

Wprowadzenie

Scilab jest środowiskiem programistycznym i numerycznym dostępnym za darmo z INRIA (*Institut Nationale de Recherche en Informatique et Automatique*). Jest programem podobnym do MATLABa oraz jego darmowego 'klonu' OCTAVE'a.

Scilab jest samodzielnym programem zawierającym wiele wbudowanych funkcji numerycznych oraz graficznych. Jest wyposażony w język programowania.

Wersje instalacyjne programu Scilab można pobrać ze strony <http://www.scilab.org>. Również na tej stronie znajdują się linki do dokumentacji. Wpisując w wyszukiwarkę na przykład słowa "Scilab tutorial" można znaleźć linki do różnego rodzaju podręczników i wykładów wprowadzających do Scilaba.

Wprowadzenie do Scilaba:

<http://www.scilab.org/content/download/1754/19024/file/introscilab.pdf>

Wprowadzenie do Scilaba:

<http://www.iecn.u-nancy.fr/~szulc/docpl.pdf>

Help - uzyskiwanie pomocy

- `help` polecenie, np. `help sin`
- `apropos` polecenie wyświetla informacje związane z danym poleceniem
- Na stronie <http://www.scilab.org/product/man>

Okna

Podczas pracy ze Scilabem, wykorzystuje się okna:

- Konsola Scilab
- Edytor Scilab (wywołanie: Applications/SciNotes)
- ? - Help (Przeglądarka pomocy)

Pierwsze kroki

Dobrym sposobem zapoznania się z działaniem programu, który mam nadzieję, że zachęci Czytelników do kontynuacji, jest wykreślanie wykresów funkcji. O niektórych sprawach teraz tylko pobieżnie wspomnimy – bardziej formalnie zostaną omówione później. Na razie chcemy narysować wykres funkcji.

Tworzenie wykresu składa się najczęściej z następujących etapów:

- utworzenie ciągu wartości 'x-ów'
- utworzenie ciągu wartości 'y-ków'
- rysowanie
- zapisanie rysunku do pliku graficznego (opcjonalnie).

Uwaga: Robienie wykresu na komputerze wykorzystuje dyskretne przedstawienie funkcji w postaci ciągu wartości 'punktowych', zawierających współrzędne punktów, w oparciu o które tworzony jest wykres. Im więcej jest takich punktów, tym dokładniejszy jest wykres.

Wartości 'x-ów':

Wartości 'x-ów' można w Scilabie zdefiniować na kilka sposobów, wpisując odpowiednie polecenie na konsoli.

`x=[0,1,2,3,4,5,5.5,10,20];` - ciąg wartości

`x=(-10:0.1:10);`

(wartość początkowa : krok : wartość końcowa)

`x=linspace(0, 3.141592, 20);`

(wartość początkowa, wartość końcowa, ile wartości)

Uwagi:

- `;` (średnik) na końcu sprawia, że wyznaczane wartości nie są wypisywane na konsoli;
- określenie 'x' jest symboliczne, równie dobrze tworzony ciąg wartości może mieć nadaną inną nazwę, np.: `z=(0: 1 : 100)`, albo `ala=(-5 : 0.1 : 5)`;
- jako wartość π można wpisać `%pi`, czyli: `x=linspace(0,%pi,20)`.
- Przy okazji zwracamy uwagę na fakt, że część ułamkowa jest oddzielona od części całkowitej **kropką**.

Wartości 'y-ków':

Uwaga: określenie 'y-ki' jest symboliczne; tworzony obiekt może mieć dowolną nazwę.

Wartości 'y-ków' można tworzyć na wiele sposobów, na razie wykorzystamy najprostsze. W poniższych przykładach wartości są budowane w oparciu o wcześniej utworzony ciąg 'x-ów'.

```
y=x;
```

```
y1=2*x;
```

```
z=2*x-1;
```

```
fun=sin(x)+cos(2*x);
```

```
y2=x^3
```

```
g=tan(x)^2;
```

Ale wartości 'y-ków' mogą również być zdefiniowane jako ciąg wartości.

```
y=[1,3,5,15,-1]
```

Uwagi:

W wyrażeniach wykorzystuje się operatory arytmetyczne `+`, `-`, `*`, `/`.

Znak `^` oznacza potęgowanie.

Można wykorzystywać wcześniej zdefiniowane wartości, np. `y3=y1+y2`;

Wykorzystaliśmy też funkcje Scilaba: `sin`, `cos` i `tan`. Scilab dysponuje dużym zestawem swoich funkcji (inaczej mówiąc, przepisami na obliczanie różnych rzeczy), część z nich poznamy w trakcie tych zajęć.

Wszystkie podane wyżej wyrażenia dotyczyły operacji wykonywanych na *ciągu wartości* określonych wspólną nazwą `x`. Wynikiem jest też ciąg wartości.

Działanie (przykładowe) `z=x*x` spowoduje pojawienie się komunikatu o błędzie. Przyczyna zostanie wyjaśniona później. Na razie zastąpmy to wyrażenie wyrażeniem: `z=x^2`.

Rysowanie – funkcja `plot`

```
plot(x,y)
```

Uwaga przed pierwszym przykładem

Zaleca się, aby każdy program rozpoczynać poleceniem:

```
clear ();
```

Polecenie to powoduje 'wyczyszczenie' wszystkich zmiennych – upewniamy się w ten sposób, że podczas obliczeń nie pojawią się żadne 'stare' wartości.

Pierwszy przykład:

```
clear();
```

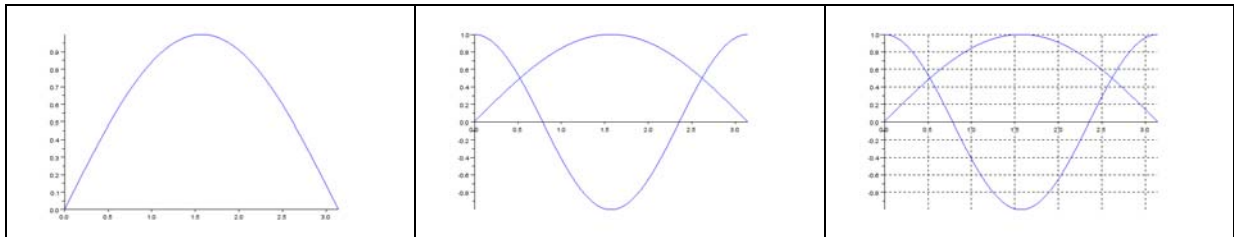
```
x=linspace(0, %pi, 50);
```

```
y=sin(x);
```

```
plot(x,y);
```

```
y1=cos(2*x);
```

```
plot(x,y1);  
xgrid();
```



Zapisanie rysunku do pliku:

W oknie graficznym (interakcyjnie):

Plik / eksportuj do / ... wybrać typ pliku (PNG, GIF, JPG,...), podać nazwę pliku

Poprzez wpisanie w oknie konsoli odpowiedniego polecenia, np:

```
xs2png (numer_okna_graficznego, 'nazwa_pliku.png')
```

Uwagi:

numer_okna_graficznego – jest wyświetlony w pasku tytułowym okna. Standardowo pierwsze utworzone okno ma numer 0.

Scilab wyróżnia katalog bieżący (Plik/ Wyświetl katalog bieżący). O ile nazwa pliku nie zostanie poprzedzona ścieżką dostępu, plik zostanie zapisany w katalogu bieżącym. Zmiana katalogu bieżącego: Plik/ Zmiana bieżącego katalogu...)

Inne formaty plików graficznych, to (między innymi):

eps – funkcja **xs2eps**,

postscript – **xs2ps**,

pdf – **xs2pdf**,

gif – **xs2gif**,

jpg – **xs2jpg**.

Więcej informacji o tworzeniu wykresów:

Kolejne polecenia 'plot' powodują dodanie ('dorysowanie') kolejnego wykresu w bieżącym oknie.

Operacje na oknach:

clf() – wyczyszczenie bieżącego okna.

clf(1) – wyczyszczenie okna nr 1.

scf(1) – utworzenie okna o numerze 1.

xdel() – usunięcie bieżącego okna.

xdel(1) – usunięcie okna o numerze 1.

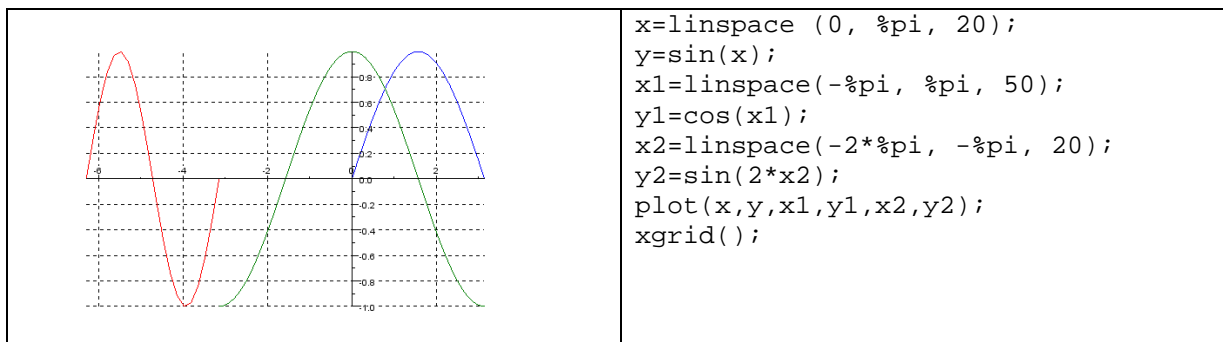
Kilka wykresów (różnych) funkcji na jednym rysunku:

Polecenie:

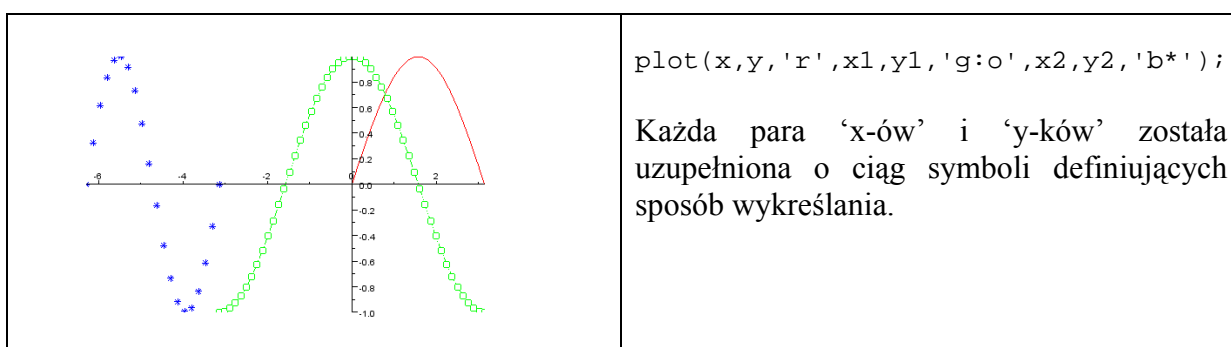
```
plot (x,y,x,y1);
```

spowoduje narysowanie wykresu funkcji przechowywanej w wektorze y , oraz drugiej funkcji przechowywanej w wektorze $y1$. Obydwa wykresy wykorzystują ten sam zestaw x -ów.

Ale również można na jednym rysunku umieszczać wykresy zdefiniowane dla różnych zestawów argumentów (zakresów lub 'gęstości' x -ów).



Kolory są ustalane automatycznie. Poprzez odpowiednie zdefiniowanie parametrów można sterować zarówno kolorami, jak i rodzajem linii oraz markerów; porównajmy z poprzednim wykresem:



Kolory:

symbol	kolor
r	czerwony
g	zielony
b	niebieski
c	cyjan
m	magenta
y	żółty
k	czarny
w	biały

```

plot(x,y,'r',x1,y1,'r',x2,y2);

```

Style wykreślenia linii:

- plot(x,y,'-'); – linia ciągła (domyślnie)
- plot(x,y,'--'); – linia przerywana
- plot(x,y,':'); – linia kropkowa
- plot(x,y,'-.'); – linia kreskowo-kropkowa

Znaczniki:

Symbol	znacznik	Symbol	znacznik
+	plus	^	▲
o	kółko	v	▼
*	gwiazdka	>	►
.	kropka	<	◄
x	krzyżyk	'pentagram'	gwiazdka

			pięciokamienna
'square' lub 's'	kwadracik	'none'	brak znacznika -
'diamond' lub 'd'	◇		domyślnie

Uwaga: Domyślnie znaczniki nie są rysowane. Jeśli wskaże się tylko znacznik, to trzeba jawnie podać symbol stylu wykreślenia linii. W przeciwnym przypadku wykres nie będzie zawierał linii.

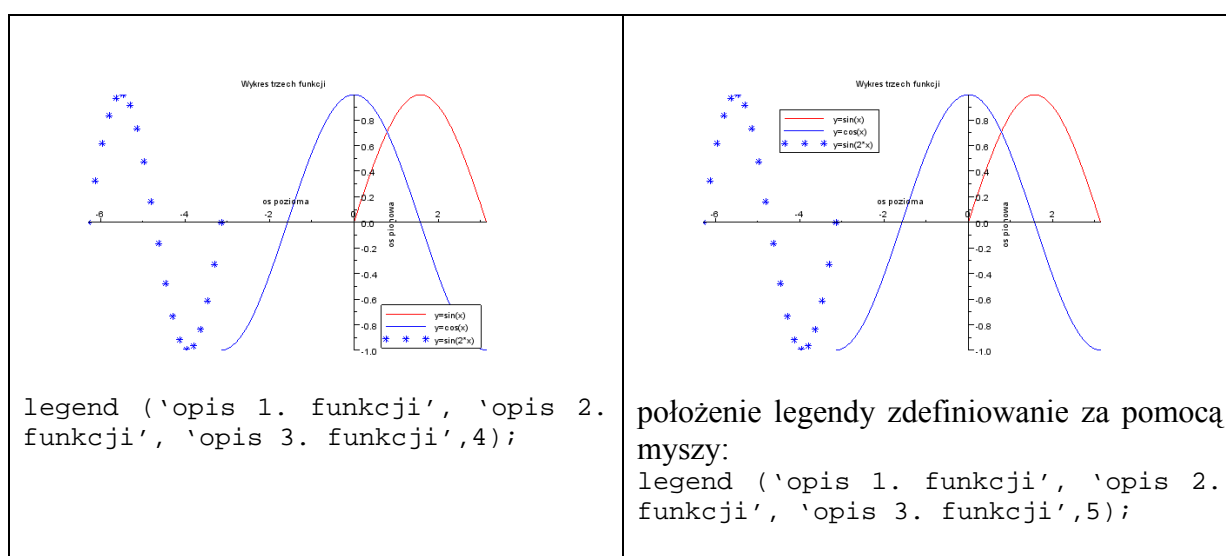
Opisywanie wykresów: tytuł, opisy osi, legenda

title ('Tytuł wykresu');

xtitle('Tytuł wykresu', 'opis osi x-ow', 'opis osi pionowej');

legend ('opis 1. funkcji', 'opis 2. funkcji', 'opis 3. funkcji');

Wszystkie opisy odnoszą się do bieżącego okna.



Uwaga: Legendę można uzupełnić o informację o jej położeniu na rysunku poprzez podanie na końcu opcjonalnego parametru. Domyślnym położeniem legendy jest prawy górny róg.

Polecenie:

legend ('opis 1. funkcji', 'opis 2. funkcji', 'opis 3. funkcji',4);

spowoduje umieszczenie legendy w lewym dolnym rogu.

Przykładowe inne możliwości – można podawać albo opis liczbowy, albo opis słowny (w apostrofach):

1 lub "in_upper_right" – prawy górny róg, przyjmowane domyślnie

2 lub "in_upper_left" – lewy górny róg

3 lub "in_lower_left" – lewy dolny róg

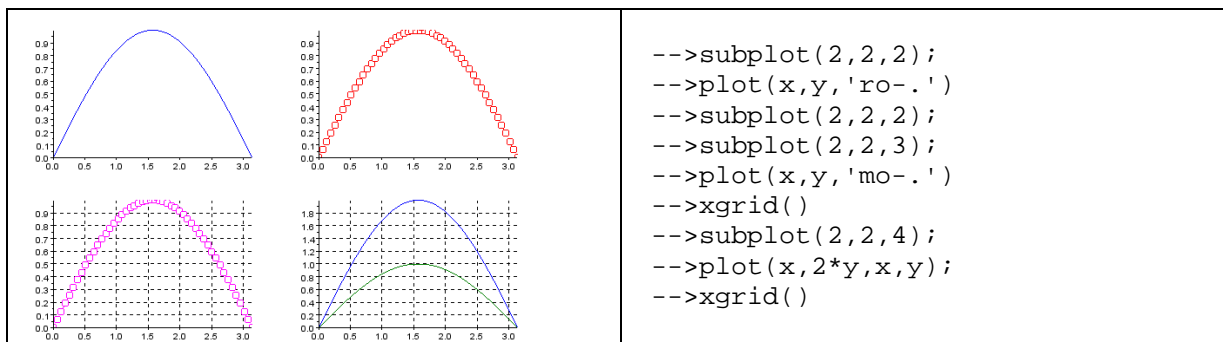
4 lub "in_lower_right" – prawy dolny róg

5 lub "by_coordinates" – położenie legendy zdefiniowanie za pomocą myszki w oknie graficznym.

Kilka rozłącznych wykresów w jednym oknie – subplot

Przykład 4 wykresów rozmieszczonych w 2 kolumnach i 2 wierszach.

	<pre>-->clf() -->subplot(2,2,1); -->plot(x,y)</pre>
--	--



Zadania:

Narysować trójkąt o wierzchołkach w punktach: (0,0), (10, 1), (5,7).

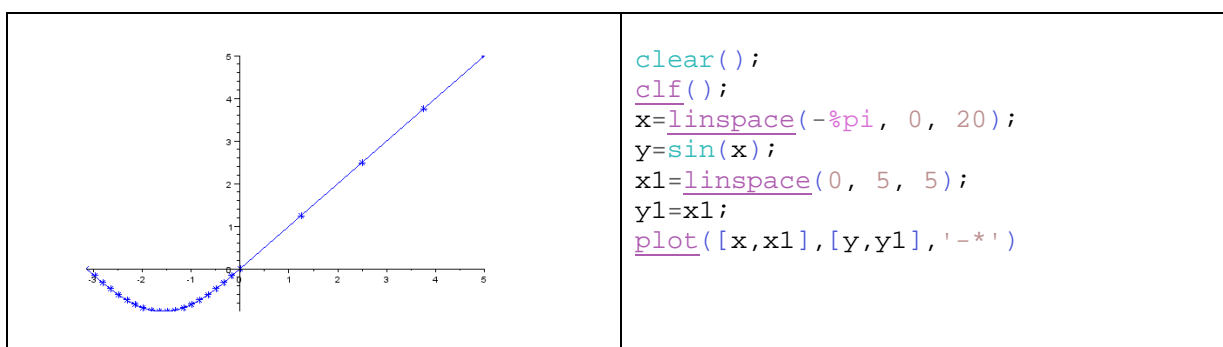
```
clear();
x=[0,10,5,0];
y=[0,1,7,0];
plot(x,y)
```

Narysować prostokąt o wierzchołkach w punktach (0,0), (10, 0), (10, 4), (0,4) oraz punktem w innym kolorze zaznaczyć jego środek.

```
clear();
clf();
x=[0,10,10,0,0];
y=[0,0,4,4,0];
plot(x,y,'r');
x1=[5];
y1=[2];
plot(x1,y1,'sg');
```

Narysować wykres funkcji opisanej wzorem:

$$y = \begin{cases} \sin(x), & x \in [-\pi, 0) \\ x, & x \in [0, 5) \end{cases}$$



Uwaga: Kolory w powyższym kodzie pochodzą z edytora Scilaba i mają za zadanie zwiększyć czytelność kodu.

SCILAB

Materiały opracowała Anna Trykozko, we współpracy z Łukaszem Czerwińskim
