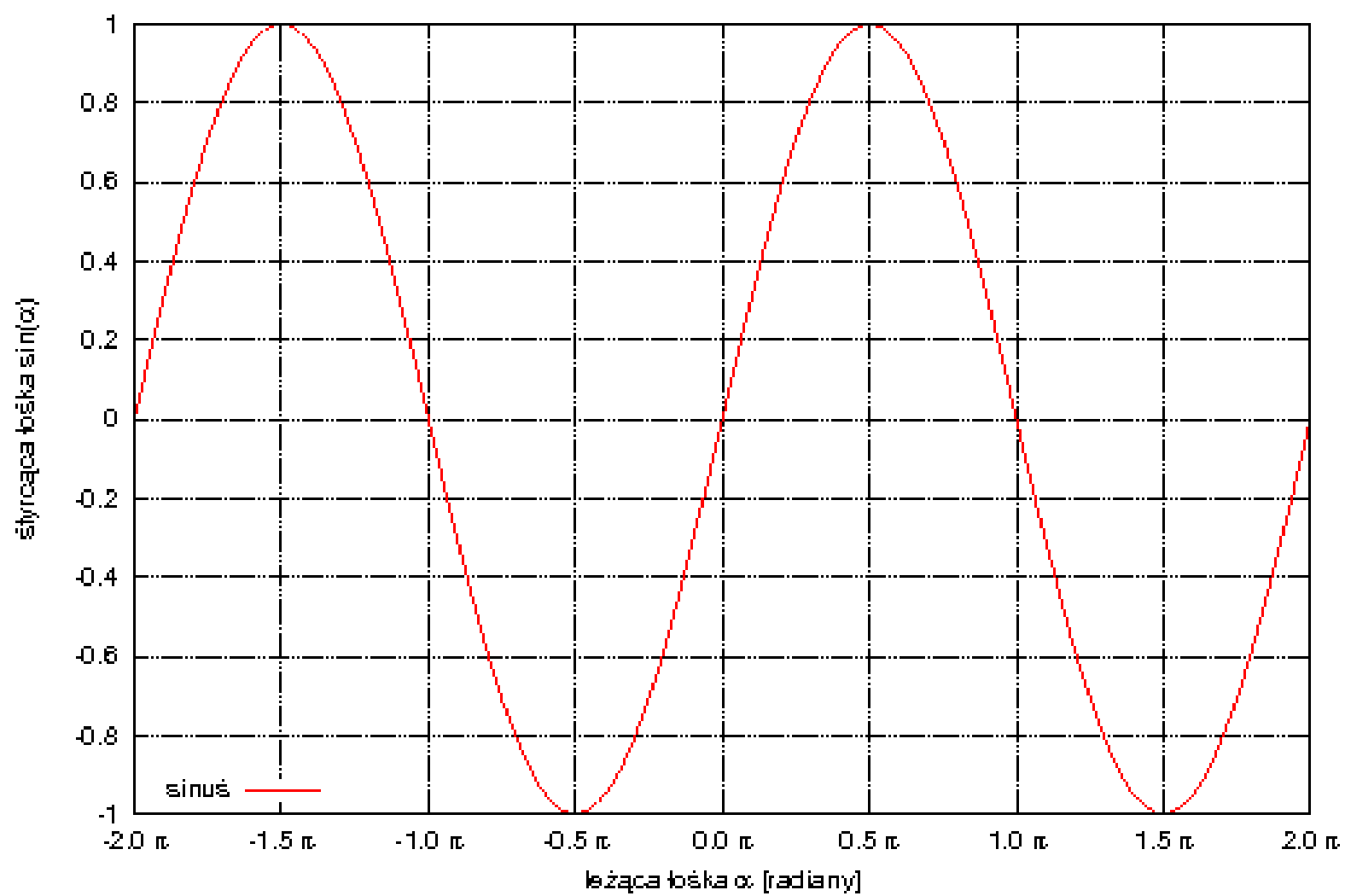


Założenie Głębokie



Adam Cebula "Głupi pomysł, czyli instrukcja rysowania wykresów"

Fahrenheit Crew

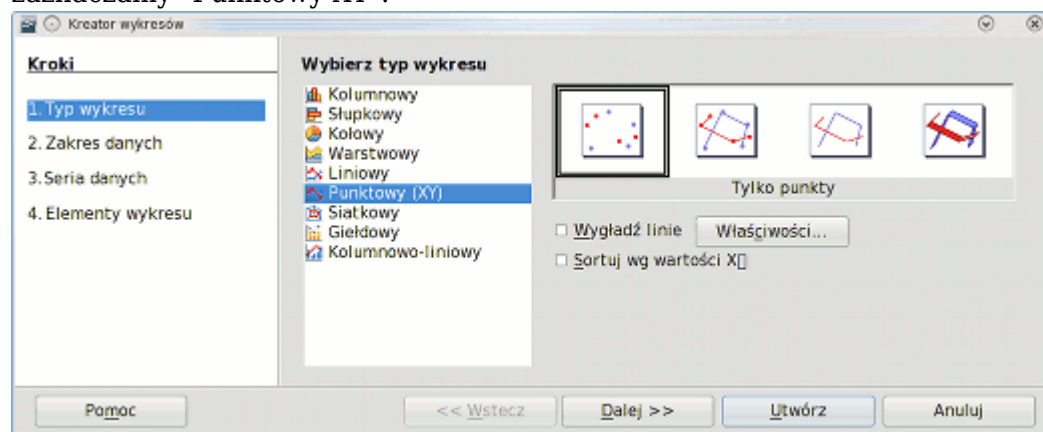
Głupi pomysł, czyli instrukcja rysowania wykresów. Pewnie to nie najmądrzejszy pomysł, by w portalu poświęconym jak najbardziej rozrywce i w którym szuka się ucieczki od codziennych utrapień ukazywał się tekst o tych utrapieniach. Jeszcze bardziej niedobrze, gdy są to utrapienia dla niektórych wyjątkowo nieprzyjemne. Jeszcze chyba mniej rozsądna jest forma pomocy, którą da się określić chyba tylko jako wpuszczenie w maliny.

Ale, rok akademicki za pasem. To raz, a dwa, fakt taki, że robienie wykresów to jedno z największych nieszczęść, jakie studenta spotyka. Diabli wiedzą z czego to wynika, ale dziesięciolecia doświadczeń pokazują, że tak właśnie jest. W obecnym czasie mamy całe mnóstwo znakomitych narzędzi do tworzenia wykresów.

Wystarczy otworzyć arkusz kalkulacyjny OpenOffice'a. W dwie kolumny wpisujemy współrzędne x, y punktów, które mają zostać umieszczone na wykresie. Klikamy komórkę A1. A potem wybieramy z menu Wstaw=>Wykres. Teraz program powinien sam zaznaczyć na arkuszu kalkulacyjnym zakres danych do wykresu.

Chciałem Ci zwrócić uwagę tylko na jeden drobny szczegół. Mianowicie, aby powstał WYKRES, a nie "grafika biznesowa" w okienku, które ci się pojawi, możesz zaznaczyć tylko jedną opcję: „Punktowy XY”. Wszystkie inne służą do tworzenia grafiki. Czym różni się "wykres" od "obrazka"? Tym, że z wykresu możemy odczytać wartości. Wykres jest do tego, by przystawić do niego ekierkę i znaleźć odpowiadające sobie na wymalowanej linii oraz osiach punkty, którym da się z rozsądną dokładnością przypisać wartości. Dlatego np. zaawansowane funkcje programów do opracowania wyników w postaci "wykresów 3D" nie za bardzo służą do tworzenia wykresów.

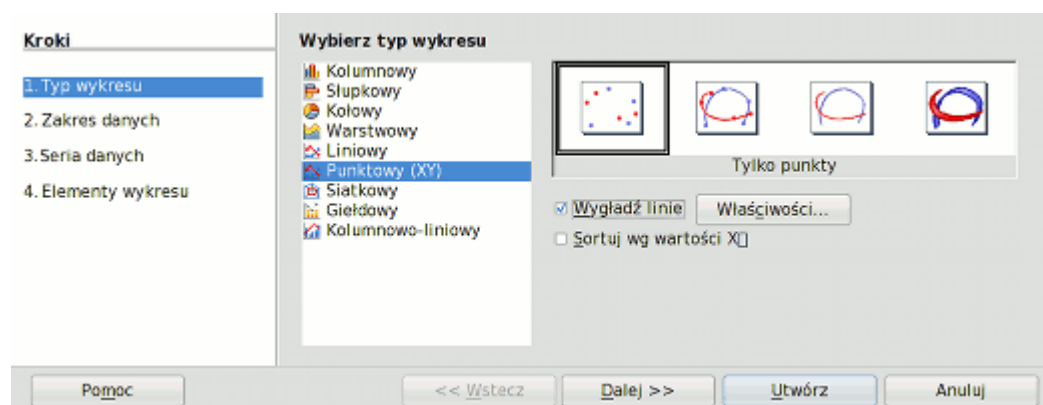
A więc wybieramy wykres "Punktowy XY". Otwiera się więc okno "Wybierz typ wykresu" i zaznaczamy "Punktowy XY".



Okno wyboru typu wykresu w OpenOffice

Taki szczegół, inne typy wytną nam psikusy i potraktują oś x jako "etykietę". Nie jako współrzędną odległości od początku układu współrzędnych, ale źródło tekstu do wstawienia w miejscu, w którym znajduje się kolejny punkt.

Bardzo pomocną funkcją jest okienko wyboru "Wyładź linie" widoczne pod okienkiem wyboru wyglądu. Jeśli nasz wykres biegnie jakoś krzywo, ten algorytm zastępuje to, co można jeszcze w starszych podręcznikach wyczytać, czyli rysowanie krzywej z pomocą krzywików.



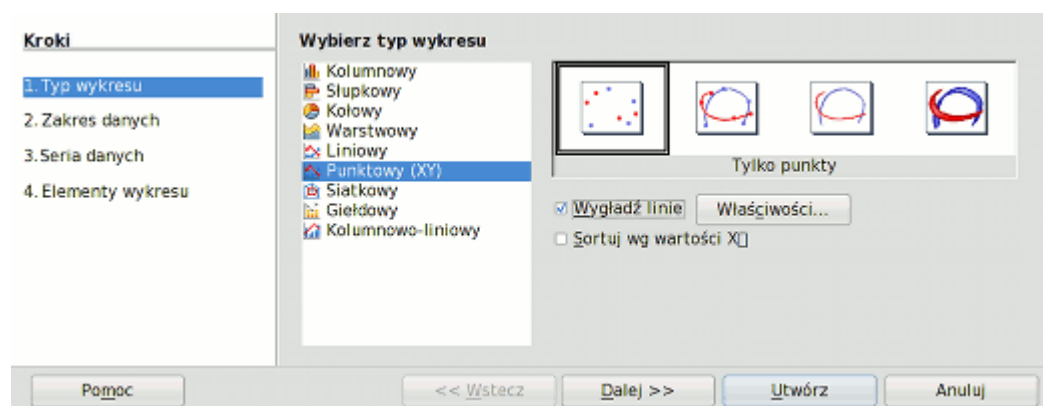
Efekt zaznaczenia okienka edycyjnego "wyładź linie"

Funkcji tej używamy, gdy nie znamy formuły matematycznej opisującej zmianę naszych danych. Jeśli mierzyliśmy zależność położenia ciała od czasu w ruchu jednostajnie przyspieszonym wiemy, że to ma być parabola i powinniśmy przybliżyć wykres parabolą, a jeśli wyniki nie pasują do niej, mamy okazję do medytacji, co się mianowicie stało.

Ale to wiedza na potem. Na teraz wystarczy tyle, że "Wyładź linie" wyrysuje nam krzywe, a jeśli klikniemy przycisk "Właściwości", to będziemy mogli zmienić stopień wyładzenia linii.

Po wybraniu rodzaju graficznego wyglądu wykresu i stopnia wyładzania, klikamy dalej i otwiera nam się okno wyboru zakresu danych. W naszym wypadku, gdy wystartowaliśmy od dwu kolumn, nic tu nie musimy zmieniać, o ile np. nie kliknęło się nam gdzieś w jakąś odległą od danych komórkę i program "zgłupiał", nie wybrał żadnych danych.

Okno to będzie niezbędne w sytuacji, gdy mamy wiele kolumn z danymi i trzeba ręcznie wybrać, z których mamy wykonać wykres. Wreszcie, jest to bardzo częsty przypadek, trzeba wybrać tylko fragment zapisanych danych do wykresu, bo reszta jest nieinteresująca.



Efekt zaznaczenia okienka edycyjnego "wyładź linie"

Krok 3 to wybór danych.

Kroki

1. Typ wykresu
2. Zakres danych
3. Seria danych
4. Elementy wykresu

Wybierz zakres danych

Zakres danych
\$Arkusz1.\$A\$1:\$C\$40

☐ Seria danych w wierszach
☒ Seria danych w kolumnach
☐ Pierwszy wiersz jako etykieta
☐ Pierwsza kolumna jako etykieta

Pomoc << Wstecz Dalej >> Utwórz Anuluj

Wybór zakresu danych

W kroku czwartym możemy opisać osie i nadać tytuł wykresowi.

Kroki

1. Typ wykresu
2. Zakres danych
3. Seria danych
4. Elementy wykresu

Wybierz tytuły, legendę i ustawienia siatki

Tytuł

Podtytuł

Oś X

Oś Y

Oś Z

Wyświetl legendę

☒ Wyświetl legendę
☐ Lewo
☒ Prawo
☐ Góra
☐ Dół

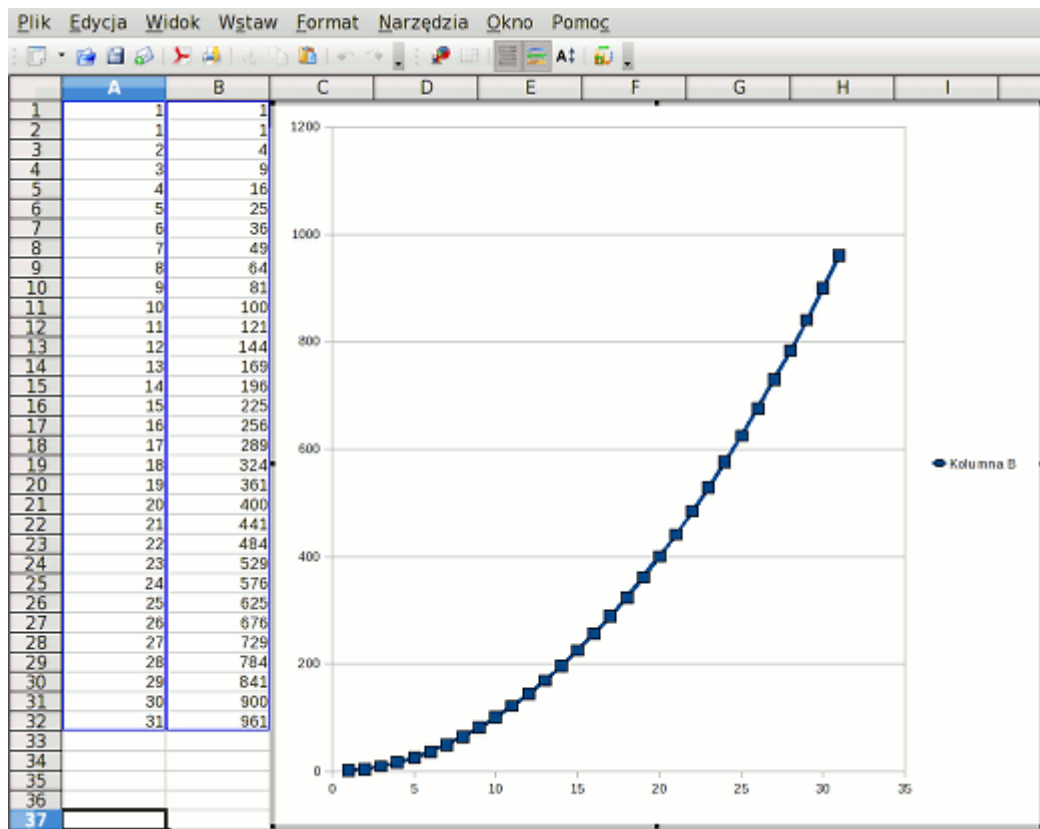
Wyświetl siatki

☐ Oś X ☒ Oś Y ☐ Oś Z

Pomoc << Wstecz Dalej >> Utwórz Anuluj

Koniec tworzenia wykresu. To możemy opisać osie i nadać tytuł.

W Open Office w czterech krokach tworzymy śliczny wykres, na dodatek całkowicie edytowalny.



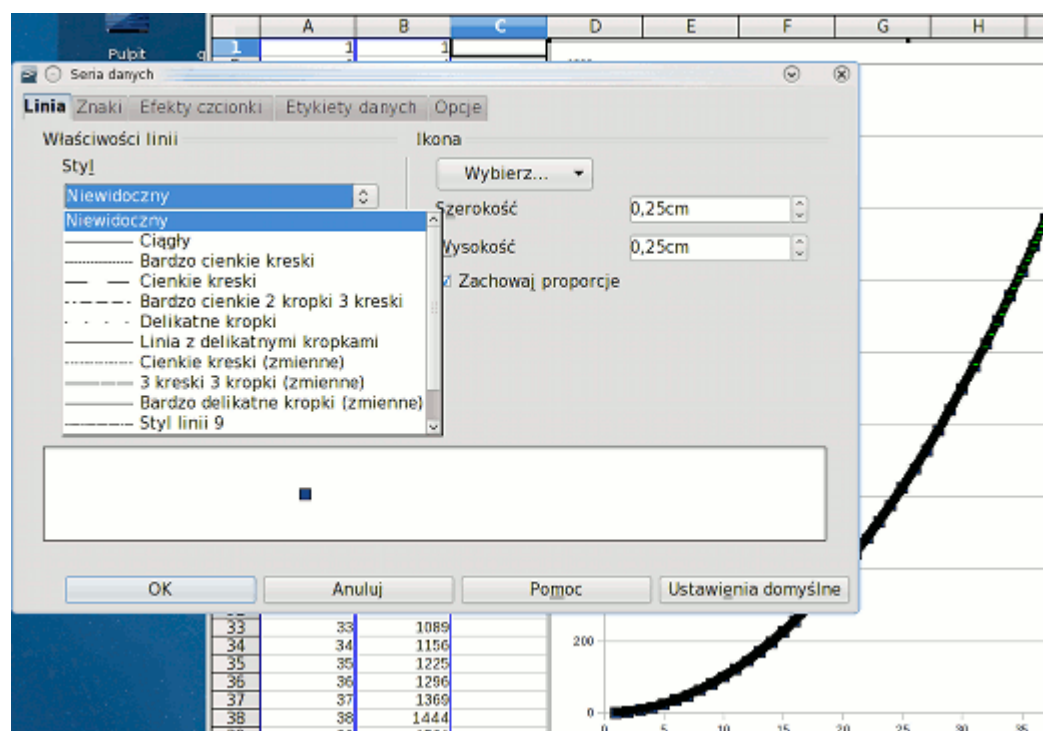
Wykres w arkuszu kalkulacyjnym OpenOffice. Widoczne otoczone niebieską linią dwie kolumny danych, które program automatycznie wybrał do rysowania wykresu

Po umieszczeniu go na arkuszu możemy ponownie uaktywnić jego okno. Jeśli nie podoba nam się tytuł wykresu albo nie wstawiliśmy opisu, wówczas (przy aktywnym oknie wykresu) możemy wybrać z menu arkusza Wstaw=>Tytuł i zobaczymy okno, w którym to wszystko możemy zmienić lub uzupełnić.

The screenshot shows the 'Wykres funkcji kwadratowej' (Quadratic Function Graph) dialog box. It contains fields for the title, subtitle, and axes. The 'Tytuł' (Title) field is filled with 'Wykres funkcji kwadratowej'. The 'Podtytuł' (Subtitle) field is empty. The 'Osie' (Axes) section has 'Oś X' (X-axis) set to 'moje dane' and 'Oś Y' (Y-axis) set to 'wyniki obliczeń'. There are also empty fields for 'Oś Z' (Z-axis) and 'Osie dodatkowe' (Additional axes). The 'OK', 'Anuluj' (Cancel), and 'Pomoc' (Help) buttons are on the right.

Wykres w arkuszu kalkulacyjnym OpenOffice.Okno opisu osi.

Linia wykresu także jest aktywna, po jej kliknięciu mamy okno, w którym możemy np. zmienić rodzaj linii łączącej punkty albo ją usunąć (opcja "niewidoczna").



Wykres w arkuszu kalkulacyjnym OpenOffice. Edycja stylu linii.

Ten kawałek poradnika możemy podsumować tak: pracownicy uczelni zazwyczaj wojują ze studentami o te same problemy z wykresami. Bodaj pierwszą sprawą jest opis osi, opis samego wykresu. Trzeba mu nadać sensowny tytuł: "zależność tego od tamtego". Na osiach oprócz wartości muszą się znajdować jednostki. Nawet jeśli są one opisane w podpisie. Trzecią sprawą jest skala. Wykres musi dotyczyć interesującego zakresu wyników, a nie np. wszystkich wyników. Dobrze, gdy na wykresie mamy ułatwiającą odczyt siatkę. Wszystko to załatwiamy, jak widać, za pomocą arkusza kalkulacyjnego OpenOffice'a.

Na dobiecie pewnego wniosku chciałbym dodać: nie ma sensu opanowywanie kilku programów do rysowania wykresów. Zysk jest bardzo niewielki. Dobrze jest znać przynajmniej jeden program, za to solidnie.

To wszystko razem wzięte przemawia silnie przeciw temu, by zajmować się czymś niejako z przeciwnego krańca informatycznej galaktyki, czyli gnuplotem.

Od razu muszę zaznaczyć, że nie mam najmniejszej ambicji pisać "prawdziwego wprowadzenia" bo jak w tym starym kawale o niemieckich dziełach owego wprowadzenia pewnie trzeba by machnąć ze 20 opasłych tomów. Tak więc, będzie to osobliwy rodzaj eseju w formie suchej instrukcji.

O programie pisałem już kilka razy, ale trzeba wspomnieć coś o jego rodowodzie, by opracować sobie strategię "jak się za to zabrać". Program ma filozofię działania sprzed czasów komputerowych kart graficznych. Jest to także "modułowa filozofia". Przerabia on dane ze wskazanego pliku i wrzuca

do modułu, który robi z nich plik w formacie, jaki wybrał użytkownik. Nie wyświetla się w oknie za każdym razem. Okno graficzne wyświetlane na ekranie jest domyślnym strumieniem, do którego są kierowane dane.

Gdy chcemy wynik pracy gdzieś zapisać musimy dość pracowicie wykonać operację utworzenia pliku wyjściowego, a następnie ustawienia formatu, w jakim dane graficzne zostaną zapisane.

Drobiazg: w oryginale program pracuje z wiersza poleceń. Owszem, mamy różne rodzaje GUI do niego, ale – powiedzmy szczerze – dość kulawe. Pełne możliwości daje tylko wklepywanie komend. Taka (bolesna) prawda.

Tu może warto wtrącić dygresję "czy to w ogóle ma sens?" No właśnie: po co – skoro już pokazaliśmy jak w 4 krokach za pomocą OpenOffice'a wykonać wykres i jeszcze dodaliśmy, że nie ma sensu za bardzo wyuczanie się kilku programów do tego samego – kręcić sobie głowę gnuplotem?

Odpowiedź nie jest całkiem prosta: owszem, można przytoczyć kilka niewątpliwych zalet programu: np. dostępność. O ile mamy sieć, w chwili obecnej zainstalujemy go prawie na każdym komputerze. Nie spotkałem się z dystrybucją Linuksa, na którą gnuplot byłby niedostępny. Pracuje na Windows, na MacOS, na dziwadłach (dla przeciętnego użytkownika) typu FreeBSD. "Paczka" do pobrania ma zwykle do 4 MB, więc przepcha się przez każde łącza. Gnuplot został napisany specjalnie do wykonywania ciężkich prac obliczeniowych i "zrobi wszystko". Jego możliwości są porównywalne z drogimi programami, a na dodatek jest wtyczką do takich obliczeniowych kombajnów, jak Maxima czy GNU Octave.

Komuś, kto się "z tą cholerą" zetknie po raz pierwszy, może się wydać trochę bulwersujące stwierdzenie, że gnuplot został faktycznie napisany tak, by użytkownikowi ułatwiać życie. A jednak z czasem przekonamy się, że to pewnie prawda. Temu służy wielość tzw. terminali. W rzeczywistości są to moduły programistyczne (sterowniki?), które produkują wyjściowe dane w "różniastych formatach".

Pewnie w 95% przypadków nie będziemy mieli problemów z informatyczną zgodnością, zwłaszcza gdy dane kierujemy po prostu do drukarki. Jednak nawet dla tych pozostałych nielicznych przypadków warto wiedzieć, że mamy furtkę w postaci właśnie gnuplota.

Obsługuje porządnie postscript, produkuje miniaturowe w rozmiarach pliki png (przy rozmiarze 600x480 mniej niż 10 kB), które są czytelne i nadają się znakomicie na strony www. Autor natknął się na problemy z formatem svg – brakowało zamykającego plik znacznika (trzeba plik wynikowy otworzyć w edytorze tekstowym i dopisać ""), lecz poza tym program produkuje takie wektorowe dziwadła, jak pliki programu Xfig czy tgif.

Gdyby ktoś się pytał po co to (czyli Xfig i tgif) komu: są to dobrze działające programy do produkowania różnej maści rysunków do prac. Źle się nadają do ozdabiania, bardzo dobrze do wykonywania schematów, diagramów, rysunków technicznych. Można dzięki temu np. uzupełnić wykres o rysunek, który dodatkowo objaśni pomiar.

Zysk z możliwości zapisu na wiele różnych sposobów staje się w pełni jasny wówczas, gdy zapoznamy się z właściwościami różnych formatów danych. Np. pliki graficzne zapisywane jako jpg znakomicie nadają się do kompresji kolorowych zdjęć. Fatalnie działają na czarno-białych rysunkach. W skrajnym wypadku plik jpg może okazać się większy od pliku np. bmp, w którym zapisano rysunek. Pliki formatów wektorowych, takich jak postscript, są małe i doskonale nadają się do skalowania, ale wymagają do odczytu "specjalistycznych" programów. Pliki eps dobrze "wchodzą" do LaTeX-a, a w Linuksie możemy je skonwertować poleceniem `ps2pdf nazwa_pliku.ps nazwa_pliku.pdf` do

popularnego formatu pdf.

Niestety, o zaletach programu gnuplot zazwyczaj przekonujemy się "po czasie". Jego chyba najważniejszy plus polega na tym, że jest on "technicznie intuicyjny". Mówiąc inaczej, działa tak, jak się może tego spodziewać informatyk.

Podam tu przykład. W tzw procesorach tekstu, których typowym przedstawicielem ma być Word istnieją wesołe funkcje typu autoformatowanie. Potrafią doprowadzić do rozpaczki piszącego, na przykład zamieniając mu słowa. Nic nie pomoże dogłębna znajomość komputerów i języków programowania, gdy chcemy to wyłączyć. Programista zakopał wedle własnej fantazji gdzieś głęboko wielki czerwony przycisk STOP od tej maszyny i możemy spędzić wiele upojnych godzin na jego poszukiwaniach.

Problemy, na jakie się nadziejemy pracując z gnuplotem, są informatycznie zrozumiałe, a nie wynikają z fantazji piszących tak zwane interface użytkownika. To, co dla początkującego staje się nieszczęściem, dla "geeka" jest właśnie zaletą.

Jedną z takich cech jest prostota używanych formatów. Najlepiej, jeśli pliki z danymi są faktycznie plikami txt. Jeśli ulegną "uszkodzeniu", możemy je bardzo łatwo przejrzeć przy pomocy dowolnego edytora tekstu i "naprawić". Nieszczęście, jakim jest konieczność wklepywania komend, zamienia się w wielką wygodę dzięki temu, że możemy pisać bardzo proste skrypty i przetwarzać dane niemal hurtowo.

Nieszczęście, że program jest stary i kolejne wersje niewiele się różnią od poprzednich gwarantuje nam, że do wyników roboty da się wrócić po latach.

Zakończmy ową dygresję i wróćmy do sedna sprawy. Jak się "tym" w ogóle posłużyć? Powiedzmy, że potrzebujemy do pracy wykresu funkcji sinus. Jako obrazka. Co robimy? Po pierwsze, musimy mieć ów program.

To już osobna para kaloszy, najlepiej zainstalować wersję przystosowaną do danej dystrybucji. Nie masz Linuksa, to... No dobrze, udajemy się na stronę <http://www.gnuplot.info/> A jak już mamy gnuplota w Linuksie, otwieramy okno terminala i w wierszu poleceń wpisujemy po prostu "gnuplot". Zobaczmy coś takiego:

```
bash-3.2$ gnuplot
```

```
G N U P L O T Version 4.2 patchlevel 3
```

```
last modified Mar 2008 System: Linux 2.6.27.41-170.2.117.fc10.x86_64
```

```
Copyright (C) 1986 - 1993, 1998, 2004, 2007, 2008
```

```
Thomas Williams, Colin Kelley and many others
```

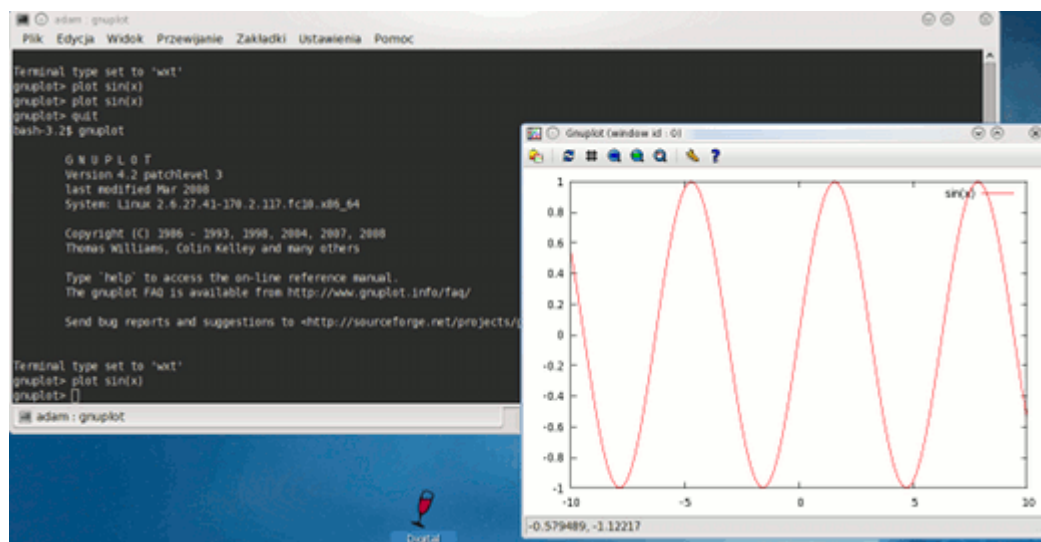
```
Type `help` to access the on-line reference manual. The gnuplot FAQ is available from  
http://www.gnuplot.info/faq/
```

```
Send bug reports and suggestions to <http://sourceforge.net/projects/gnuplot>
```

```
Terminal type set to 'wxt' gnuplot>
```

Teraz w wierszu poleceń programu wpisujemy: `plot sin(x)` Pojawi się nam okno graficzne z naszą

funkcją.

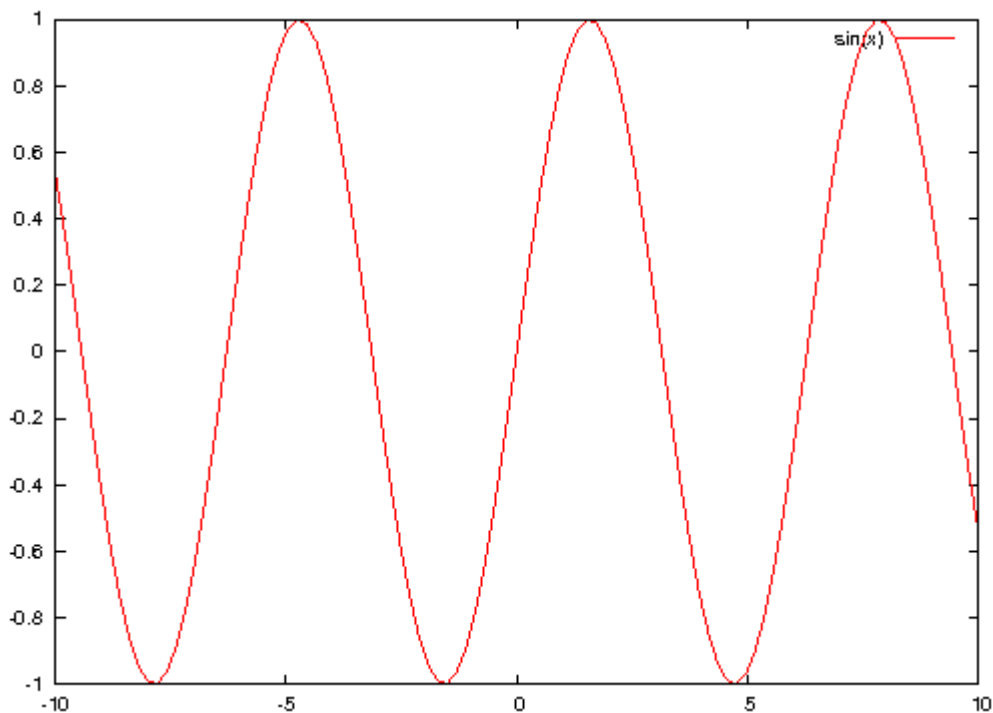


Okno terminala tekstowego oraz okno graficzne programu gnuplot.

Słowo się rzekło, nie ma tak pięknie, że to już koniec. Możemy oczywiście na siłę wykonać zrzut ekranu, ale to bez sensu. Trzeba zdecydować się na typ pliku, do jakiego nasze dzieło zostanie zapisane. Ustawiamy go na, powiedzmy, eps.

Dlaczego eps? Jeśli piszemy pracę, do której chcemy wstawić wykres w OpenOffice, to warto utworzyć rysunek w formacie, który ten pakiet czyta. Akurat pliki eps "wchodzą". Lepiej (jeśli to możliwe) w takich dokumentach unikać obrazków bitmapowych, bo się źle skalują. Potrafią przy tym wyciąć niespodziewane numery np. ze znikaniem linii.

Wyboru dokonujemy komendą: set terminal postscript eps color. Następnie tworzymy plik docelowy komendą: set output'sinus.eps'. Obrazek zostanie po ponownym wykonaniu komendy "plot sin(x)", przy czym w Linuksie poruszamy się za pomocą strzałek po historii poleceń i już nie musimy niczego wpisywać.



Sinus z gnuplota na szybko.

Czego brak naszemu wykresowi? Po pierwsze, skala x jest bez związku z funkcją, która jest wyrysowana. Argumentem sinusa są tu radiany. Musimy zmienić i zakres osi, i jej skalę. Podziałkę zmieniamy poleceniem: `set xtics pi/2`. Krokiem przy jej rysowaniu będzie liczba pi podzielona przez 2, a nie kolejne liczby naturalne. Nasz program jest dość mądry, żeby rozumieć, że chodzi nam o "to pi".

Zakres wyświetlonego wykresu ustawiamy poleceniem: `set xrange [-2*pi:2*pi]`. Dodamy siatkę poleceniem: `set grid`. Dodamy opis osi x: `set xlabel'radiany'` oraz osi y: `set ylabel'sinus'`, dodajemy tytuł wykresu: `set title'wykres funkcji sinus'`.

Nie za bardzo? No to poprawimy. Zrobimy tak, by na wykresie pojawiły się greckie litery: `set terminal postscript eps color enhanced`. Dodaliśmy do polecenia utworzenia terminala "enhanced". Teraz utworzymy podpis osi: `set xlabel'{\Symbol a} [radiany]'`. Podobnie zmienimy opis osi pionowej: `set ylabel'sin({\Symbol a})'`.

Znawcy komputerów już wietrzą problem z polskimi znakami. Jest pewien problem. Wynika on z tego, że gnuplot został dostosowany do obsługi kodowania iso 8859-2. W chwili obecnej systemy operacyjne (dystrybucje Linuksa) przeszły na UTF-8. Dlatego napis, który został wprowadzony z polskimi znakami, nie może być prawidłowo wyświetlony. Takie kodowanie mamy w terminalach tekstowych.

Warto zwrócić uwagę na to, że gdyby słowa kluczowe używane w komendach zawierały też jakiegokolwiek znaki spoza ASCII, program też by ich nie zrozumiał. Lecz narodowe znaki mogą się znaleźć tylko w opisach. Kodowanie UTF-8 i znaki ASCII pokrywają się, dlatego mamy taki efekt, że mimo zmiany kodowania program działa, ale nie działają narodowe czcionki.

Poradziłem sobie z tym w następujący sposób. Gnuplot przetwarza skrypty czyli proste pliki tekstowe. Piszemy taki plik wsadowy, który jest po prostu listą poleceń, takich samych, jakie wpisalibyśmy bezpośrednio z terminala. Po czym (użyłem programu gedit) zapisujemy nasze dzieło z

kodowaniem iso 8858 - 2.

Poniżej mamy coś takiego, przy czym dokonałem w tym skrypcie jeszcze kilku innych wyczynów. Pierwsza linijka naszego skryptu została "zakomentowana" znaczkiem "#". Informuje on program, że to jest opis dla użytkownika i ma tę linijkę pominąć. Kolejna linijka ustawia kodowanie na środkowoeuropejskie. Warto zwrócić uwagę na zapis "iso_8859_2" który różni się od zwykle używanego opisu tego kodowania (m.in. używanego w stronach html).

Linijka "set key left bottom" wymusza przeniesienie opisu linii wykresu z prawego górnego rogu wykresu w lewy dolny. Linijka set format x "%3.1P {/Symbol p}" zmienia opis punktów podziałki osi x. "Tak mniej więcej", by nie wdawać się w opis składni, znak % odpowiada za wstawianie kolejnych wartości wygenerowanej podziałki, a następujące po nim litery coś z tą wartością robią. %P wymusza podawanie jej wartości w wielokrotnościach liczby pi. Liczba 3.1 to informacja, że w opisie mają się znaleźć maksymalnie 3 cyfry, w tym jedna po przecinku, bo mamy 1 po kropce. W nawiasie klamrowym mamy wymuszenie wydrukowania litery greckiej pi. Poniżej tekst skryptu.

```
#to jest plik wsadowy gnuplota
```

```
set encoding iso_8859_2
```

```
set terminal postscript eps color enhanced
```

```
set output'polski1.eps'
```

```
set title"Zażółć Gęślą Jaźń"
```

```
set key left bottom
```

```
set xtics pi/2
```

```
set format x "%3.1P {/Symbol p}"
```

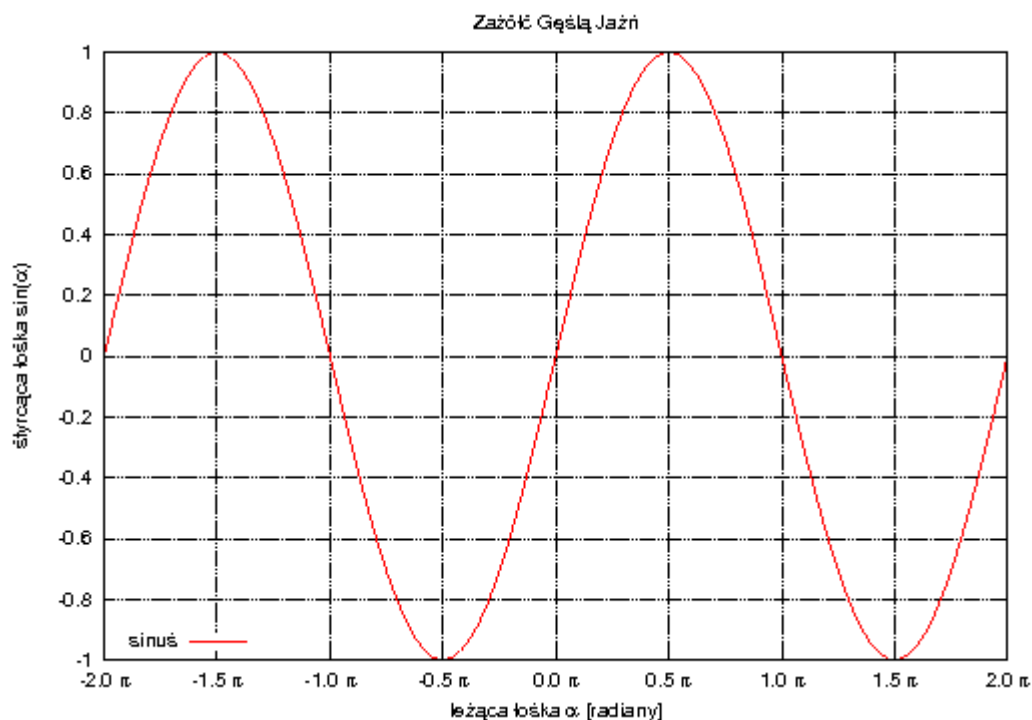
```
set xrange [-2*pi:2*pi]
```

```
set grid
```

```
set xlabel"leżąca łożka {/Symbol a} [radiany]"
```

```
set ylabel"śtyrcząca łożka sin({/Symbol a})"
```

```
plot sin(x) title"sinus"
```



Wynik działania skryptu.

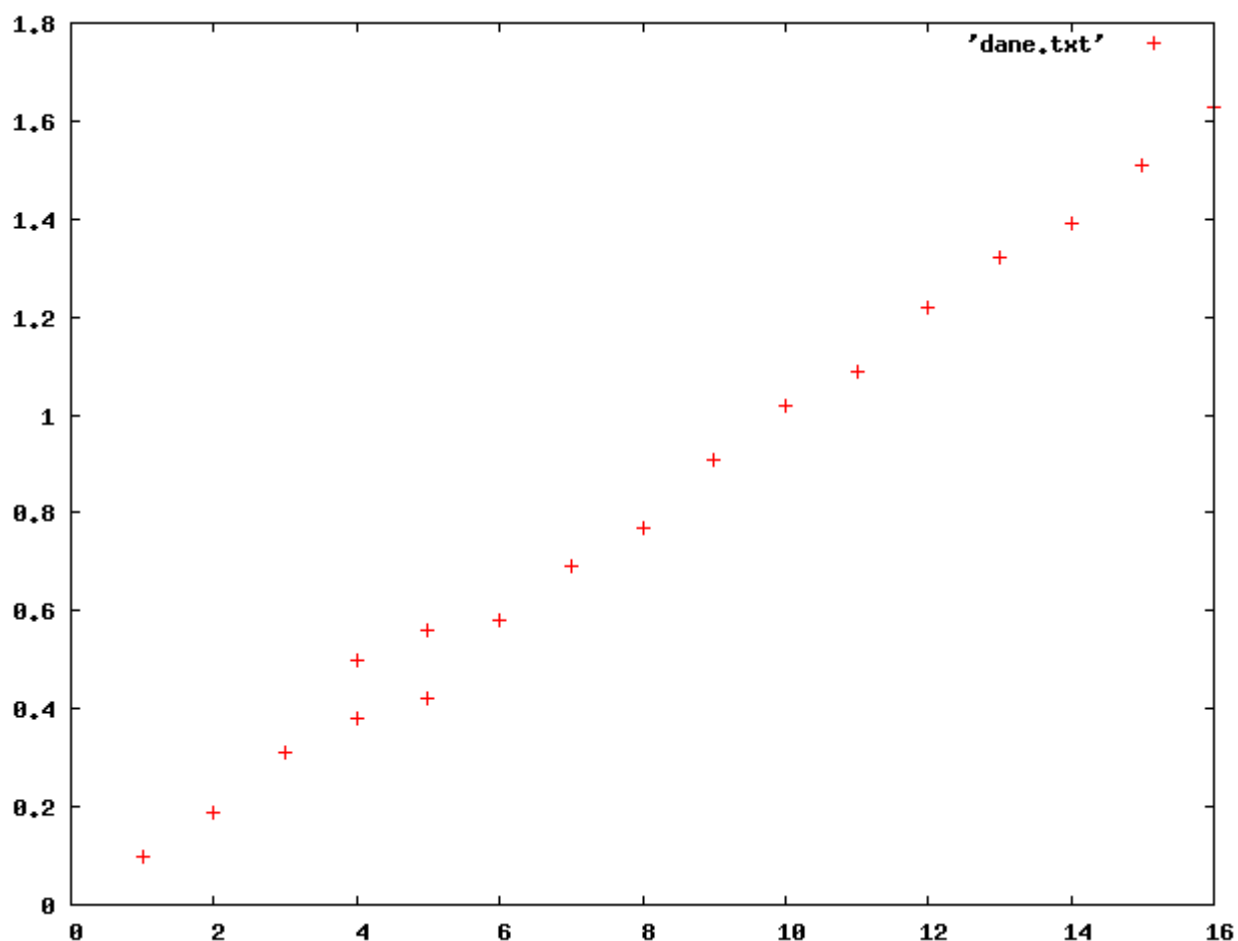
Chyba najważniejszą nauką płynącą z powyższego przykładu jest jak szukać pomocy do komend. Niestety, trzeba szczerze powiedzieć: koncepcja helpa dla gnuplota jest także archaiczna. Pomoc jest skuteczna w przypadku osoby bardzo pracowitej, która przeryła wielką dokumentację i w trakcie roboty potrzebuje tylko przypomnienia, jak wygląda np. składnia polecenia. Dlatego tworząc własne skrypty szukamy ratunku pisząc np. "help set", help set format, help tics itd. Da się coś zrobić w rozsądnym czasie, jeśli znamy kluczowe słowa, o jakie chcemy zapytać.

Skoro tak dobrze się udało, zrobimy coś innego: wykonamy wykres danych eksperymentalnych. Najprościej to zrobić, gdy są one zapisane w pliku tekstowym w postaci dwu kolumn rozdzielonych zwyczajnie spacją. Np. może to wyglądać tak:

```
1 0.1
2 0.19
3 0.31
4 0.38
5 0.42
4 0.50
5 0.56
6 0.68
7 0.69
```

8 0.77
9 0.91
10 1.02
11 1.09
12 1.22
13 1.32
14 1.39
15 1.51
16 1.63

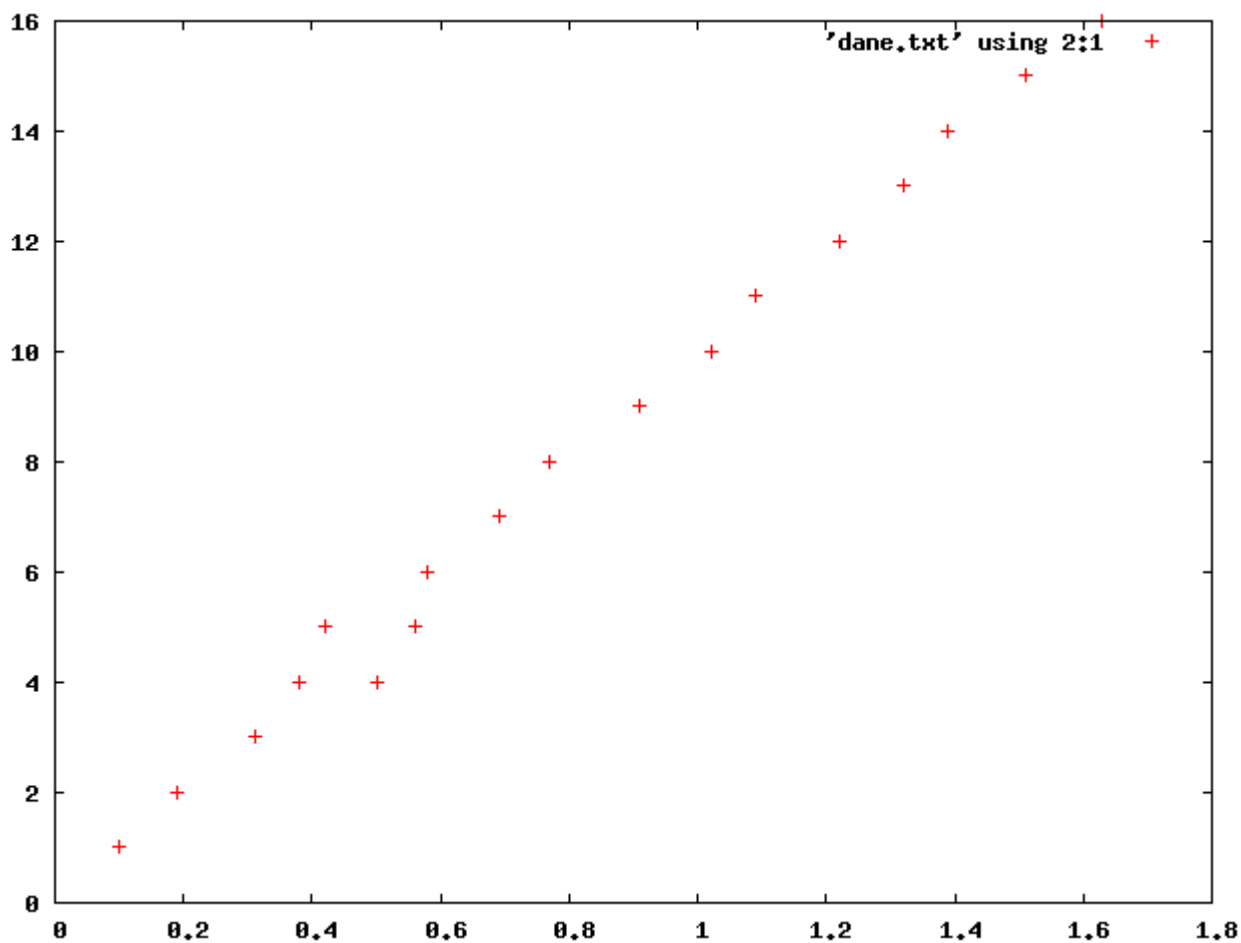
Dane zapiszemy w pliku "dane.txt". Sensowne przedstawienie wyników zależy od okoliczności. Najprościej jest uzyskać wykres złożony z samych punktów. Piszemy po prostu w wierszu poleceń programu: `plot'dane.txt'`.



Najprostszy wykres danych z pliku w gnuplocie.

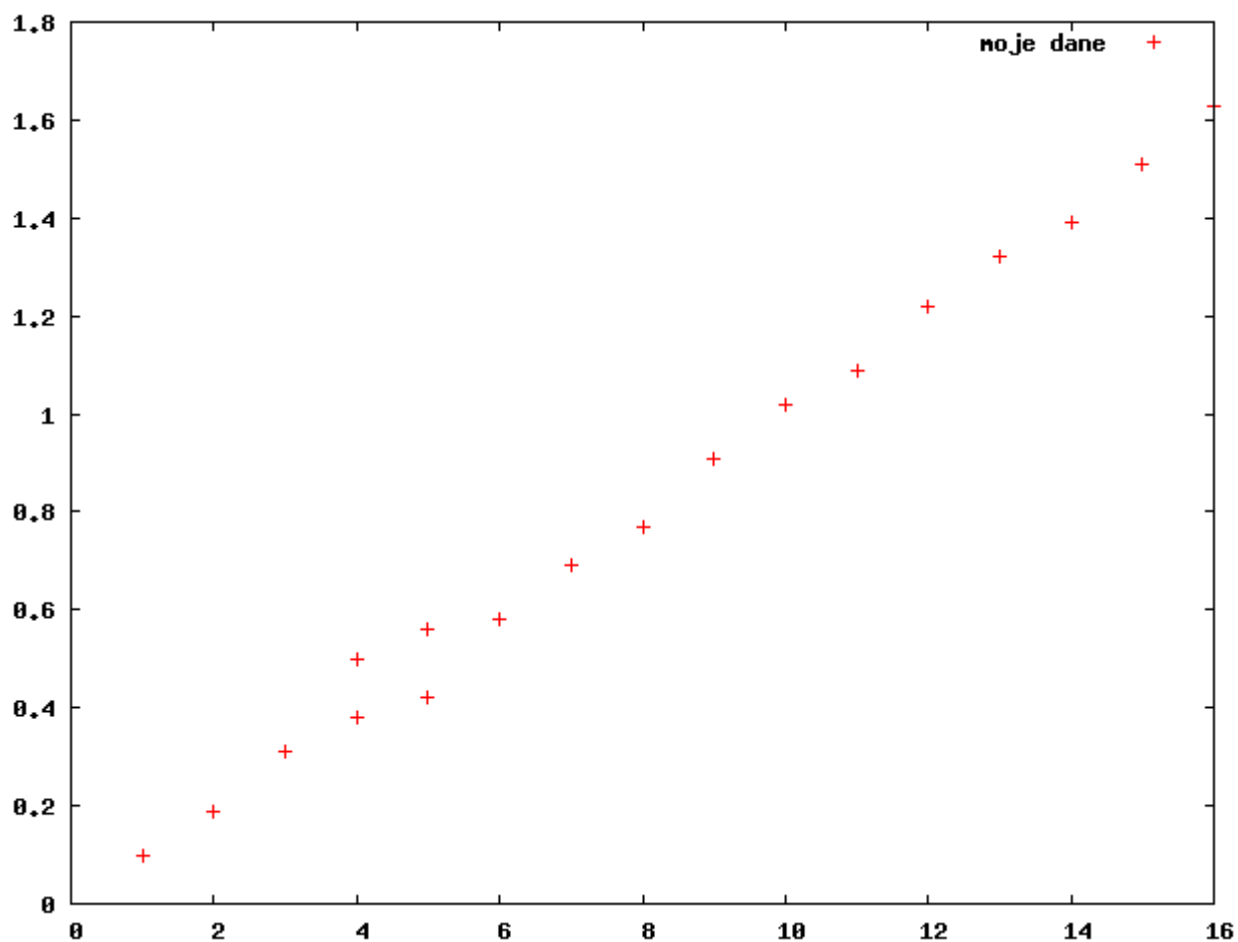
Uwaga, to zadziała, o ile dane będą porządnie zapisane w dwu kolumnach oddzielonych spacją. Jeśli

w pliku będą trzy kolumny, a zechcemy wykonać wykres zależności kolumny 2 od 3, to nie wyjdzie. Program domyślnie traktuje kolumnę pierwszą jako współrzędną x , a drugą jak y . Jeśli chcielibyśmy to zmienić, musimy poinformować o tym program za pomocą polecenia `using`. Polecenie wykonania "odwrotnego" wykresu będzie wyglądało: `plot'dane.txt' using 2:1`.



wykres, w którym kolumna 2 była danymi x , zaś kolumna 1 - y .

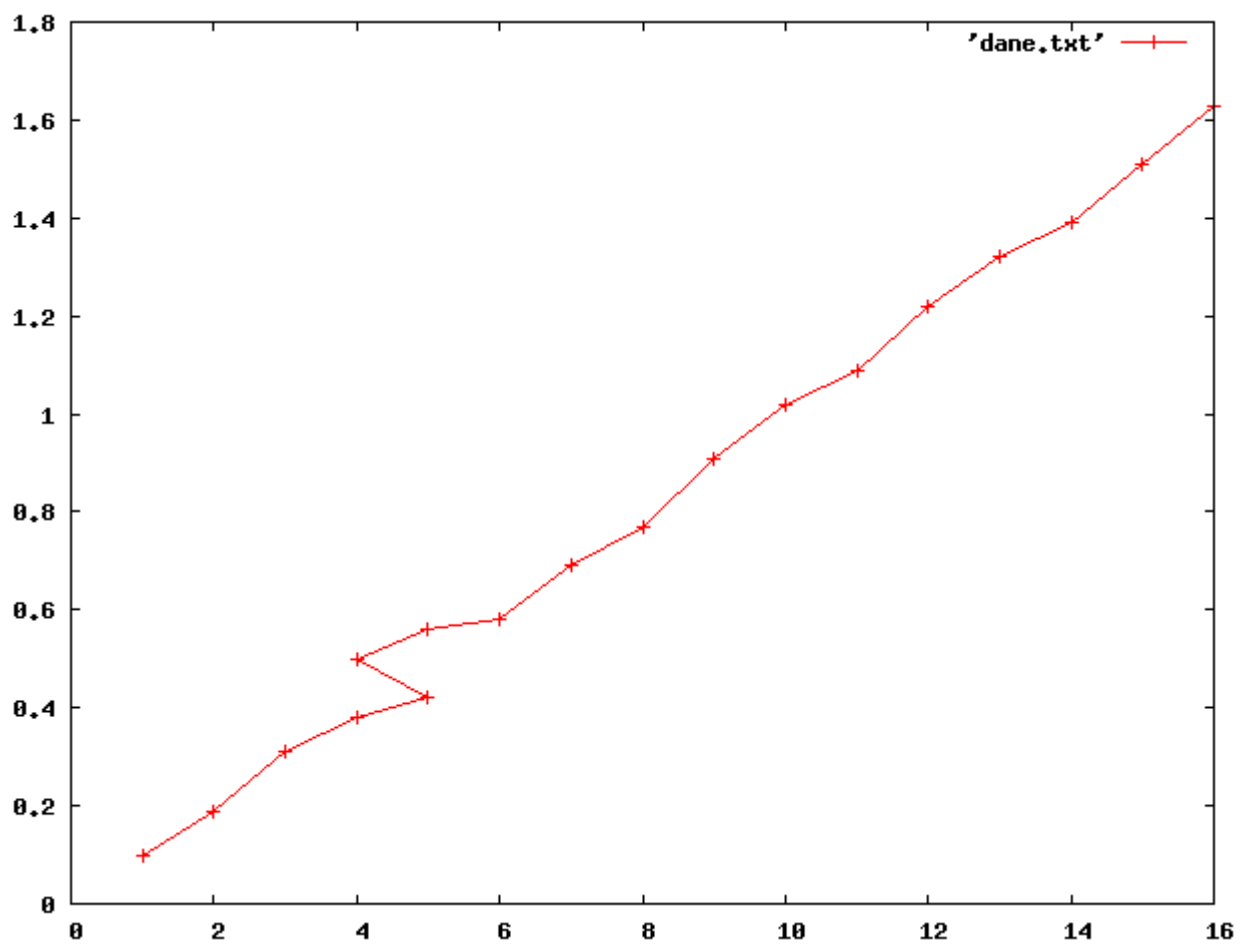
Program dokonał samodzielnie skalowania wykresu, bo domyślnie włączone jest autoskalowanie. Jeśli wykres wygląda dziwnie, możemy spróbować polecenia `set autoscale`. Wybrał też podziałkę dla skali oraz opisał linię wykresu w domyślny sposób za pomocą nazwy zbioru, jaki był wykreślany. Pamiętajmy, że ten opis możemy zmienić dodając do polecenia `plot'dane.txt'` polecenie `title'moje dane'`. Czyli wpiszemy: `plot'dane.txt' title'moje dane'`.



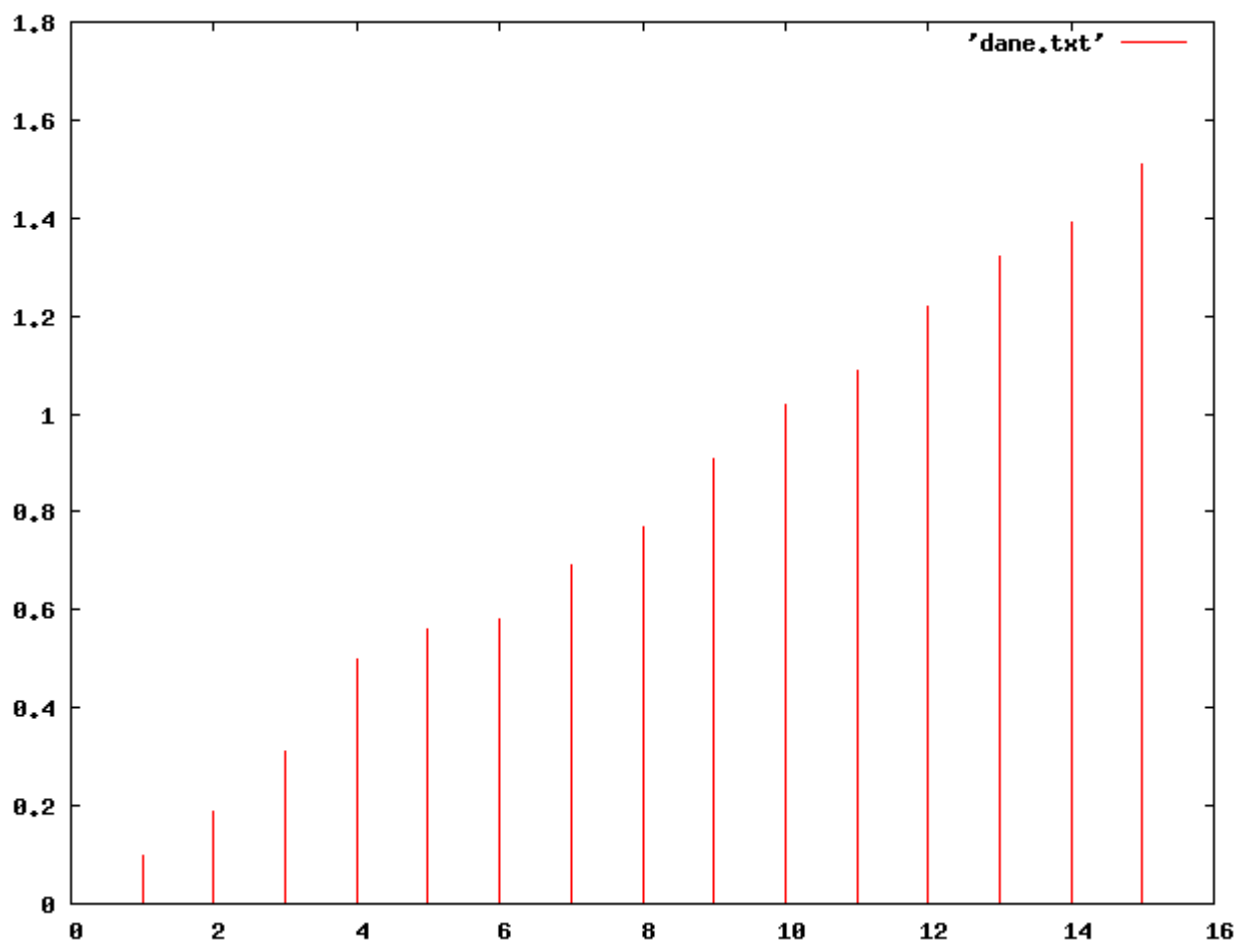
Działanie polecenia title.

Opis możemy wyłączyć poprzedzając polecenie plot poleceniem set noksy. W przypadku gdy chcemy przedstawić historię jakichś zdarzeń, np. wykres notowań na giełdzie, stosujemy połączenie wszystkich punktów, lecz musimy pamiętać, że jest to metoda, która może prowadzić do dziwnych rezultatów.

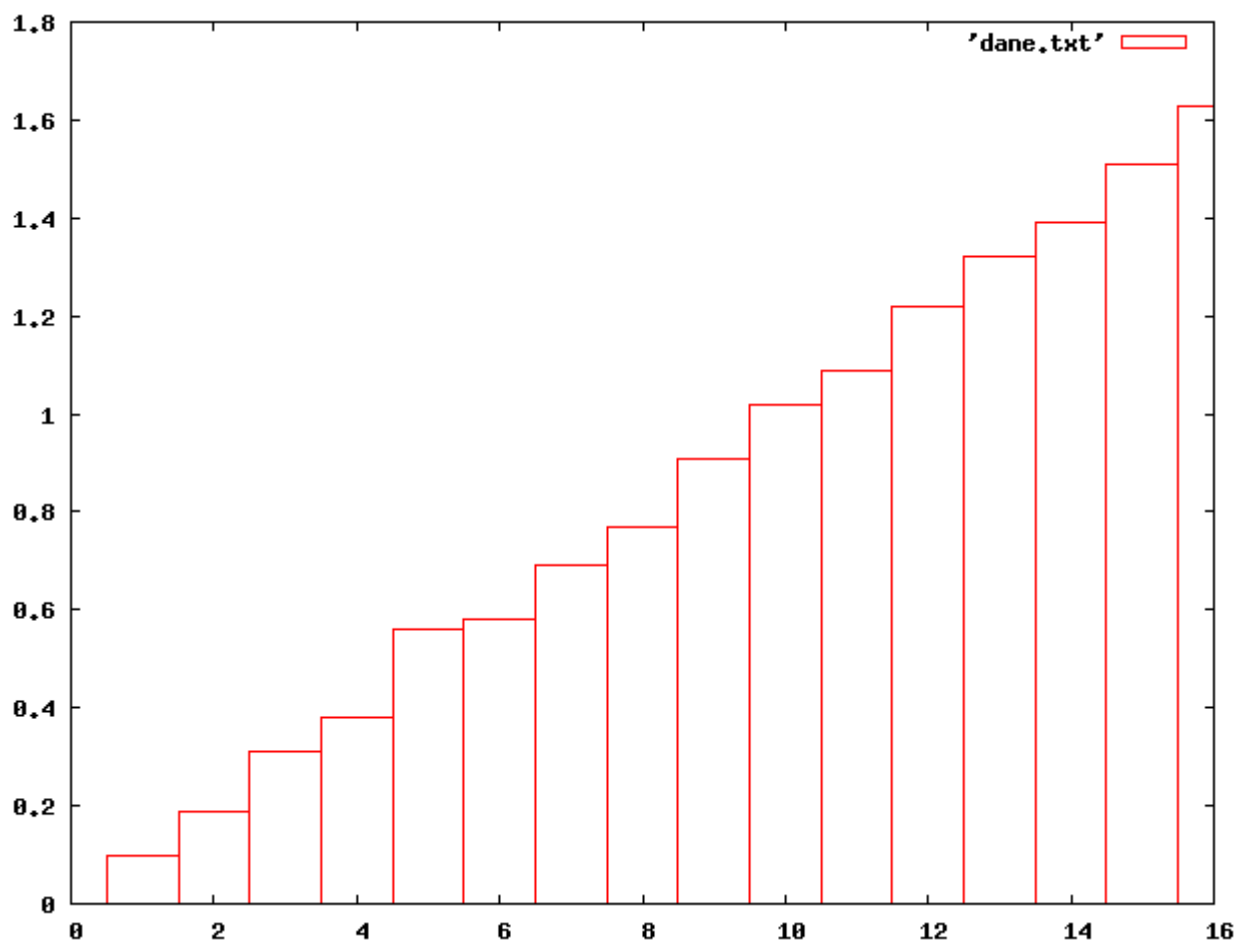
Zmianę sposobu przedstawienia wykresu włączamy słowem with, po którym następuje określenie stylu: np. linespoints, impulses, boxes, steps, oraz inne, które poznamy wpisując w wiersz poleceń gnu plota: help with.



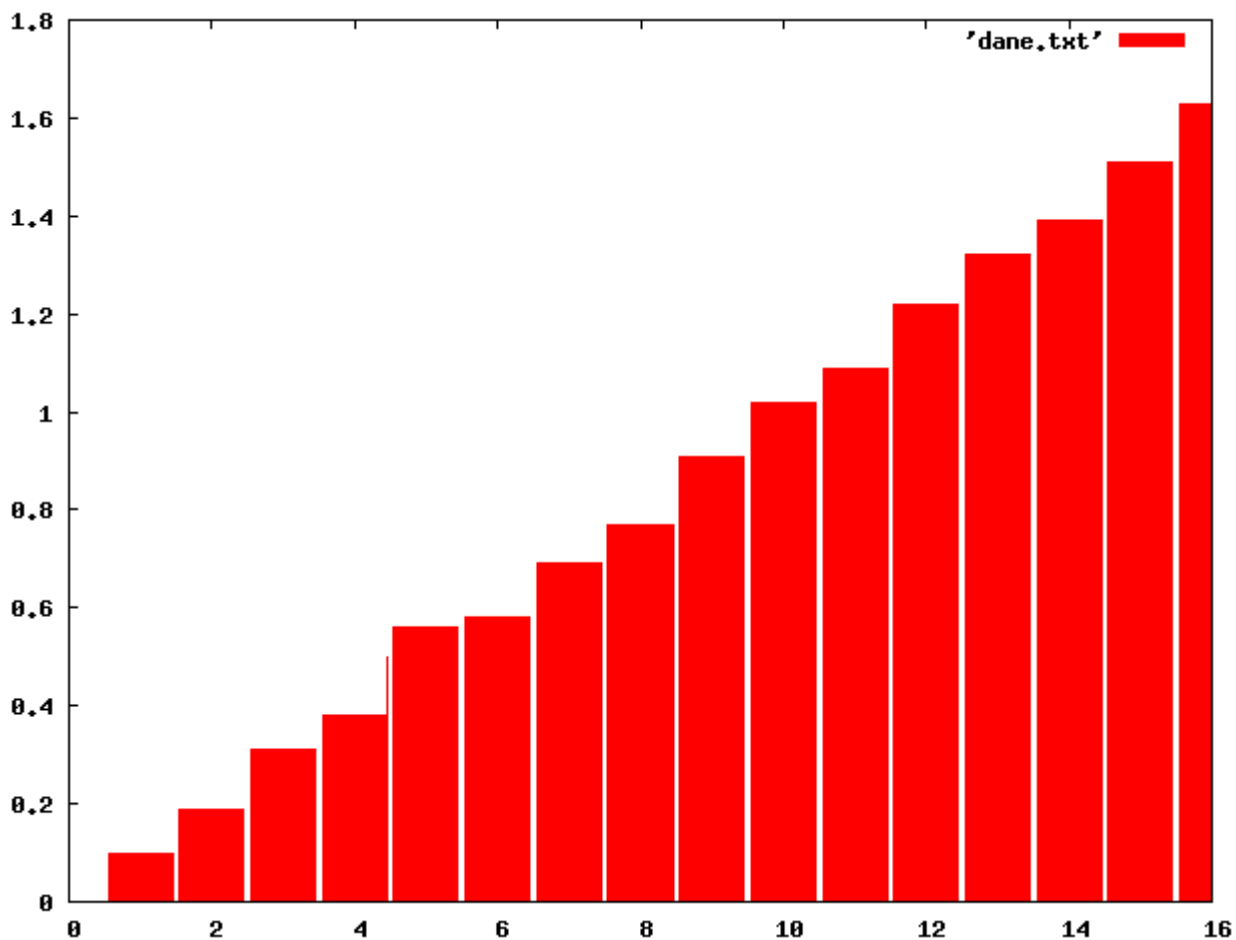
Styl linespoints.



Styl impulses.



Styl boxes.

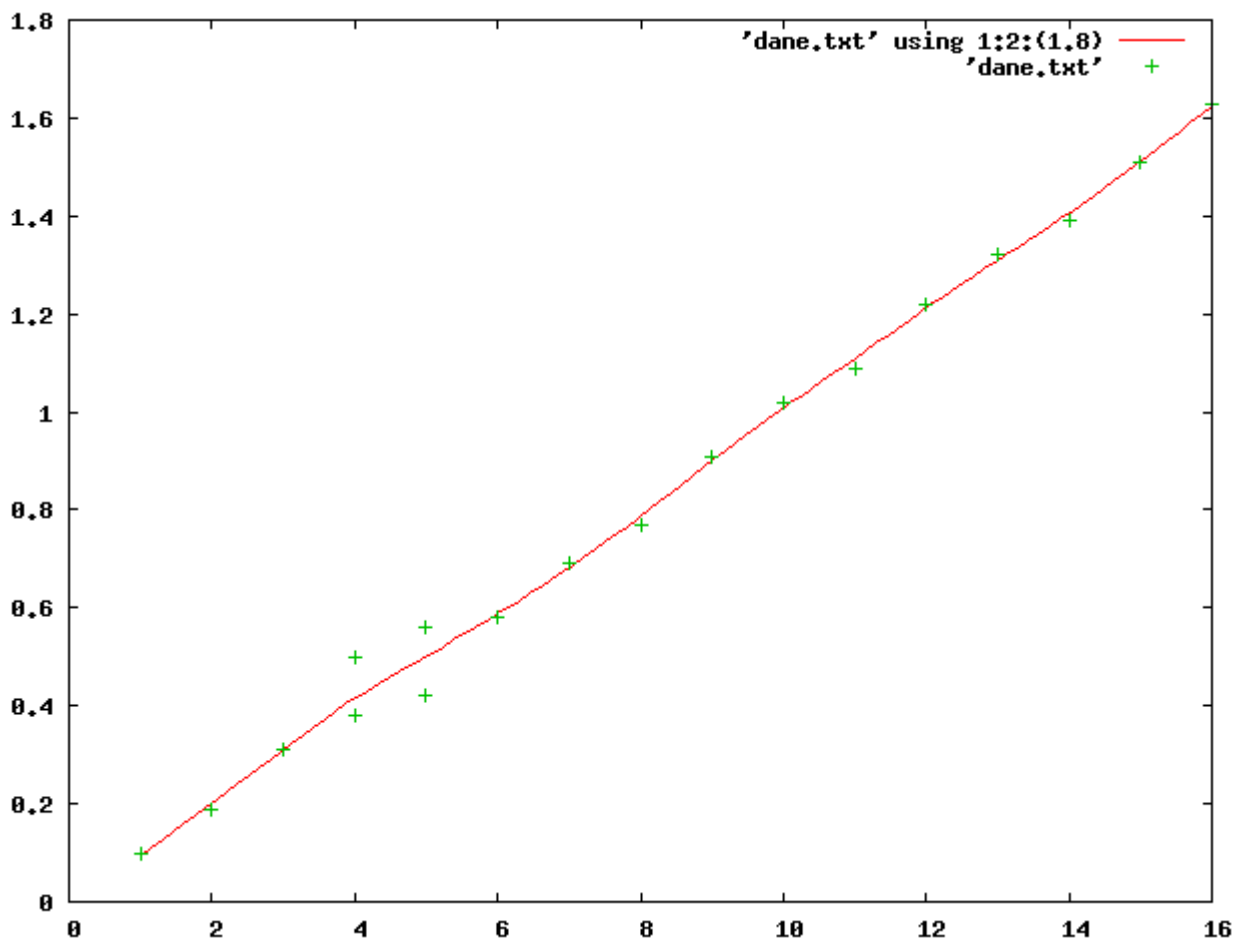


Styl boxes wypełniony.

Jednym z najpopularniejszych sposobów przedstawiania danych doświadczalnych jest przybliżanie ich za pomocą wygładzonej krzywej.

Uzyskamy to poleceniem `plot'dane.txt' using 1:2:(1.8) smooth acspline`. Pierwsza część polecenia jest jasna – wskazujemy plik, z którego chcemy wykonać wykres. Słowo kluczowe `using` wskazuje oczywiście kolumny danych, które mają być użyte. Algorytm potrzebuje wagi, z jaką ma wygładzać poszczególne punkty. Jeśli mamy kolumnę z wagami (różne wagi mogą wynikać np. z tego, że pomiary były prowadzone w różnych warunkach), napiszemy po prostu `using 1:2:3`, o ile wagi znajdują się w trzeciej kolumnie.

Zazwyczaj jednak stopień wygładzenia trzeba "wymyślić" i wówczas używamy takiego zapisu jak w naszym poleceniu. Mamy jedną wagę dla wszystkich punktów wpisaną w nawiasie. Słowo `smooth` nakazuje rysować krzywą wygładzoną. Po nim musi nastąpić określenie algorytmu. Wybraliśmy trochę na oślep `acspline`, ale działa i nie musimy się martwić, czy np. `bezier` nie byłoby lepsze. Aby na wykresie znalazły się także punkty danych, po poleceniu wyrysowania wygładzonej linii powtarzamy po przecinku nazwę zbioru z danymi.

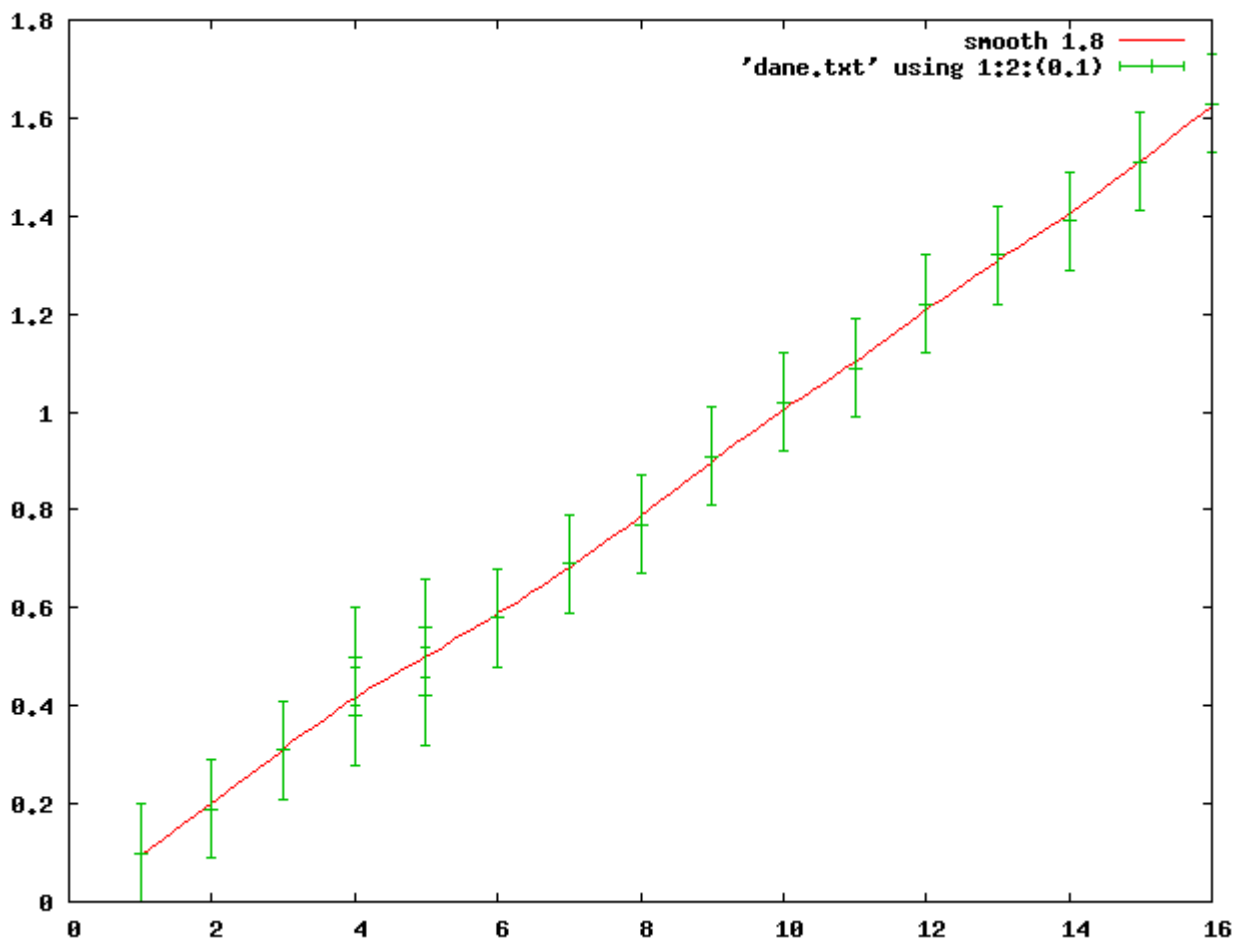


Wynik polecenia `plot'dane.txt' using 1:2:(1.8) smooth acspline,'dane.txt'`.

Polecenie `plot` działa nadal, nie wolno go powtórzyć.

Warto zauważyć, że w ten sam sposób na jednym wykresie możemy uzyskać kilka przebiegów z różnych plików, o ile dane mają podobne zakresy. Dla różnych danych możemy zdefiniować drugą oś x oraz y, ale o tym innym razem.

Ostatnim zagadnieniem, jakie omówimy, jest umieszczenie znacznika – jak się obecnie mówi – uchybu pomiarowego. Jego wartość powinniśmy mieć – w postaci osobnej kolumny – wyliczoną dla każdego pomiaru. Jeśli jednak jej nie mamy, do polecenia `plot'dane.txt'` dołączamy `using 1:2:(0.1) with errorbars`. W tym wypadku, po słowie `using`, tak jak przy wygładzaniu, wskazujemy kolumny z danymi, przy czym w nawiasie mamy wartość błędu, który zostanie naniesiony jako jednakowy dla wszystkich punktów.



Wynik polecenia `plot 'dane.txt' using 1:2:(1.8) smooth acspline title 'smooth 1.8', 'dane.txt' using 1:2:(0.1) with errorbars`.

Poniżej skrypt, który może posłużyć do zabawy z poleceniami. Tylko jedna linia z poleceniem `plot` jest odkomentowana. Jeśli chcesz uruchomić inne polecenie, odkomentuj je i zakomentuj bieżące. (Niestety, trzeba trochę uważać...).

```
set terminal png set output 'dane10.png'
```

```
# odkomentuj linię niżej, jeśli chcesz wyłączyć opis linii wykresu
```

```
#set nokey #plot 'dane.txt' #zamieniamy osie plot 'dane.txt' using 2:1
```

```
#plot 'dane.txt' title "moje dane"
```

```
#plot 'dane.txt' with linespoints
```

```
#plot 'dane.txt' with impulses
```

```
#polecenia do wydrukowania wykresu w stylu with boxes
```

```
# jeśli chcesz wykonać wykres ze słupkami wypełnionymi, odkomentuj dwie następne linie. Jeśli chcesz puste słupki, zostaw je zakomentowane. Przykład z "helpa"
```

```
#set boxwidth 0.9 relative
```

```
#set style fill solid 1.0
```

```
#plot'dane.txt' with boxes
```

```
#plot'dane.txt' with steps
```

```
#plot'dane.txt' using 1:2:(1.8) smooth acspline,'dane.txt'
```

```
#plot 'dane.txt' using 1:2:(1.8) smooth acspline title'smooth 1.8','dane.txt' using 1:2:(0.1) with  
errorbars
```

```
# Wykonywanie wykresu z pliku danych z kolumną uchybów pomiarowych
```

```
#plot 'dane0.txt' using 1:2:3 smooth acspline title'smooth 1.8','dane.txt' using 1:2:(0.1) with  
errorbars
```

W rzeczywistości ledwie "liznęliśmy" możliwości gnuplota. Polecam stronę <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>. Mamy jeszcze przed sobą szereg bardzo frapujących możliwości, jak "fitowanie" przebiegów funkcjami zmiany stylów linii, w tym wymuszanie kolorów i kształtów punktów, wykresy trójwymiarowe, operacje matematyczne na danych i wiele innych.