t) = t \end{cases}$$

för $-2 < t < 2$ och

$$\begin{cases} x(t) = \cos(t) \\ y(t) = \sin(t) \end{cases}$$

för $0 < t < 270$.

```
# Script som exemplifierar en kurva på parameterform
# Johan Wild 2008-09-17
```

```
unset multiplot
reset
```

```
set xrange[-1:5]
set yrange[-3:3]
```

```
set grid
set xtics 1
set ytics 1
set xzeroaxis
set yzeroaxis
```

```
set multiplot
```

```
set parametric
set trange[-2:2]
```

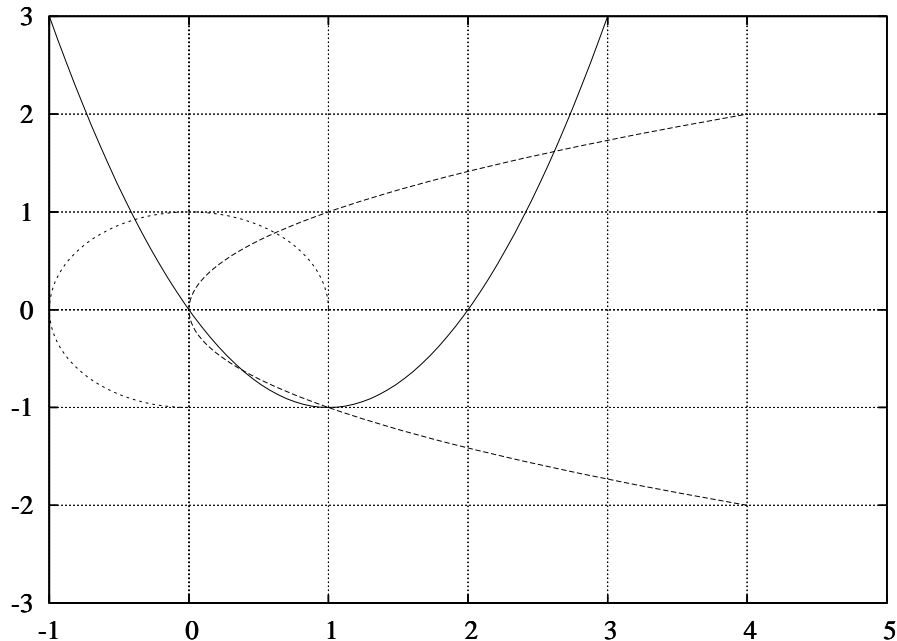
```
plot t**2,t lt 2 notitle
```

```
set trange[0:270]
set angles degrees
```

```
plot cos(t),sin(t) lt 3 notitle
```

```
unset parametric
plot (x-1)**2-1 notitle

unset multiplot
```



Först kommer några standardinställningar för att ställa in lämpliga intervall på det område som skall ritas och för att få en lämplig grid. Sedan sätts läget för kurvor på parameterform på med `set parametric`. Därefter specificeras det intervall på parametern `t` som vi vill använda i den första kurvan.

I läget `parametric` tar `plot` två uttryck som argument, ett för x - och ett för y -koordinaten. Båda skall vara uttryckta som funktion av `t`.

Nästa kurva (läget `multiplot` gäller!) ritas en kvarts cirkel. För denna kurva sätts alltså ett nytt värde på `trange`. Funktionerna $\cos(x)$ och $\sin(x)$ är trigonometriska funktioner. Gnuplot använder normalt radianer som enhet för vinklar, men det går att ställa om till grader med `set angles degrees`.

Man kan blanda kurvor av vanlig typ (graf till en funktion) med kurvor på parameterform i samma bild. Det är bara att slå av parameterläget med `unset parametric` och använda `plot` på vanligt sätt.

Övning 10.1. Gör ett script som ritas grafen till kurvor som ges av följande funktioner. Fundera på varför resultatet ser ut som det gör.

$$\begin{cases} x(t) = t^2 \\ y(t) = t^2 \end{cases} \quad \text{för } -2 < t < 1$$

$$\begin{cases} x(t) = t \\ y(t) = 2t \end{cases} \quad \text{för } -2 < t < 1$$

$$\begin{cases} x(t) = \cos(at) \\ y(t) = \sin(bt) \end{cases} \quad \text{för } 0 < t < 360$$

$$\begin{cases} x(t) = r(t) \cos(at) \\ y(t) = r(t) \sin(bt) \end{cases} \quad \text{för } 0 < t < 360, r(t) = 1 + 0.2 \sin(ct)$$

Lek med olika värden på a, b och c i de sista exemplen.

11 Nivåytor

De flesta funktioner du stött på hittills har varit funktioner av typen $f : \mathbb{R} \rightarrow \mathbb{R}$. Det finns dock funktioner av två variabler på formen $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. De tar alltså två argument (två tal) och ger ett tal.

Ett exempel på en sådan funktion är $f(x, y) = x + y$. Man kan i någon mening illustrera "grafnen" till dessa genom att tänka på (x, y) -planet som liggandes i botten, och rita funktionsvärdet som höjden över detta, $z = f(x, y)$. Grafen blir därför en yta.

Gnuplots funktion för att rita ytor heter `splot`. Nedan visas en bild med grafen till två funktioner. Notera hur scriptet är förenklat genom att bilda en radie-funktion $r(x, y) = \sqrt{x^2 + y^2}$, som sedan används i en av de funktioner som ritas (som då blir en sammansatt funktion!).

```
# Script som visar exempel på multiplot
# Johan Wild 2009-11-05

reset

set encoding iso_8859_1

set terminal postscript eps enhanced 'Times-Roman' 20
set output 'Script8.ps'

set xrange[-4:4]
set yrange[-4:4]
set zrange[-8:8]

set xtics 1
set ytics 1
set ztics 2

set ticslevel 0

set grid

set xzeroaxis lt 1
set yzeroaxis lt 1

set xlabel '{/Times-Italic x}'
```

```

set ylabel '{/Times-Italic y}'
set xlabel '{/Times-Italic z}'

set hidden3d

set isosamples 40

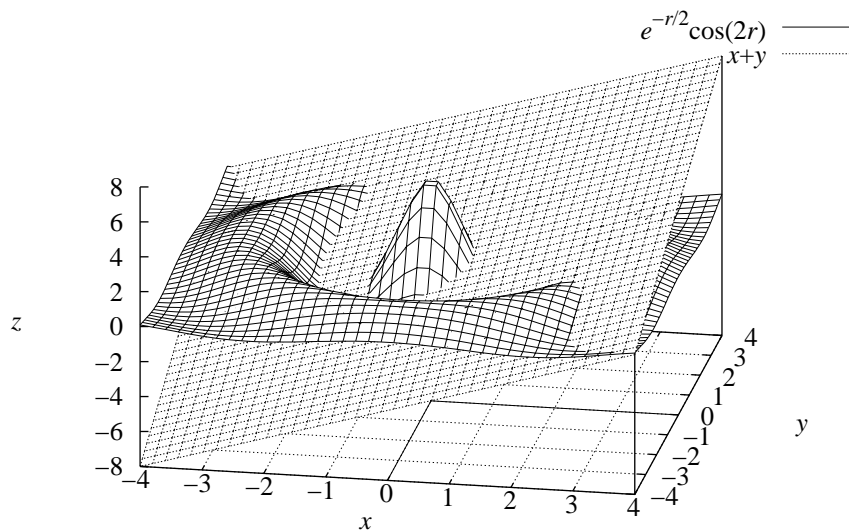
set view 60,10

r(x,y)=sqrt(x**2+y**2)

f(r)=5*exp(-r/2)*cos(2*r)

splot f(r(x,y)) title '{/Times-Italic e}^{-{/Times-Italic r}/2}cos(2{/Times-Italic r})',\
x+y title '{/Times-Italic x}+{/Times-Italic y}' lt 4

```



Övning 11.1. Skriv av scriptet ovan, men för terminalen windows (utlämna randerna `set terminal ...` och `set output`

Kommentera bort raden `set hidden3d` och ändra på värdet för `isosamples`. Konstatera vad som händer.

Ändra också på vinklarna i `set view` för att se vad som händer. I terminalen windows kan man ta tag i grafen och vrida den hur som helst. Kommandot `set view` är mer relevant för terminalen `postscript`.

12 Parametriserade ytor

Alla ytor kan inte bildas som i förra avsnittet. Precis som en cirkel inte går att bilda som grafen till någon funktion $y = f(x)$, kan man inte bilda en sfär med någon funktion $z = f(x, y)$. Däremot kan man skriva vissa ytor på parameterform, precis som det var möjligt med cirkel.

För att parametrisera en yta måste man använda två parametrar (det är naturligt eftersom den topologiska dimensionen av en yta är två!). Scriptet nedan visar parametriseringen av en sfär.

```
# Script som visar exempel på multiplot
# Johan Wild 2009-11-05

reset

set encoding iso_8859_1

set terminal postscript eps enhanced 'Times-Roman' 20
set output 'Script9.ps'

set xrange[-4:4]
set yrange[-4:4]
set zrange[-4:4]

set xtics 1
set ytics 1
set ztics 2

set ticslevel 0

set grid

set xzeroaxis lt 1
set yzeroaxis lt 1

set xlabel '{/Times-Italic x}'
set ylabel '{/Times-Italic y}'
set zlabel '{/Times-Italic z}'

set isosamples 40
set hidden3d

set view 60,30

set angles degrees

set parametric

R=2

r(x,y)=sqrt(x**2+y**2)

f(x)=x/180

set urange[0:360]
set vrange[0:180]

plot R*cos(u)*sin(v),R*sin(u)*sin(v),R*cos(v)+2 notitle,\
      4*R*cos(u*3/4-10)*f(v), 4*R*sin(u*3/4-10)*f(v),f(v*6)**2-4 notitle

unset parametric
```

