

Wprowadzenie do Scilab: funkcje i wykresy

Magdalena Deckert, Izabela Szczęch,
Barbara Wołyńska, Bartłomiej Prędko

Politechnika Poznańska, Instytut Informatyki

Narzędzia Informatyki



Agenda

1 Funkcje

- Korzystanie z funkcji
- Definiowanie funkcji
- Biblioteki funkcji
- Zarządzanie zmiennymi wyjściowymi
- Polecenie return

2 Wykresy

- Tworzenie wykresów
- Formatowanie wykresu
- Eksport wykresów
- Tworzenie wykresów z wieloma seriami
- Tworzenie wykresów z podwykresami
- Tworzenie wykresów 3D
- Polecenia Scilab do tworzenia wykresów

Korzystanie z funkcji

- Jeśli blok tych samych operacji jest wykonywany wielokrotnie, to warto te operacje zapisać w postaci funkcji. Dzięki temu będzie można ją wykorzystać wielokrotnie.
- Najprostszy sposób wywołania funkcji to:
`wynik = funkcja (parametr).`
- Parametry wejściowe funkcji nie są przez nią modyfikowane.
- Funkcje mogą przyjmować wiele parametrów wejściowych i zwracać więcej niż jedną wartość. W takim przypadku ogólniejsza postać wywołania jest następująca:
`[wyn_1, ..., wyn_n] = funkcja (param_1, ..., param_m).`

Polecenia Scilab do zarządzania funkcjami

Nazwa funkcji	Działanie
function endfunction	początek definicji funkcji koniec definicji funkcji
argn varargin varargout	liczba parametrów wejściowych lub wyjściowych zmienna liczba parametrów wejściowych zmienna liczba parametrów wyjściowych
fun2string get_function_path getd head_comments listfunctions macrovar	zwraca definicję funkcji jako ciąg znaków zwraca ścieżkę do pliku z definicją funkcji zwraca funkcje zdefiniowane w danym katalogu zwraca komentarze z nagłówka funkcji zwraca właściwości wszystkich funkcji zwraca zmienne danej funkcji

Tablica 1 : Polecenia Scilab do zarządzania funkcjami

Definiowanie funkcji

- Do definiowania funkcji wykorzystuje się słowa kluczowe *function* i *endfunction*.
- Każda funkcja składa się z *nagłówka* oraz *ciała* funkcji. Nagłówkiem funkcji jest jej nazwa wraz ze zdefiniowanymi parametrami wejściowymi oraz wyjściowymi. Na ciało funkcji składają się wszystkie polecenia zawarte między nagłówkiem funkcji a słowem kluczowym *endfunction*.
- Funkcje można definiować na 3 możliwe sposoby:
 - bezpośrednio w konsoli Scilab
 - w oddzielnym pliku uruchamianym w edytorze
 - za pomocą polecenia `exec`

Definiowanie pierwszej funkcji

Przykład 1

- Przejdź do konsoli Scilab.
- Zdefiniuj poniższą funkcję:

```
function r = kwadrat ( x )  
r = x^2  
endfunction
```
- Wywołaj powyższą funkcję i sprawdź, czy zwróciła poprawny wynik.

Definiowanie pierwszej funkcji - wynik

Konsola Scilab

```
-->function r = kwadrat (x)
-->r=x^2
-->endfunction

-->w = kwadrat(7)
w =
    49.

-->
```

Przeglądarka zmiennych

	Nazwa	Rozmiar	Typ	Widoczność
	w	1x1	Double	local

Historia poleceń

```
-- // -- 09/11/2012 15:53:45 -- //
function r = kwadrat (x)
r=x^2
endfunction
w = kwadrat(7)
```

Definiowanie pierwszej funkcji

Przykład 2

- Przejdź do konsoli Scilab.
- Zdefiniuj poniższą funkcję:

```
function r = kwadrat2 ( x )  
y = x^2  
endfunction
```

- Spróbuj wywołać powyższą funkcję. Czy operacja zakończyła się poprawnie?

Wywołanie funkcji z przykładu 2 zakończyło się niepowodzeniem, gdyż nie przypisaliśmy wartości do zmiennej wyjściowej. Interpreter Scilab sprawdza, czy parametry wyjściowe zostały zdefiniowane.

Definiowanie pierwszej funkcji - wynik

Konsola Scilab

```
-->function r = kwadrat2 (x)
-->y = x^2
-->endfunction

-->w = kwadrat2(4)
!--error 4
Niezdefiniowana zmienna: r
at line      4 of function kwadrat2 called by :
w = kwadrat2(4)

-->|
```

Przeglądarka zmiennych

	Nazwa	Rozmiar	Typ	Widoczność
	w	1x1	Double	local

Historia poleceń

```
--09/11/2012 15:59:42 --//
function r = kwadrat2 (x)
y = x^2
endfunction
w = kwadrat2(4)
```

Biblioteki funkcji

- Biblioteka funkcji jest zbiorem funkcji przechowywanych w wielu plikach, które są ze sobą powiązane w logiczny sposób.
- Biblioteki funkcji tworzy się, aby udostępnić zbiór pewnych wyspecjalizowanych funkcji. Takie podejście znacznie ułatwia tworzenie nowych skryptów, gdyż nie ma konieczności definiowania od nowa tych samych funkcji za każdym razem gdy są potrzebne.
- Pliki z definicjami funkcji powinny mieć rozszerzenie *.sci*.
- W każdym pliku może znajdować się wiele definicji funkcji, jednakże tylko pierwsza jest traktowana jako publiczna (czyli jest widoczna na zewnątrz). Pozostałe funkcje są funkcjami prywatnymi.
- Nazwa pliku z definicją funkcji powinna być taka sama jak nazwa funkcji publicznej.

Definiowanie biblioteki funkcji

Nazwa funkcji	Działanie
genlib	generowanie biblioteki z funkcjami z danego katalogu
lib	wczytanie danej biblioteki funkcji

Tablica 2 : Polecenia Scilab do zarządzania bibliotekami funkcji

Aby zdefiniować bibliotekę funkcji należy:

- skompilować pliki z definicjami funkcji do plików binarnych za pomocą polecenia *genlib*;
- załadować bibliotekę funkcji do środowiska Scilab za pomocą polecenia *lib*.

Definiowanie pierwszej biblioteki funkcji

Przykład 3

- Stwórz katalog mojabiblioteka zawierający 2 pliki z definicjami funkcji.
- Pierwszy plik kwadrat.sci zawiera następującą definicję:

```
function r = kwadrat ( x )  
r = x^2  
endfunction
```

- Drugi plik suma.sci zawiera następującą definicję:

```
function r = suma ( x )  
r = x+x  
endfunction
```

- Stwórz bibliotekę funkcji "mojabiblioteka".

Definiowanie pierwszej biblioteki funkcji

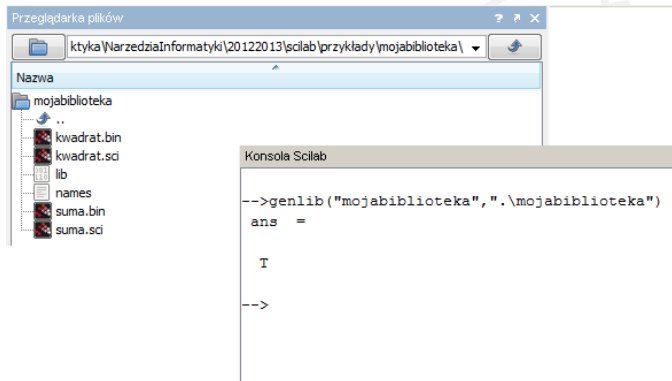
Przykład 3 c.d.

- Przejdź do konsoli Scilab i wykonaj polecenie:
`-->genlib("mojabiblioteka", ".\mojabiblioteka").`
- Powyższe polecenie generuje pliki binarne ze zdefiniowanymi funkcjami, plik *lib* z binarną wersją biblioteki oraz plik tekstowy *names* zawierający listę funkcji zdefiniowanych w bibliotece.
- Po wykonaniu polecenia *genlib* funkcje są gotowe do użycia.

W przypadku, gdy biblioteka funkcji była już wcześniej wygenerowana, to nie ma konieczności jej ponownego budowania, lecz można ją od razu wczytać do środowiska Scilab za pomocą metody *lib*:

```
-->mojabiblioteka = lib(".\mojabiblioteka").
```

Definiowanie pierwszej biblioteki funkcji - wynik



Wczytanie istniejącej biblioteki funkcji - wynik

```
Konsola Scilab

Wykonanie rozruchu:
  ładowanie środowiska początkowego

-->mojabiblioteka = lib("./mojabiblioteka")
mojabiblioteka =

Lokalizacja plików funkcyjnych: D:\Dydaktyka\NarzedziaInformatyki\20122013\scilab\przykłady\mojabib

kwadrat  suma

-->kwadrat(7)
ans =

    49.

-->suma(7)
ans =

    14.

-->function r = s_k (x)
-->y = suma(x)
-->r = kwadrat(y)
-->endfunction

-->s_k(3)
ans =

    36.

-->
```

Zarządzanie zmiennymi wyjściowymi

Przykład 4

- Zdefiniuj następującą funkcję:

```
function [s, i] = sumil (a, b)
s = a+b
i = a*b
endfunction
```

- Istnieje możliwość odczytania 0, 1 lub 2 zmiennych wyjściowych.
- W przypadku, gdy nie odczytujemy żadnej zmiennej, to wartość pierwszego parametru wyjściowego zapisywana jest do zmiennej *ans*.
- Sprawdź w konoli Scilab wszystkie możliwości odczytywania zmiennych wyjściowych.

Zarządzanie zmiennymi wyjściowymi - wynik

Konsola Scilab

```
-->function [s, i] = sumil (a, b)
-->s = a+b
-->i = a*b
-->endfunction

-->sumil(2,3)
ans =

    5.

-->s = sumil(2,3)
s =

    5.

-->[s, i] = sumil(2,3)
i =

    6.
s =

    5.

-->
```

Przeglądarka zmiennych

	Nazwa	Rozmiar	Typ	Widoczność
	i	1x1	Double	local
	s	1x1	Double	local
	ans	1x1	Double	local
	w	1x1	Double	local

Historia poleceń

```
// --09/11/2012 16:05:13 --//
function [s, i] = sumil (a, b)
s = a+b
i = a*b
endfunction
sumil(2,3)
s = sumil(2,3)
[s, i] = sumil(2,3)
```

Polecenie *return*

Polecenie *return* służy do przerywania wykonywania danej funkcji. Polecenie to powinno być używane z rozważą, gdyż komplikuje ono debugowanie działania funkcji.

Przykład 5

- Zdefiniuj następującą funkcję:

```
function r = pierwiastek (x)
if (x < 0) then
disp("Podano liczbę ujemną!")
return
end
r = sqrt(x)
endfunction
```

- Wykonaj powyższą funkcję dla następujących wartości $x=-4$, $x=0$ oraz $x=4$.

Polecenie *return* - wynik

Konsola Scilab

```
-->function r = pierwiastek (x)
-->if (x<0) then
-->disp("Podano liczbe ujemna!")
-->return
-->end
-->r = sqrt(x)
-->endfunction

-->pierwiastek(-4)

Podano liczbe ujemna!
!--error 4
Niezdefiniowana zmienna: r
at line 4 of function pierwiastek called by :
pierwiastek(-4)

-->pierwiastek(4)
ans =

    2.

-->pierwiastek (0)
ans =

    0.

-->
```

Przeglądarka zmiennych

	Nazwa	Rozmiar	Typ	Widoczność
	ans	1x1	Double	local
	i	1x1	Double	local
	s	1x1	Double	local
	w	1x1	Double	local

Historia poleceń

```
--// --09/11/2012 16:09:36 --//
function r = pierwiastek (x)
if (x<0) then
disp("Podano liczbe ujemna!")
return
end
r = sqrt(x)
endfunction
pierwiastek(-4)
pierwiastek(4)
pierwiastek (0)
```

Wykresy

- Tworzenie wykresów jest bardzo pomocne w analizowaniu danych i tworzeniu raportów.
- Scilab umożliwia tworzenie różnych typów wykresów 2D lub 3D oraz ich dostosowywanie.
- Scilab umożliwia także eksport stworzonych wykresów do różnego typu plików graficznych.

Pierwszy wykres 2D

Przykład 6

- Tworzenie wykresów należy rozpocząć od zdefiniowania funkcji jaka ma zostać wykreślona.
- Przejdź do konsoli Scilab i zdefiniuj następującą funkcję:

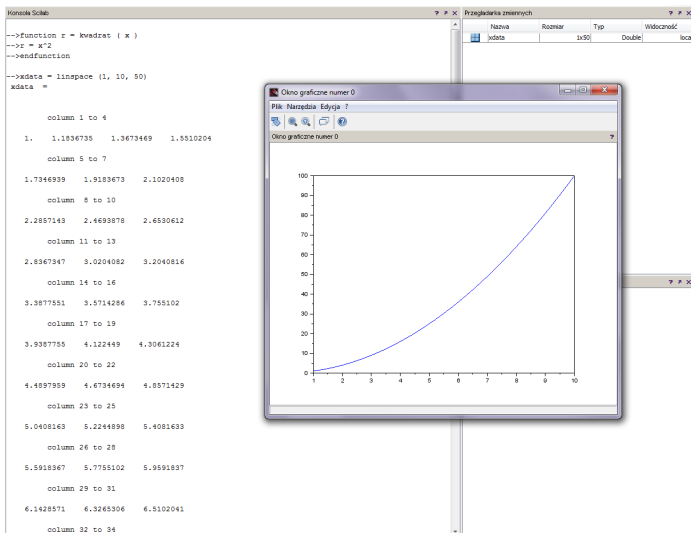
```
function r = kwadrat ( x )  
r = x^2  
endfunction
```

Pierwszy wykres 2D

Przykład 6 c.d.

- W kolejnym kroku definiujemy serię danych dla osi X. Za pomocą funkcji *linspace* tworzymy określoną liczbę punktów z podanego zakresu. Przykładowe polecenie
--> `xdata = linspace (1, 10, 50)`
tworzy 50 punktów z zakresu `[1;10]`.
- Stworzona seria danych jest przekazywana jako parametr wejściowy do wcześniej zdefiniowanej funkcji --> `ydata = kwadrat (xdata)`. W rezultacie uzyskujemy wektor 50 wartości y – *ydata*.
- W ostatnim kroku wywołujemy funkcję tworzącą wykres 2D z uzyskanych wcześniej danych --> `plot (xdata, ydata)`.
- Drugim parametrem zamiast wartości *ydata* mogłaby być wcześniej zdefiniowana funkcja *kwadrat*.

Pierwszy wykres 2D - wynik



Formatowanie wykresów

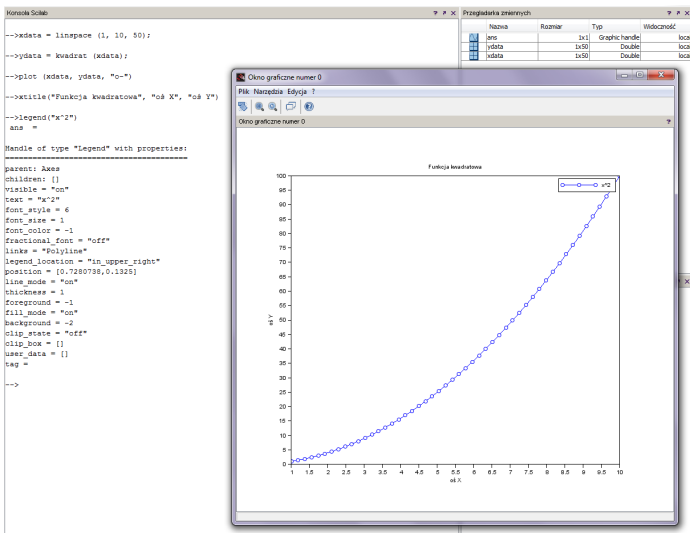
- System graficzny Scilab umożliwia zorientowany obiektowo dostęp do poszczególnych elementów wykresu. Pozwala on m.in. na zmianę typu linii związanej z danym wykresem, dodanie tytułu wykresu lub podpisów poszczególnych osi, modyfikację zawartość legendy, itp.
- Wiele spośród własności wykresu można modyfikować różnymi funkcjami takimi jak:
 - Funkcja *title* wstawia podany tytuł wykresu
 - Funkcja *xtitle* wstawia podany tytuł wykresu wraz z podpisami osi.
 - Funkcja *legend* wstawia legendę wraz z podanymi nazwami serii danych.
 - Za pomocą trzeciego parametru funkcji *plot* możliwa jest modyfikacja typu linii związanego z daną funkcją.

Formatowanie pierwszego wykresu 2D

Przykład 7

- Sprawdź, czy funkcja *kwadrat* jest nadal zdefiniowana w środowisku Scilab. Jeśli nie, to załaduj ją ponownie.
- Wygeneruj ponownie serię danych *xdata*:
`--> xdata = linspace (1, 10, 50);`
- Stwórz wartości zmiennej *y*:
`--> ydata = kwadrat (xdata);`
- Wykreśl funkcję *kwadrat* za pomocą polecenia:
`--> plot (xdata, ydata, "o-").`
- Dodaj tytuł wykresu oraz podpisy osi:
`--> xtitle ("Funkcja kwadratowa", "oś X", "oś Y").`
- Dodaj legendę do wykresu:
`--> legend("x^2").`

Formatowanie pierwszego wykresu 2D - wynik



Eksport wykresów

- Eksportowanie wykresów w środowisku Scilab możliwe jest na dwa sposoby:
 - wybierając opcję z menu
 - z poziomu konsoli za pomocą określonych poleceń
- Pierwszy ze sposobów polega na wybraniu opcji menu *Plik* → *Eksportuj do...* W tym momencie pozostaje tylko podanie nazwy pliku i wybór typu pliku.
- Alternatywny sposób polega na uruchomieniu odpowiedniego polecenia Scilab, które stworzy nam dany plik graficzny z wybranego wykresu. Ogólne wywołanie funkcji eksportujących wygląda następująco: `xs2*(numer_okna, nazwa_pliku)`, gdzie w miejsce * należy wpisać pożądany typ pliku graficznego.
- Scilab eksportuje wykresy do różnych typów plików grafiki wektorowej lub bitmapowej.

Polecenia Scilab do eksportu wykresów

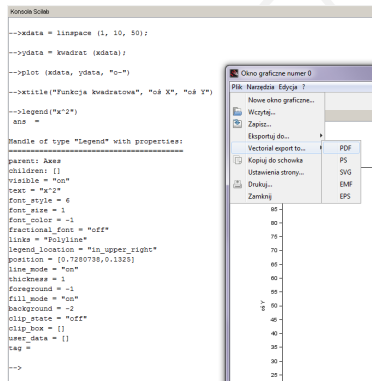
Funkcja	Działanie
xs2png	eksport do pliku PNG
xs2pdf	eksport do pliku PDF
xs2svg	eksport do pliku SVG
xs2eps	eksport do pliku Encapsuated PostScript
xs2ps	eksport do pliku PostScript
xs2emf	eksport do pliku EMF
xs2fig	eksport do pliku FIG
xs2gif	eksport do pliku GIF
xs2jpg	eksport do pliku JPG
xs2bmp	eksport do pliku BMP
xs2ppm	eksport do pliku PPM

Tablica 3 : Polecenia Scilab do eksportu wykresów

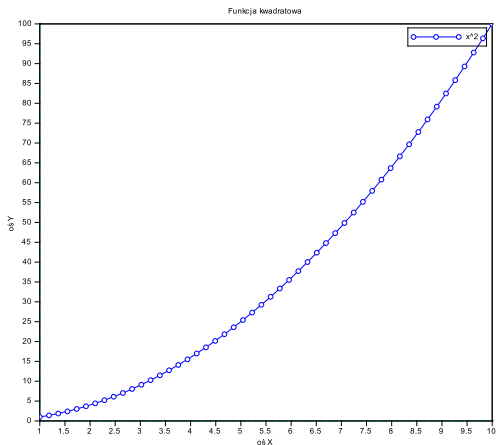
Eksport wykresu 2D

Przykład 8

- Wyeksportuj wykres z przykładu 7 do formatu PDF.



Eksport wykresu 2D - wynik



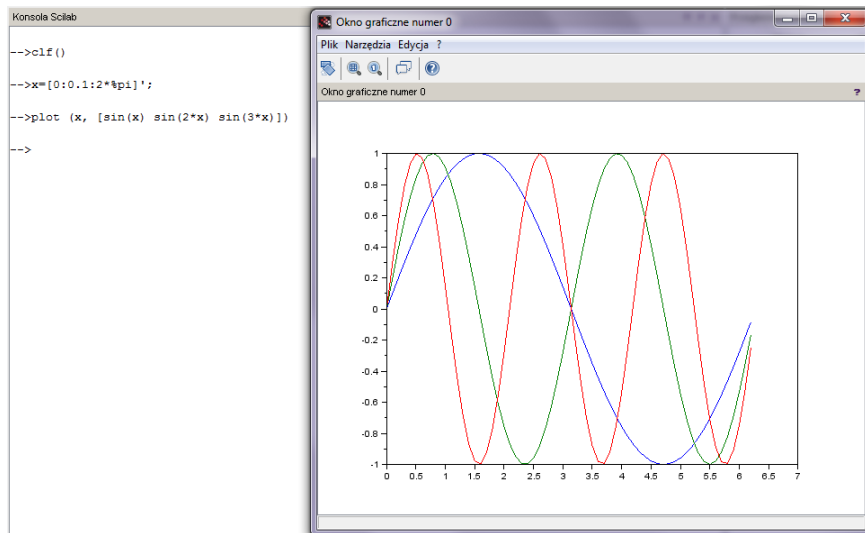
Tworzenie wykresu 2D z kilkoma seriami

Przykład 9

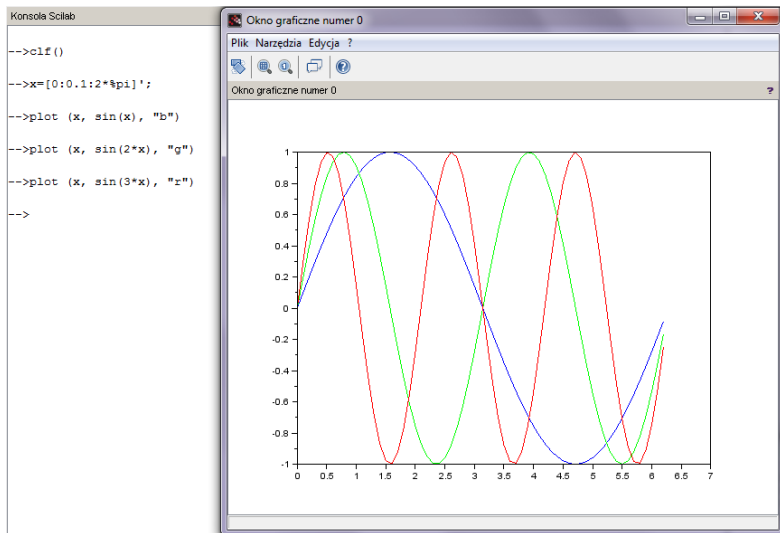
- Wyczyść aktualne okno z grafiką za pomocą polecenia: `--> clf()`.
- Wygeneruj serię danych x : `--> x=[0:0.1:2*%pi]'`;
- Wykreśl kilka funkcji na jednym wykresie za pomocą polecenia:
`--> plot (x, [sin(x) sin(2*x) sin(3*x)])`.
- Taki sam efekt uzyskalibyśmy wywołując kolejno funkcje *plot*:
`--> plot (x, sin(x), "b")`
`--> plot (x, sin(2*x), "g")`
`--> plot (x, sin(3*x), "r")`

UWAGA: Okna z wykresem nie należy zamykać.

Tworzenie wykresu 2D z kilkoma seriami - wynik



Tworzenie wykresu 2D z kilkoma seriami - wynik



Tworzenie wykresu 2D z kilkoma podwykresami

Przykład 10

- Wyczyść aktualne okno z grafiką za pomocą polecenia: `--> clf()`.
- Wygeneruj serię danych x : `--> x=[0:5:360]'` ;.
- Stwórz wykres z 4 podwykresami za pomocą polecenia:

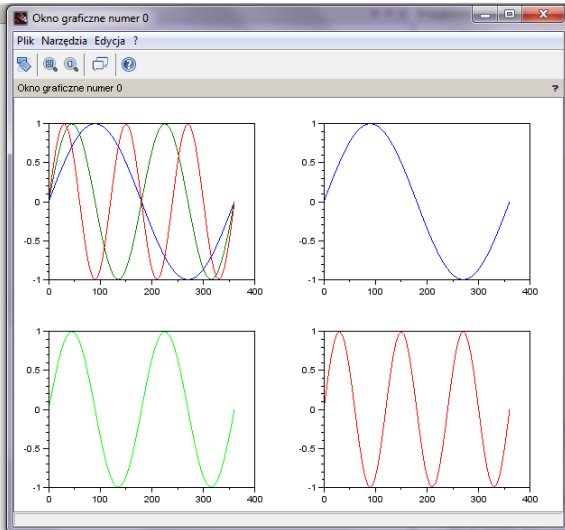
```
--> subplot(221)
--> plot (x, [sind(x) sind(2*x) sind(3*x)])
--> subplot (222)
--> plot (x, sind(x), "b")
--> subplot (223)
--> plot (x, sind(2*x), "g")
--> subplot(224)
--> plot (x, sind(3*x), "r")
```

UWAGA: Okna z wykresem nie należy zamykać.

Tworzenie wykresu z kilkoma podwykresami - wynik

Konsola Scilab

```
-->clf()
-->x=[0:5:360]';
-->subplot(221)
-->plot (x, [sind(x) sind(2*x) sind(3*x)])
-->subplot (222)
-->plot (x, sind(x), "b")
-->subplot (223)
-->plot (x, sind(2*x), "g")
-->subplot(224)
-->plot (x, sind(3*x), "r")
-->
```

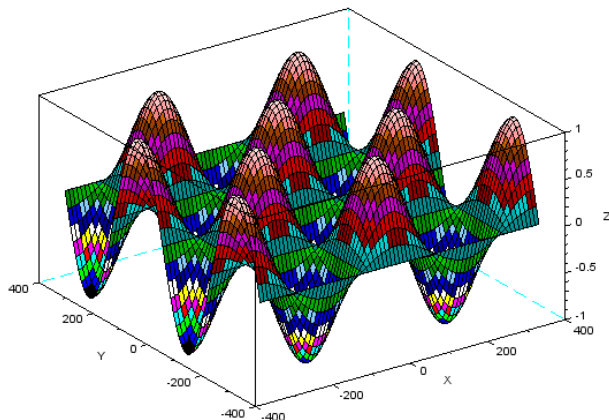


Pierwszy wykres 3D

Przykład 11

- Wyczyść aktualne okno z grafiką za pomocą polecenia: `--> clf()`.
- Wygeneruj serię danych x : `--> x=[-360:10:360];`.
- Wygeneruj serię danych y : `--> y=[-360:10:360];`.
- Stwórz tablicę funkcji $z = \sin(x) \cdot \cos(y)$:
`--> z=sind(x')*cosd(y);`.
- Stwórz wykres powierzchniowy za pomocą polecenia:
`--> surf(x,y,z).`

Pierwszy wykres 3D - wynik



Polecenia Scilab do tworzenia wykresów

Funkcja	Działanie
plot	wykres 2D
surf	wykres 3D
contour	wykres konturowy
pie	wykres kołowy
histplot	histogram
bar	wykres kolumnowy
barh	wykres słupkowy
hist3d	histogram 3D
polarplot	wykres ze współrzędnymi biegunowymi
Matplot	kolorowy wykres 2D macierzy
Sgrayplot	wygładzony kolorowy wykres powierzchniowy 2D
grayplot	kolorowy wykres powierzchniowy 2D

Tablica 4 : Polecenia Scilab do tworzenia wykresów

Literatura

Materiały przygotowane na podstawie "Introduction to Scilab".
<http://www.scilab.org/support/documentation/tutorials>