

Krzysztof Trajkowski

Gnuplot

24 listopada 2013

Wprowadzenie

Gnuplot jest wolnym, wieloplatformowym oprogramowaniem do rysowania wykresów matematycznych oraz do graficznej prezentacji już obróbianych danych np. zapisanych w pliku. Ta aplikacja ma tylko kilka wbudowanych funkcji do obróbki danych np. krzywe Beziera, interpolacja (natural cubic splines algorithm), aproksymacja tzw. fitowanie (Marquardt-Levenberg algorithm). Warto też wspomnieć, że za pomocą funkcji **stats** możemy obliczyć podstawowe statystyki tj. min., max., średnia, kwartyle. Wiele aplikacji wykorzystuje gnuplota do grafiki np. Octave – pakiet do obliczeń numerycznych, Maxima – pakiet do obliczeń symbolicznych, Gretl – pakiet do estymacji modeli ekonometrycznych. Warto podkreślić, że Gretl daje możliwość zapisu wykresu do pliku w formie poleceń gnuplota. Dzięki tej opcji możemy dowolnie modyfikować parametry wykresów. Dodatkowo w Gretlu możemy pisać i uruchamiać pliki skryptowe gnuplota. Warto więc poznać język komend tego programu graficznego.

1. Gnuplot a L^AT_EX

Gnuplot ma wiele formatów wyjściowych np. PostScript, GIF, EMF, SVG, PDF lub PNG. Aby zobaczyć wszystkie jakie są dostępne wystarczy w konsoli programu wpisać poniższą komendę.

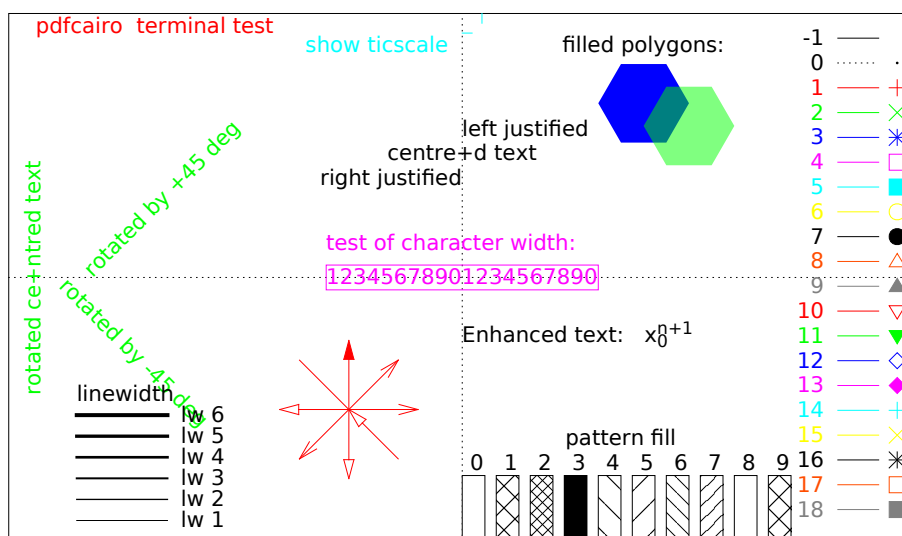
```
set terminal
```

Więcej informacji o ustawieniach danego terminala można znaleźć w dokumentacji

```
help terminal pdf
```

Wiele graficznych możliwości wybranego wcześniej terminala możemy zobaczyć na rysunku, wpisując w konsoli poniższą komendę.

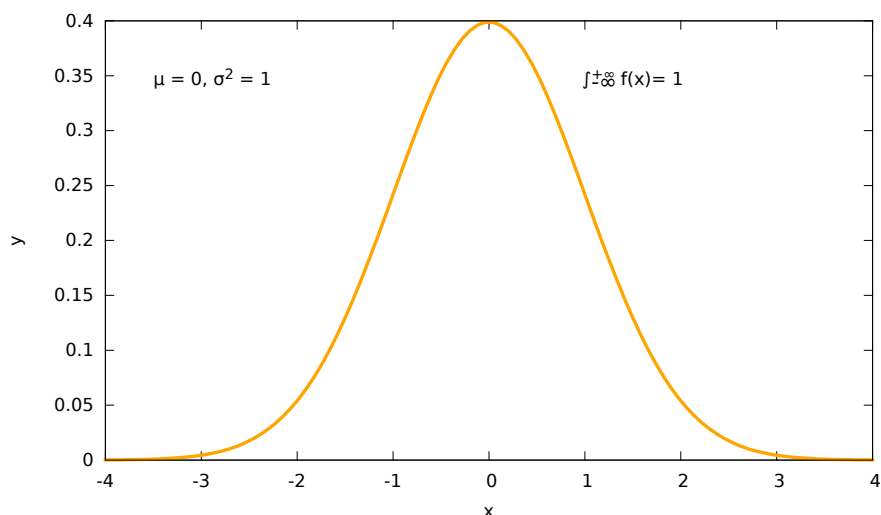
```
set terminal pdf
test
```



Rysunek 1.1. Terminal PDF.

Dla osób składających swoje dokumenty w \LaTeX -u ciekawym rozwiązaniem jest użycie pakietu `gnuplottex` dzięki któremu możemy umieszczać kod gnuplota w pliku \LaTeX -a.

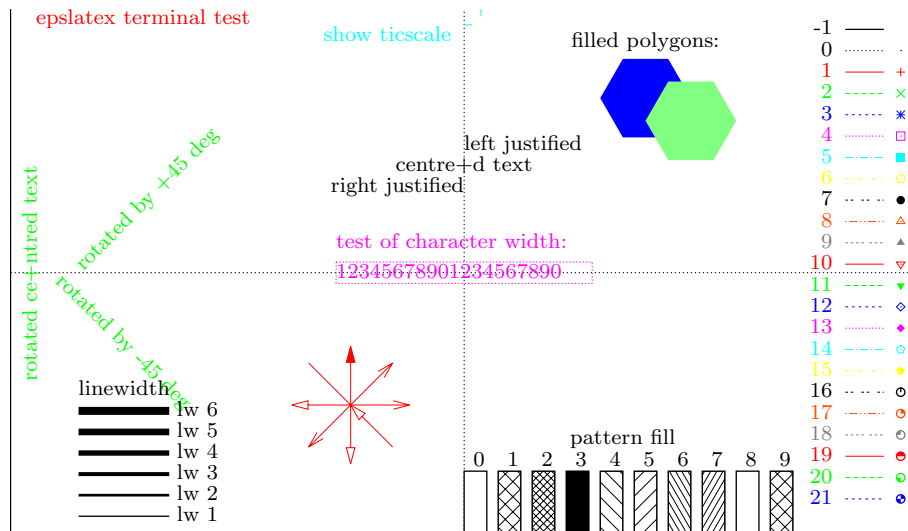
```
\begin{figure}[!ht]
\begin{center}
\begin{gnuplot}[terminal=pdf,terminaloptions={enhanced font "Times-Roman,10" size 10cm,6cm}]
# tutaj wpisujemy kod gnuplota:
set xrange [-4:4]
set xlabel "x"
set ylabel "y"
set label "{/Symbol m} = 0, {/Symbol s}^2 = 1" at -3.5, 0.35 left
set label "{/Symbol \362@_{/=12 - \245}^{\/=8 + \245}} f(x)= 1" at 1.5, 0.35 center
Gauss(x,mu,sigma) = 1./(sigma*sqrt(2*pi)) * exp( -(x-mu)**2 / (2*sigma**2) )
plot Gauss(x,0,1) t "" lc rgb "orange" lw 5
\end{gnuplot}
\end{center}
\end{figure}
```



Rysunek 1.2. Standardowy rozkład normalny – terminal PDF.

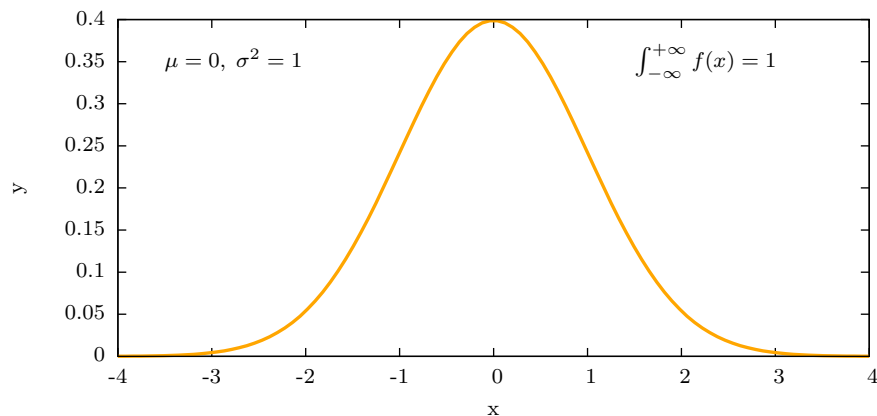
Jeśli na wykresie chcemy umieszczać wzory matematyczne lub symbole greckie używanie terminala PDF lub EPS nie jest optymalnym rozwiązaniem – Rysunek 1.2. Jednym z rozwiązań tego problemu może być modyfikacja wykresu tzn. podmiana ciągu znaków (np. symbole matematyczne) w pliku EPS na \LaTeX -owe. Można w tym celu wykorzystać pakiet `psfrag`. Jednak lepszym rozwiązaniem jest wykorzystanie terminala `epslatex`. Jest to dobra alternatywa dla zestawu makr `PSTricks`.

```
set terminal epslatex
test
```



Rysunek 1.3. Terminal epslatex.

```
\begin{gnuplot}[scale=1,terminal=epslatex,terminaloptions={font 8 ←
color colortext size 12cm,5.5cm}]
set xrange [-4:4]
set xlabel "x"
set ylabel "y"
set label "$\mu=0, \sigma^2=1$" at -3.5, 0.35 left
set label "$\int_{-\infty}^{+\infty} f(x)=1$" at 1.5, 0.35 left
Gauss(x,mu,sigma) = 1./(sigma*sqrt(2*pi)) * exp( -(x-mu)**2 / (2*←
sigma**2) )
plot Gauss(x,0,1) t "" lc rgb "orange" lw 5
\end{gnuplot}
```



Rysunek 1.4. Standardowy rozkład normalny – terminal epslatex.

2. Wykresy danych z pliku

2.1. Szereg czasowy

Pierwszy przykład to prezentacja kształtowania się cen otwarcia indeksu giełdowego WIG20 oraz ich wygładzenie krzywą Beziera.



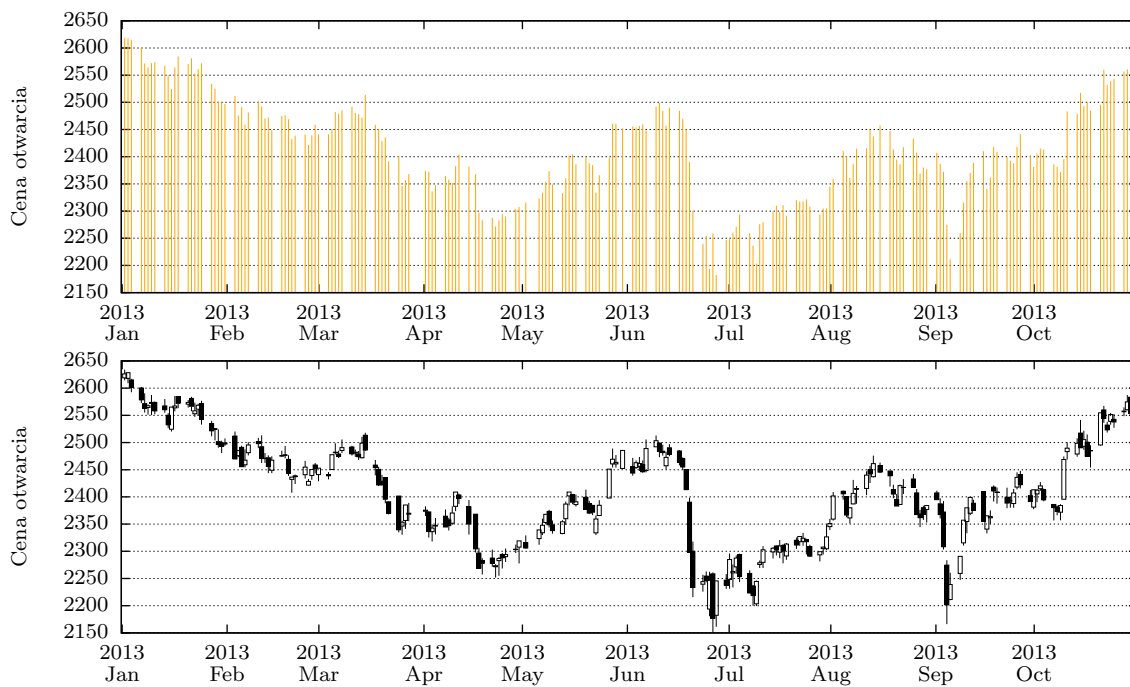
Rysunek 2.1. Cena otwarcia indeksu WIG-20 w okresie od 01-01-2012 do 01-10-2013.

Lista komend gnuplota jaką wykorzystaliśmy do wygenerowania tego wykresu jest zamieszczona poniżej.

```
set xdata time
set timefmt "%Y-%m-%d"
set format x "%Y\n%b"
set datafile separator ","
set xrange ["2011-12":"2013-12"]
set mxtics 3
set ylabel "Cena otwarcia"
set bmargin 0
set grid
plot "/home/krz/Pulpit/GNUwykres/DATA/wig20_d.csv" u 1:2 with lines \
title columnhead lc rgb "orange" lw 5, "/home/krz/Pulpit/\
GNUwykres/DATA/wig20_d.csv" u 1:2 smooth sbezier title "krzywa \
Bezier" lc rgb "black" lw 1 lt 1
```

Poniżej kolejny przykład wykresów giełdowych.

```
set xdata time
set timefmt "%Y-%m-%d"
set format x "%Y\n%b"
set datafile separator ","
set xrange ["2013-01":"2013-11"]
set bmargin 2
unset key
set grid y
set ylabel "Cena otwarcia"
set mxtics 0
set multiplot
set size 1, 0.5
set origin 0, 0.5
plot "/home/krz/Pulpit/GNUwykres/DATA/wig20_d.csv" u 1:2 with \
impulses lc rgb "orange" lw 1 lt 1
set origin 0, 0
plot "/home/krz/Pulpit/GNUwykres/DATA/wig20_d.csv" u 1:2:4:3:5 with \
candlesticks lc rgb "black"
```

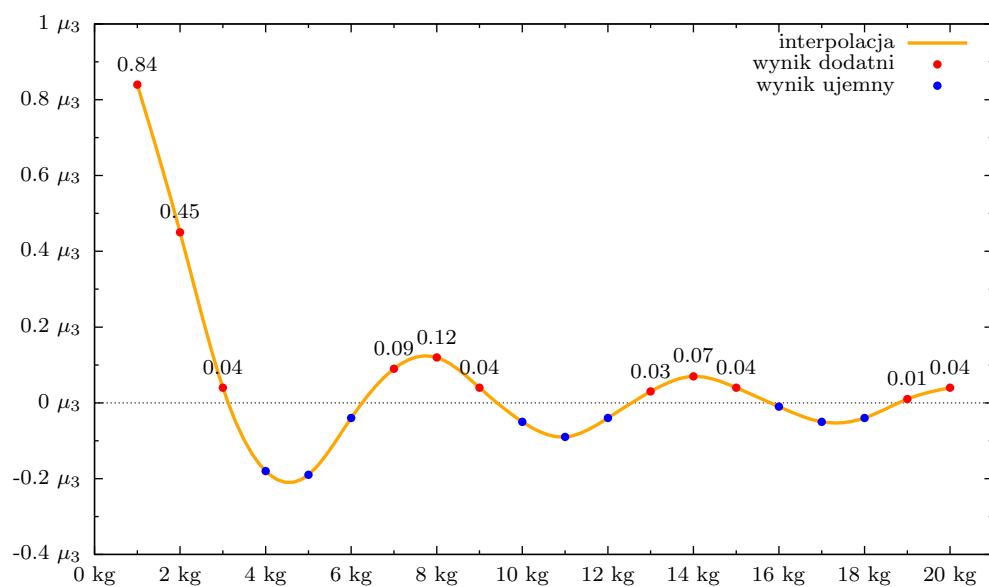


Rysunek 2.2. Cena otwarcia indeksu WIG-20 w okresie od 01-01-2012 do 01-10-2013.

```
unset multiplot
```

2.2. Dane doświadczalne

Gnuplot bardzo często jest wykorzystywany do prezentacji danych doświadczalnych w takich naukach jak: fizyka czy chemia. Poniżej kilka przykładowych wykresów.

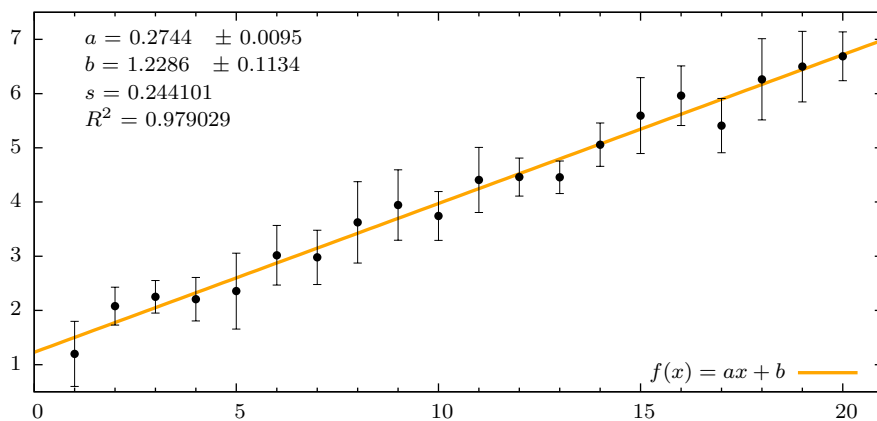


Rysunek 2.3. Interpolacja.

```

# Interpolacja
set datafile separator ","
set bmargin 0.5
set format x "%g kg"
set format y "%g $\mu_3$"
set mxtics 2
set mytics 2
set xtics 2
set xrange [0:21]
set xzeroaxis
p "/home/krz/Pulpit/GNUwykres/DATA/a.csv" u 2:3 t "interpolacja" \
smooth csplines lw 5 lc rgb "orange" lt 1, "/home/krz/Pulpit/\
GNUwykres/DATA/a.csv" u 2:($3>=0 ? $3:1/0) t "wynik dodatni" w \
points pt 7 lc rgb "red", "/home/krz/Pulpit/GNUwykres/DATA/a.csv" \
u 2:($3<0 ? $3:1/0) t "wynik ujemny" w points pt 7 lc rgb "blue" \
", "/home/krz/Pulpit/GNUwykres/DATA/a.csv" u 2:($3>0 ? ($3+0.05) \
:1/0 ):3 t "" smooth uniq w labels

```

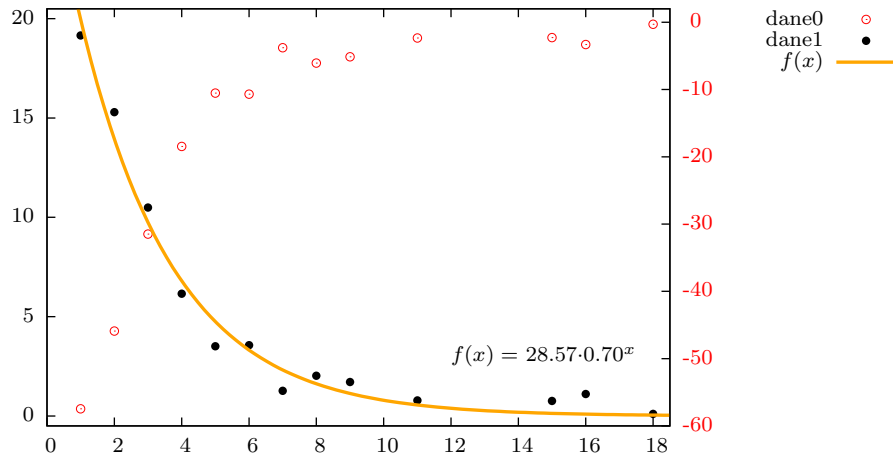


Rysunek 2.4. Regresja liniowa.

```

# Regresja liniowa:
set bmargin 0.5
set datafile separator ","
set key right bottom
set fit errorvariables
set xrange [0:21]
set yrange [0.5:7.5]
set mxtics 5
f(x)=a*x+b
a=1;b=1
fit f(x) "/home/krz/Pulpit/GNUwykres/DATA/b.csv" u 1:2 via a,b
print "error of a is:", a_err
print "error of a is:", b_err
set label "$a$ = %6.4f", a, "$\pm$ %6.4f", a_err at 1.25,7.0
set label "$b$ = %6.4f", b, "$\pm$ %6.4f", b_err at 1.25,6.5
set label "$s$ = %10.6f", FIT_STDFIT at 1.25,6.0
stat "/home/krz/Pulpit/GNUwykres/DATA/b.csv" u 1:2
set label "$R^2$ = %1.6f", STATS_correlation**2 at 1.25,5.5
p f(x) title "$f(x)=ax+b$" lc rgb "orange" lw 5, "/home/krz/Pulpit/\
GNUwykres/DATA/b.csv" u 1:2:3 with yerrorbars title "" lt 1 pt 7 \
lc rgb "black"

```



Rysunek 2.5. Regresja nieliniowa.

```
# Regresja nieliniowa
set bmargin 0.5
set key outside
set fit errorvariables
set xrange[0:18.5]
set yrange[-0.5:20.5]
set y2range[-60:2]
set datafile separator ","
set y2tics tc rgb "red"
set ytics nomirror
f(x)=a*b**x
a=1;b=1
fit f(x) "/home/krz/Pulpit/GNUwykres/DATA/bb.csv" u 1:2 via a,b
set label "$f(x)=$ %1.2f",a, "$\\cdot$%1.2f$^x$", b at 12,3
p "/home/krz/Pulpit/GNUwykres/DATA/bb.csv" u 1:(-3*$2) axis x1y2 w \
points title "dane0" lt 6 lc rgb "red", "/home/krz/Pulpit/\
GNUwykres/DATA/bb.csv" u 1:2 axis x1y1 w points title "dane1" lt \
7 lc rgb "black", f(x) axis x1y1 title "$f(x)$" lc rgb "orange" \
lw 5 lt 1
```

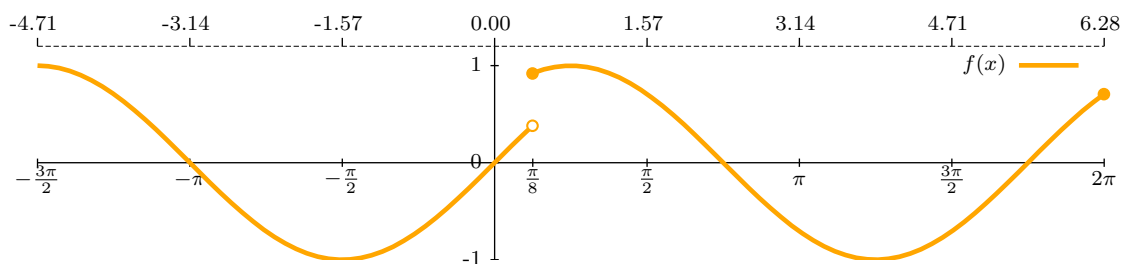
3. Wykresy matematyczne

3.1. Funkcje

Gnuplot bardzo dobrze sprawdza się w rysowaniu funkcji matematycznych.

$$f(x) = \begin{cases} \sin(x) & x < \frac{\pi}{8} \\ \cos\left(x - \frac{\pi}{4}\right) & x \in \left[\frac{\pi}{8}; 2\pi\right) \end{cases}$$

```
set xrange [-3*pi/2:2*pi]
set yrange [-1:1.2]
set ytics 1
set xtics 1
set size ratio 0.2000805
set xzeroaxis lt 1 lw 2 lc rgb 'black'
```

Rysunek 3.1. Wykres funkcji $f(x)$.

```
set yzeroaxis lt 1 lw 2 lc rgb 'black'
set border 0
set xtics axis
set ytics axis
set xtics ('$-\frac{3\pi}{2}$' -3*pi/2, '$-\pi$' -pi, \
'$-\frac{\pi}{2}$' -pi/2, '$\frac{\pi}{2}$' pi/2, '$\pi$' pi, \
'$\frac{3\pi}{2}$' 3*pi/2, '$2\pi$' 2*pi, '$\frac{\pi}{8}$' pi/8) \
scale 0.5 offset 0,0.4
set ytics scale 0.5
set format xy '%.0f'
set format x2 '%.2f'
set x2tics scale 1 offset -0.3,-0.4
set x2tics out pi/2
set arrow from -1.5*pi,1.2 to 2*pi,1.2 nohead lt 2 lc 0
set key right top
set bmargin 0
set lmargin 0
set rmargin 0
plot (x<0.4)?sin(x):1/0 t '$f(x)$' lc rgb 'orange' lw 7, \
(x>=0.39 && x<=2*pi)?cos(x-pi/4):1/0 t '' lc rgb 'orange' lw 7 lt ←
1, \
"<echo '0.39 0.38'" w points pt 7 lc rgb 'orange' ps 1.5 t "", \
"<echo '0.39 0.38'" w points pt 7 lc rgb 'white' ps 1 t "", \
"<echo '0.39 0.92'" w points pt 7 lc rgb 'orange' ps 1.5 t "", \
"<echo '6.28 0.707'" w points pt 7 lc rgb 'orange' ps 1.5 t ""
```

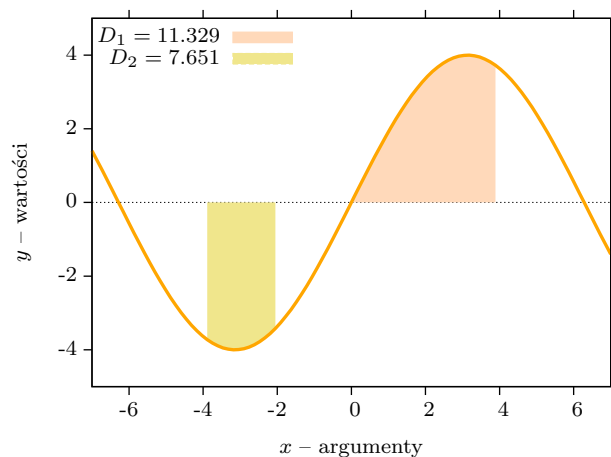
Kolejny przykład to zaznaczenie dwóch pól D_1 i D_2 pod wykresem danej funkcji.

$$D_1 = \int_0^4 4 \sin\left(\frac{x}{2}\right) dx = 4 [2 - 2 \cos(2)] \approx 11.329$$

$$D_2 = \int_{-2}^{-4} 4 \sin\left(\frac{x}{2}\right) dx = -8 [\cos(2) - \cos(1)] \approx 7.651$$

```
set xzeroaxis
set key top left
set ylabel "$y$ -- wartości"
set xlabel "$x$ -- argumenty"
set xrange [-7:7]
set yrange [-5:5]
set tics out
set xtics offset 0,0.25
set ytics offset 0.5,0
set size ratio 0.71
```

```
plot (x>=0 && x<=4)?4*sin(x/2):1/0 with filledcurve y1=0 lc rgb "FFDAB9" t "$D_{1}=11.329$", (x>=-4 && x<=-2)?4*sin(x/2):1/0 with filledcurve y1=0 lc rgb "F0E68C" t "$D_{2}=7.651$", 4*sin(x/2) lc rgb "orange" lw 5 lt 1 t ""
```



Rysunek 3.2. Całka oznaczona.

3.2. Krzywe

Kilka przykładów rysowania krzywych opisanych za pomocą równań parametrycznych.

- równanie parametryczne okręgu o postaci $x^2 + y^2 = 1$:

$$\begin{cases} x = \sin t \\ y = \cos t \end{cases} \quad t \in \langle 0, 2\pi \rangle$$

- równanie parametryczne elipsy o postaci $16x^2 + y^2 = 16$:

$$\begin{cases} x = \frac{1}{2} \sin t \\ y = 4 \cos t \end{cases} \quad t \in \langle 0, 2\pi \rangle$$

- równanie parametryczne hiperboli o postaci $9x^2 - 2.25y^2 = 36$:

$$\begin{cases} x = \pm 2 \cosh t \\ y = \pm 4 \sinh t \end{cases} \quad t \in \mathcal{R}$$

- równanie parametryczne paraboli o postaci $x = 0.1y^2 - 3$:

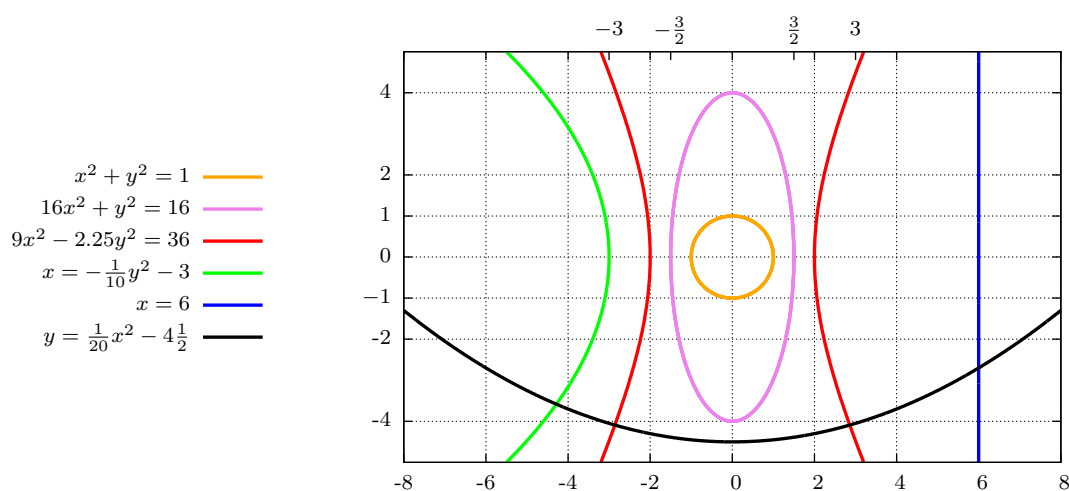
$$\begin{cases} x = \frac{1}{10}t^2 - 3 \\ y = t \end{cases} \quad t \in \mathcal{R}$$

- równanie parametryczne linii o postaci $x = 6$:

$$\begin{cases} x = 6 \\ y = t \end{cases} \quad t \in \mathcal{R}$$

- równanie parametryczne wielomianu o postaci $y = 0.05x^2 - 4.5$:

$$\begin{cases} x = t \\ y = \frac{1}{20}t^2 - 4\frac{1}{2} \end{cases} \quad t \in \mathcal{R}$$



Rysunek 3.3. Wykresy.

```

set parametric
set bmargin 0
set grid
set sample 1000
set key outside left center spacing 1.5
set size ratio 0.625
set x2tics ("$\frac{3}{2}$" 1.5,"$-\frac{3}{2}$" -1.5,"$-3$" -3,"↔
    "$3$" 3)
set ytics add ("-$1$" -1,"$1$" 1)
plot [-3*pi:3*pi] [-8:8] [-5:5]\
sin(t),cos(t) lc rgb "orange" lt 1 lw 5 t "$x^2+y^2=1$",\
1.5*sin(t),4*cos(t) lc rgb "violet" lt 1 lw 5 t "$16x^2+y^2=16$",\
2*cosh(t),4*sinh(t) lc rgb "red" lt 1 lw 5 t "$9x^2-2.25y^2=36$",\
-2*cosh(t),-4*sinh(t) lc rgb "red" lt 1 lw 5 t "",\
-0.1*t**2-3,t lc rgb "green" lt 1 lw 5 t "$x=-\frac{1}{10}y^2-3$",\
6,t lc rgb "blue" lt 1 lw 5 t "$x=6$",\
t, 0.05*t**2-4.5 lc rgb "black" lt 1 lw 5 t "$y=\frac{1}{20}x^2-4\frac{1}{2}$"
unset parametric

```

Uwagi końcowe

Każdy wykres można zawsze dowolnie formatować modyfikując jego domyślne ustawienia. Poniżej zostanie wypisanych kilka najbardziej przydatnych i najczęściej używanych ustawień tak aby dostosować go do indywidualnych potrzeb.

```

# nazwa osi X:
set xlabel "nazwa"
# zakres osi X:

```

```
set xrange [-3:5]
# częstotliwość głównej podziałki na osi X:
set xtics 1
# częstotliwość pomocniczej podziałki na osi X:
set mxtics 4
# skala logarytmiczna na osi X:
set logscale x
# pozycja legendy - z lewej strony, na górze oraz poza wykresem:
set key outside left top
# pozycja tekstu:
set label "tekst" at 1,2 left
# tytuł wykresu:
set title "tytuł"
# siatka wykresu tylko dla osi X:
set grid xtics
# rozmiar górnego marginesu:
set tmargin 0
# kolor osi, podziałki i czcionki osi:
set border 31 lc rgb "blue"
```

Wiele bardzo ciekawych materiałów na temat gnuplota można znaleźć pod adresem www.gnuplotting.org. Także na stronie domowej programu www.gnuplot.info są prezentowane ciekawe przykłady oraz jego dokumentacja. Zaznaczmy, że z poziomu konsoli gnuplota także mamy dostęp do opisu wybranej funkcji, wystarczy wpisać komendę `help` a następnie parametr jaki nas interesuje.

```
# opis funkcji fitowania - dopasowywanie krzywej:
help fit
```