Podstawy analizy danych

gnuplot

Wykład 3

Janusz Andrzejewski

Plan

- Wyrażenia arytmetyczne
- Funkcje
  - Definiowanie własnych funkcji
  - Wbudowane funkcje
- Tworzenie pliku
- Zarządzanie opcjami
- Dopasowanie liniowe
- Dopasowanie nieliniowe

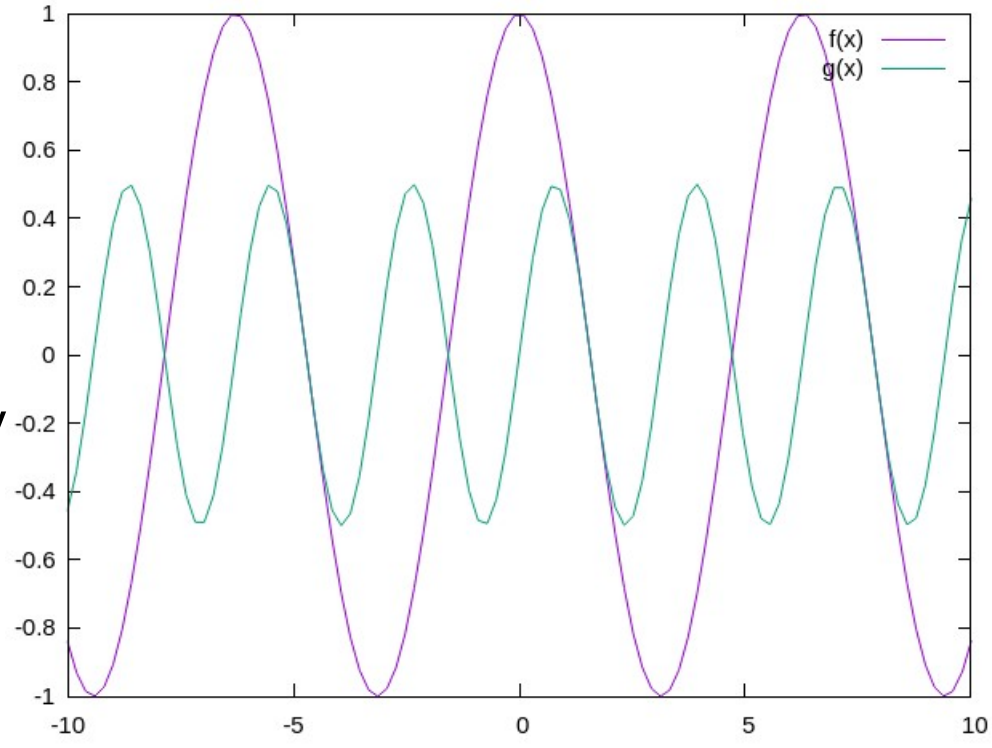
Wyrażenia arytmetyczne & funkcje własne

gnuplot umożliwia obliczanie wyrażeń i definiowanie własnych funkcji

```
gnuplot> A=3  # parametr użytkownika
gnuplot> a=2  #liczba calkowita, rozrozniamy wielkosc liter
gnuplot> print A/a  #komenda drukujaca wartosc wyrazenia
1
gnuplot> b=2.0  #liczba rzeczywista
gnuplot> print A/b  #Zwroc uwage na roznice pomiedzy A/a oraz A/b
1.5
gnuplot>
```

```
gnuplot> f(x)=cos(x)  #definicja funkcji
gnuplot> g(x)=sin(x)*cos(x)
gnuplot> plot f(x), g(x)
gnuplot>
```

Podczas definiowania zmiennych istotne jest to, czy przypisując wartość zmiennej w momencie jej deklarowania użyjemy kropki czy nie. W pierwszym przypadku zdefiniujemy zmienną rzeczywistą, w drugim całkowitą. Wpłynie to oczywiście na wyniki operacji na zmiennych.



Definicja funkcji

- Funkcje mogą mieć do 10 zmiennych i mogą zawierać inne funkcje i operatory.

```
f(x) = -x* log(x)

min( a, b ) = ( a < b ) ? a : b

binom( n, k ) = n!/(k!*(n-k)!)

step( x ) = ( x < 0 ) ? 0 : 1
```

- gnuplot domyślnie zakłada, że zmienna niezależna, tzw. dummy, oznaczona jest przez x.
- Można to zmienić za pomocą polecenia: `set dummy t` teraz t przejmuje rolę zmiennej niezależnej x.

```
plot [-2*pi:2*pi] sin(x)
set dummy t #zmiana zmiennej niezależnej x na t
plot [-2*pi:2*pi] sin(t)
```

- Wszystkie inne zmienne występujące w definicji funkcji (parametry) muszą mieć przypisane wartości przed próbą ich kreślenia.

```
g(x) = cos(a*x)/a

plot a=1, g(x), a=2, g(x)
```

- Wszystkie funkcje i zmienne mają zasięg globalny.

Funkcje wbudowane

- Funkcje matematyczne: abs, acos, acosh, arg, asin, asinh, atan, atan2, atanh, besj0, besj1, besy0, besy1, ceil, cos, cosh, erf, erfc, exp, floor, gamma, ibeta, inverf, igamma, imag, invnorm, int, lgamma, log, log10, norm, rand, real, sgn, sin, sinh, sqrt, tan, tanh.
- Liczby zespolone: {1,0}.
- Operator potęgowania: **.
- Operator „silnia”: !.
- „Numer kolumny”: $, np: p ’x.txt’ u ($1):($3).
- Pozostałe operatory jak w języku C: !, ~, & (binarne i), | (binarne lub), *, % (reszta z dzielenia - modulo), /, +, -, ==, !=, <, <=, >, >=, && (logiczne i), || (logiczne lub).
- Operator ? : (trójargumentowy) czyli np.: a ? b : c (jeśli a jest prawdziwe to wartością wyrażenia jest b, w przeciwnym przypadku wartością wyrażenia jest c).

```
plot [0:2*pi] sin(x)/(sin(x) >= 0)
```

Wartością wyrażenia logicznego jest 1 (prawda) lub 0 (fałsz). Gnuplot pomija punkty o nieokreślonej wartości, np. 1/0.

```
p [0:2*pi] [-1:1] (abs(sin(x)) <= 0.9) ? sin(x) : sgn(sin(x))*0.9  # ciekawa funkcja

p rand(0) w p  # rand(0) funkcja losowa, daje liczby z przedzialu [0,1]
```

Komenda set table

- Komenda set table daje nam dostęp do danych, które tworzy wykres w postaci tekstu, czyli uzyskujemy wartości wszystkich punktów przedstawionych na wykresie jako wyrażenia numeryczne.
- Aby wygenerować dane wyjściowe w postaci pliku tekstowego zawierającego liczby, a nie w postaci graficznej, należy zastosować polecenie:
  set table "nazwa_pliku" #dane beda zapisane do pliku nazwa_pliku
- Komenda set table bez nazwy pliku wypisze dane na konsole
- Aby przywrócić generowanie wykresów należy wydać komendę:
  unset table

Zapis do pliku danych

```
set table "cosinus.txt"
plot [0:2*pi] cos(x)
```

Wygeneruje plik o nazwie ‘cosinus.txt’, o zawartości:

```
# Curve 0 of 1, 1000 points
# Curve title: "cos(x)"
# x y type
 0  1  i
 0.00628947  0.99998  i
```

- Trzecia kolumna zawiera znacznik wskazujący, czy dany punkt danych był w zakresie danych lub poza nim:
  - i - „w zakresie”,
  - o - „poza zakresem”,
  - u - punkt jest niezdefiniowany.

Zarządzanie opcjami

- gnuplot ma stosunkowo niewiele poleceń (np. plot, save, load, print), ale wyposażony jest w dużą liczbę opcji.
- Opcje używane są do kontroli wszystkiego, od formatu przecinka do nazwa pliku wyjściowego.
- Do manipulowania poszczególnymi opcjami służą następujące polecenia:
  - show (skrucona forma sh)- wyświetla aktualną wartość opcji
    - Opcja all - pokazuje wszystkie możliwe opcje wraz z ich wartościami domyślnymi.
    - Opcja version long – podaje info nt. kompilacji gnuplota
    - Opcja style - info o sposobie rysowania funkcji i danych z pliku
    - Opcja functions – info o zdefiniowanych przez użytkownika funkcjach
    - Opcja variables – info o zdefiniowanych przez użytkownika i wbudowanych zmiennych
  - set - zmienia wartość opcji
  - unset - wyłącza konkretną opcję lub przywraca wartości domyślne np.: unset tics
  - reset - przywraca wartości domyślne dla wszystkich opcji, za wyjątkiem opcji terminala i wyjścia.

Zarządzanie opcjami

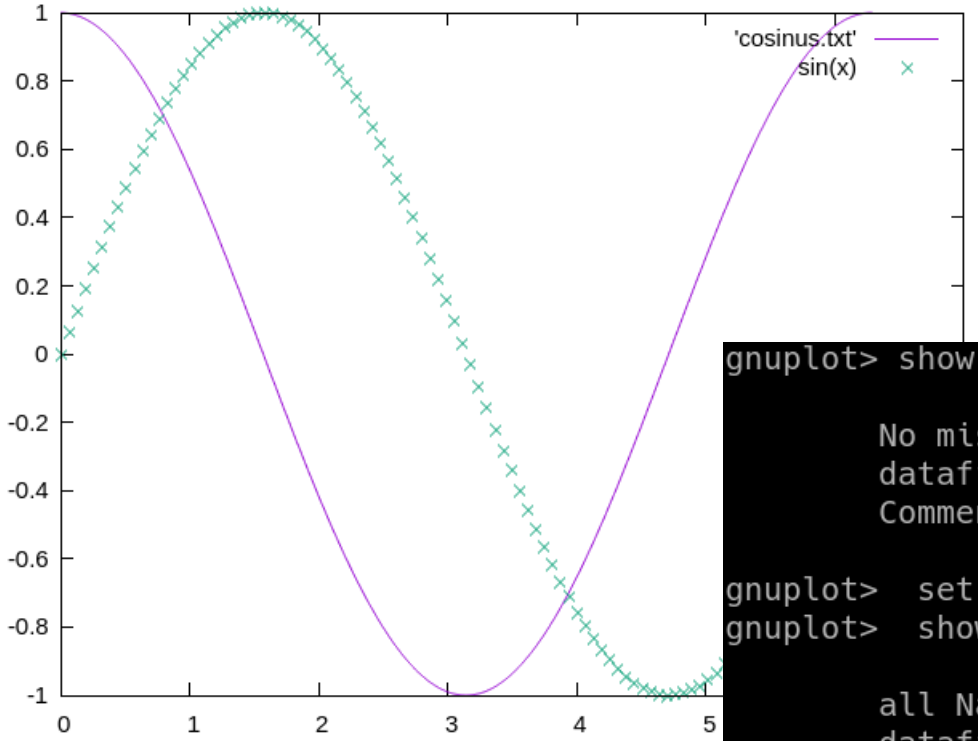
```
gnuplot> show style function #jaki jest domyślny styl rysowania funkcji

        Functions are plotted with lines

gnuplot> set style function points #od teraz styl rysowania funkcji to punkty
gnuplot> show style data #jaki jest domyślny styl rysownia danych z pliku

        Data are plotted with points

gnuplot> set style data lines #od teraz styl rysowania danych z pliku to linia
gnuplot> plot 'cosinus.txt', sin(x)
```




```
gnuplot> show datafile # sposob wczytywania danych z pliku

        No missing data string set for datafile
        datafile fields separated by whitespace
        Comments chars are "#"

gnuplot>  set datafile missing "NaN"
gnuplot>  show datafile

        all NaN (not-a-number) values will be treated as missing data
        datafile fields separated by whitespace
        Comments chars are "#"
```

Fitowanie

Polecenie fit umożliwia dopasowanie zdefiniowanej przez nas funkcji do zbioru danych pomiarowych. Jego działanie oparte jest na metodzie najmniejszych kwadratów przy użyciu algorytmu Marquardt-Levenberga.

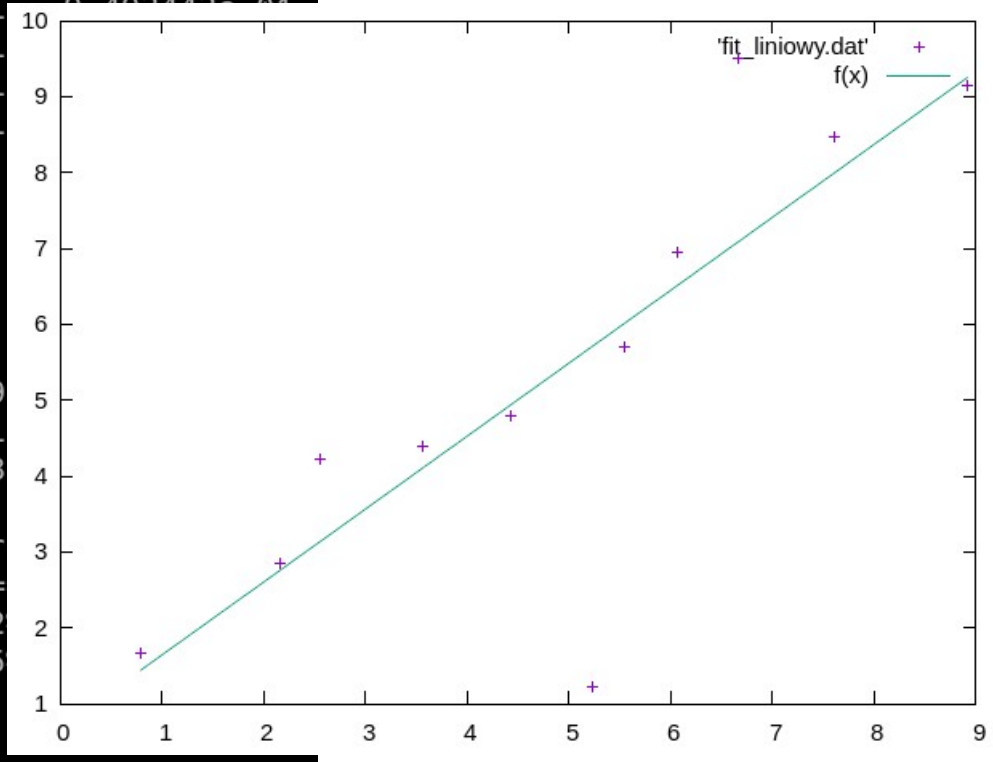
```
gnuplot> f(x)=a*x+b #definicja funkcji liniowej z parametrami a i b
gnuplot> fit f(x) 'fit_liniowy.dat' via a, b
iter      chisq       delta/lim  lambda   a             b
   0 3.0705600000e+01   0.00e+00  3.88e+00    1.000000e+00   1.000000e+00
   1 2.7924550705e+01  -9.96e+03  3.88e-01    9.200856e-01
   2 2.7767384991e+01  -5.66e+02  3.88e-02    9.586003e-01
   3 2.7766641189e+01  -2.68e+00  3.88e-03    9.617811e-01
   4 2.7766641188e+01  -1.51e-06  3.88e-04    9.617835e-01
iter      chisq       delta/lim  lambda   a             b

After 4 iterations the fit converged.
final sum of squares of residuals : 27.7666
rel. change during last iteration : -1.50607e-11

degrees of freedom    (FIT_NDF)                        : 9
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 1
variance of residuals (reduced chisquare) = WSSR/ndf   : 3

Final set of parameters            Asymptotic Standard Err
=======================            =======================
a               = 0.961784         +/- 0.2262       (23.52
b               = 0.676919         +/- 1.222        (180.5

correlation matrix of the fit parameters:
                a      b
a               1.000
b              -0.901  1.000
gnuplot> plot 'fit_liniowy.dat', f(x)
```



Składnia komendy fit

fit {[xrange] {[yrange]}} <function> '<datafile>' {datafile-modifiers}

via '<parameter file>' | <var1>{,<var2>,...}

[xrange] [yrange] Zakresy zmiennych, w obrębie których chcemy dopasowywać funkcję. Wszystkie zmienne spoza zakresu zostaną zignorowane.

<function> Wyrażenie definiujące dopasowywaną funkcję lub odwołanie do wcześniej zdefiniowanej funkcji np. f(x) lub f(x,y).

<datafile> Plik z danymi, do których dopasowywana jest funkcja. Określenia służące sprecyzowania sposobu odczytu danych z pliku są takie same jak w poleceniu plot(z wyjątkiem smooth i thru, które nie mają tu zastosowania).

via Określenie precyzujące, które z parametrów funkcji mają być optymalizowane. Jak widać w składni polecenia można je podać albo bezpośrednio oddzielając przecinkami albo poprzez odwołanie do pliku z definicjami parametrów. Jeśli podajemy je bezpośrednio, wówczas wszystkie zmienne, które nie zostały wcześniej zdefiniowane zostają utworzone i przypisana im zostaje wartość 1.0. W przypadku gdy odwołujemy się do pliku z definicjami parametrów powinniśmy zadbać o to by miał on następującą postać:

nazwa_parametru = wartość_początkowa

przy czym definicja każdego z parametrów powinna być podana w osobnej linii.

Uwaga: Parametry podane w nawiasach {} są opcjonalne

Kilka rad dotyczących fitowania

Ponieważ dopasowanie polega na numerycznej minimalizacji wartości funkcji χ**2 (kwadrat odległości między zadanymi punktami a wartością funkcji obliczoną dla tej samej wartości x) dla różnych kombinacji zadeklarowanych parametrów, konieczne jest określenie momentu do którego algorytm będzie próbował zminimalizować tą wartość. Można to zrobić poprzez ustawienie parametru FIT_LIMIT, wówczas jeśli pomiędzy dwoma krokami iteracji algorytm wykryje, że wyliczana wartość zmieniła się mniej niż FIT LIMIT proces dopasowania zostanie uznany za zakończony. Domyślną wartością FIT_LIMIT=1.0e-5

Alternatywną metodą jest ustawienie parametru FIT_MAXITER określającego maksymalną liczbę iteracji. Brak deklaracji lub ustawienie tego parametru na zero oznacza że nie ma limitu wykonanych iteracji.

Zawsze warto porównać wynik dopasowania z danymi. W przypadku niezgodności należy spróbować zmienić wartości początkowe współczynników dopasowania.

Przykład: dopasowanie liniowe

```
gnuplot> f(x)=a*x+b # funkcja do dopasowania liniowego
gnuplot> F(x, a, b)=a*x+b  #funkcja z parametrami a oraz b
gnuplot> fit f(x) 'fit_liniowy.dat' via a, b #fit dla wszytkich punktów
```

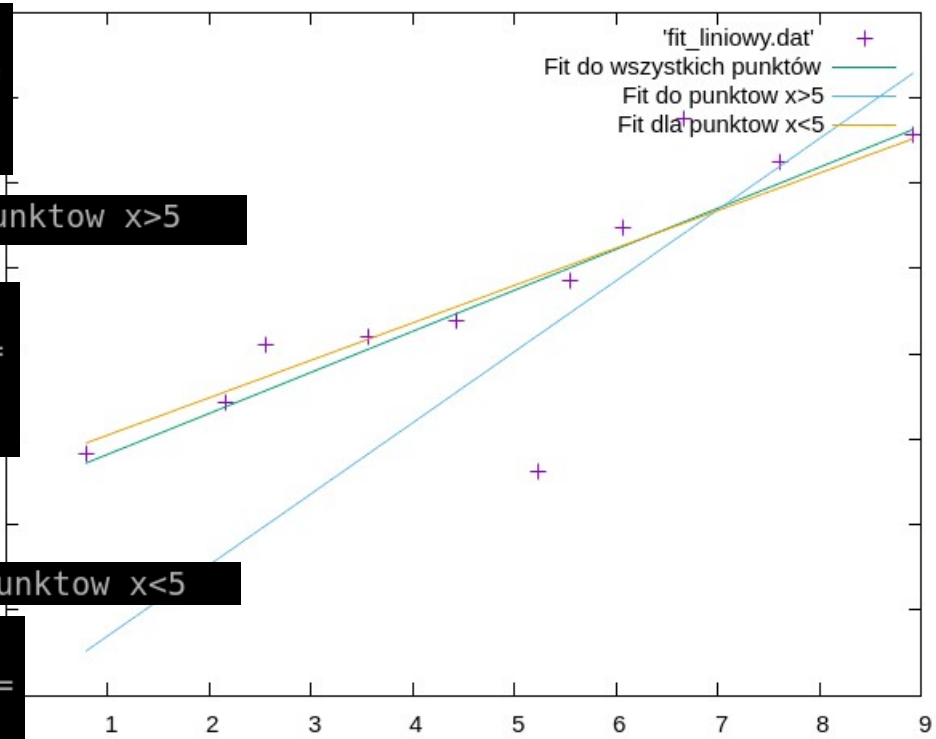
```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 0.961784       +/- 0.2262       (23.52%)
b                 = 0.676919       +/- 1.222        (180.5%)
```

```
gnuplot> fit [5:] f(x) 'fit_liniowy.dat' via a, b #fit dla punktow x>5
```

```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 1.66687        +/- 0.744        (44.64%)
b                 = -4.28416       +/- 5.052        (117.9%)
```

```
gnuplot> fit [:5] f(x) 'fit_liniowy.dat' via a, b #fit dla punktow x<5
```

```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 0.877505       +/- 0.1866       (21.27%)
b                 = 1.21298        +/- 0.555        (45.75%)
```




```
gnuplot> p'fit_liniowy.dat' ps 1.5, F(x, 0.9617, 0.6769) t 'Fit do wszystkich punktów', \
>F(x, 1.666, -4.284) t 'Fit do punktow x>5',  F(x,0.8775, 1.212) t 'Fit dla punktow x<5'
```

Przykład: Dopasowanie wielomianem

```
gnuplot> f(x)=a0+a1*x+a2*x**2+a3*x**3+a4*x**4 #def wielomian stopnia 4
gnuplot> a0=0  #init dla wspolczynnik wielomianu
gnuplot> a1=0
gnuplot> a2=0
gnuplot> a3=0
gnuplot> a4=0
gnuplot> fit f(x) 'cosinus.txt' via a0, a1, a2 # wiel. st 2
Warning: Initial value of parameter 'a0' is zero.
Warning: Initial value of parameter 'a1' is zero.
Warning: Initial value of parameter 'a2' is zero.
  Please provide non-zero initial values for the parameters, at least of
  the right order of magnitude. If the expected value is zero, then use
  the magnitude of the expected error. If all else fails, try 1.0
```

```
Final set of parameters
=======================
a0              = 1.5167
a1              = -1.44884
a2              = 0.23059
```

```
gnuplot> fit f(x) 'cosinus.txt' via a0, a1, a2, a3, a4  # wiel. st 4
Warning: Initial value of parameter 'a0' is zero.
Warning: Initial value of parameter 'a1' is zero.
Warning: Initial value of parameter 'a2' is zero.
Warning: Initial value of parameter 'a3' is zero.
Warning: Initial value of parameter 'a4' is zero.
  Please provide non-zero initial values for the parameters, at least of
  the right order of magnitude. If the expected value is zero, then use
  the magnitude of the expected error. If all else fails, try 1.0
```

```
Final set of parameters
=======================
a0              = 0.938306
a1              = 0.400581
a2              = -1.09551
a3              = 0.328417
a4              = -0.0261346
```

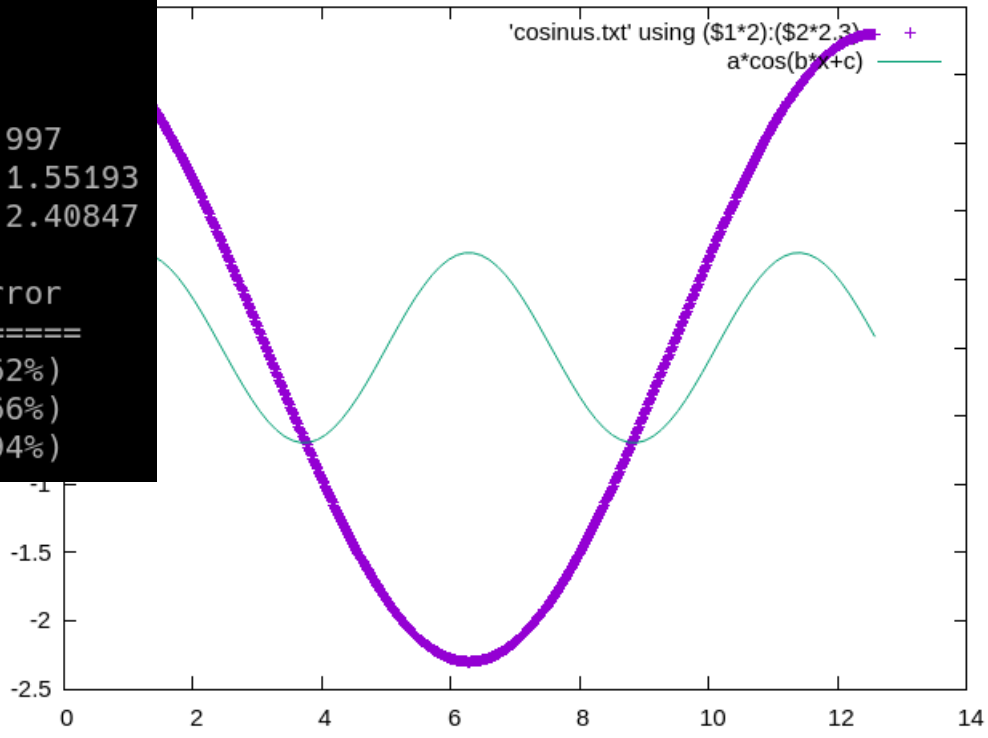
Dopasowanie nieliniowe

```
gnuplot> g(x)=a*sin(b*x+c)  # Funkcja nieliniowa wzg. par.b i c
gnuplot> a=1 # Poczatkowa wart. par.
gnuplot> b=1
gnuplot> c=0
gnuplot> fit g(x) 'cosinus.txt' using ($1*2):($2*2.3) via a, b, c
Warning: Initial value of parameter 'c' is zero.
  Please provide non-zero initial values for the parameters, at least of
  the right order of magnitude. If the expected value is zero, then use
  the magnitude of the expected error. If all else fails, try 1.0
```

```
After 16 iterations the fit converged.
final sum of squares of residuals : 2401.25
rel. change during last iteration : 0

degrees of freedom    (FIT_NDF)                        : 997
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 1.55193
variance of residuals (reduced chisquare) = WSSR/ndf   : 2.40847

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 0.696745       +/- 0.06941      (9.962%)
b                 = -1.23016       +/- 0.02787      (2.266%)
c                 = 3.01644        +/- 0.2019       (6.694%)
```



```
gnuplot> a=1
gnuplot> b=1
gnuplot> c=1
gnuplot> fit g(x) 'cosinus.txt' using ($1*2):($2*2.3) via a, b, c
```

```
After 22 iterations the fit converged.
final sum of squares of residuals : 2.07413e-08
rel. change during last iteration : -8.14577e-08

degrees of freedom    (FIT_NDF)                       : 997
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 4.56111e-06
variance of residuals (reduced chisquare) = WSSR/ndf   : 2.08037e-11

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 2.3            +/- 2.129e-07    (9.258e-06%)
b                 = -0.5           +/- 2.785e-08    (5.57e-06%)
c                 = 7.85398        +/- 1.961e-07    (2.497e-06%)
```

```
gnuplot> p 'cosinus.txt' using ($1*2):($2*2.3), g(x) t 'a*cos(b*x+c)'
```