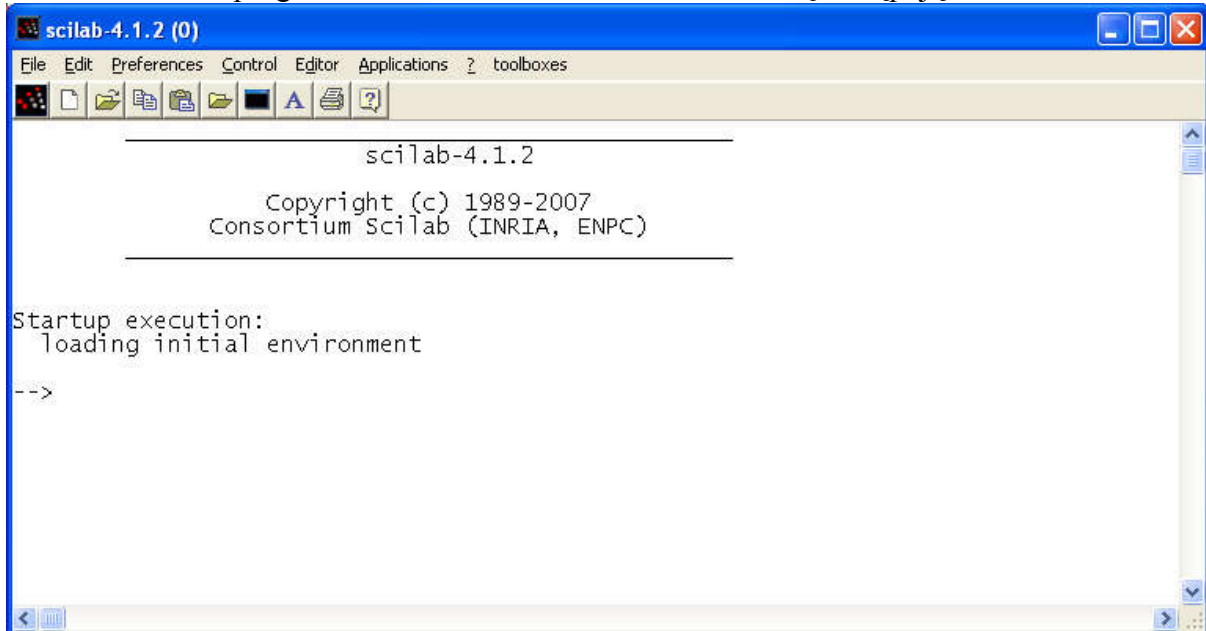


SCILAB

Krótkie wprowadzenie na potrzeby szybkiej programowej realizacji wybranych metod sztucznej inteligencji

Preferowane środowisko programistyczne to SCILAB 4.1.2

Po uruchomieniu programu SCILAB 4.1.2 na ekranie ukaze się następujące okno:



Rys. 1 – Okno główne programu SCILAB (okno konsolowe)

W oknie z Rys. 1 można wykonywać wszystkie polecenia SCILAB'a oraz wyświetlać dane zapisane w zmiennych.

Podstawowe operacje:

Definicja zmiennej X z przypisaniem wartości 5:

```
X=5;
```

Definicja macierzy X wypełnionej zerami o 5 wierszach i 2 kolumnach:

```
X=zeros(5,2);
```

Definicja macierzy X wypełnionej jedynekami o 6 wierszach i 3 kolumnach:

```
X=ones(6, 3);
```

Przypisanie do elementu w pierwszym wierszu i pierwszej kolumnie macierzy X wartości 8:

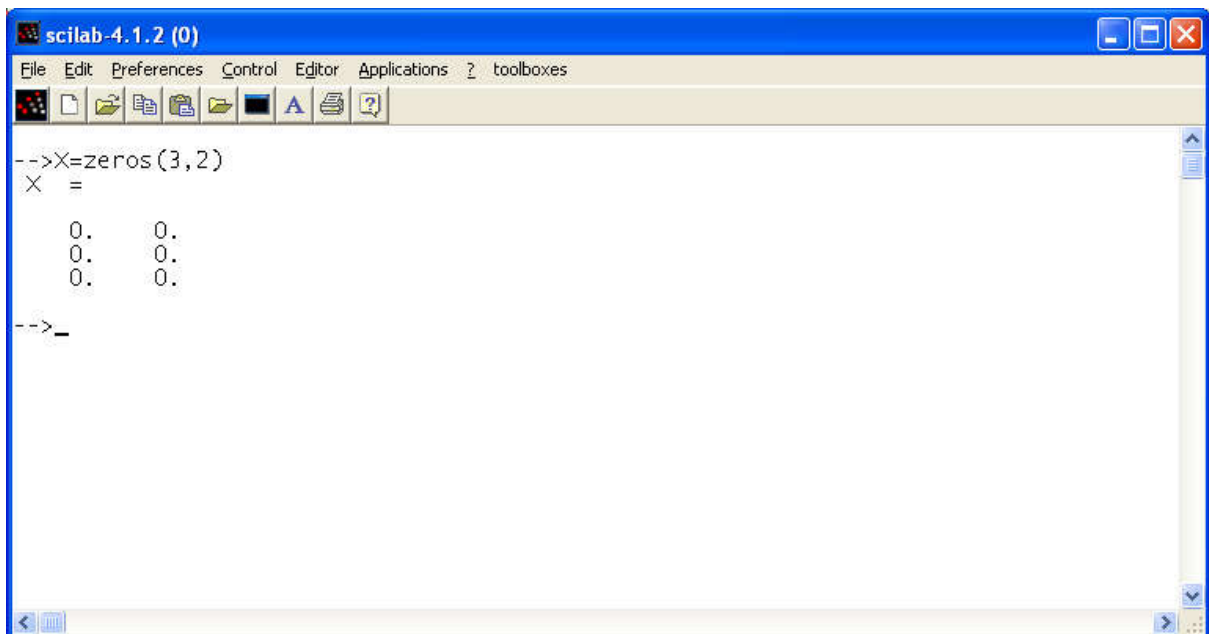
```
X(1,1)=8;
```

Pobranie wartości z drugiego wiersza i trzeciej kolumny macierzy X do zmiennej Y:

```
Y=X(2,3);
```

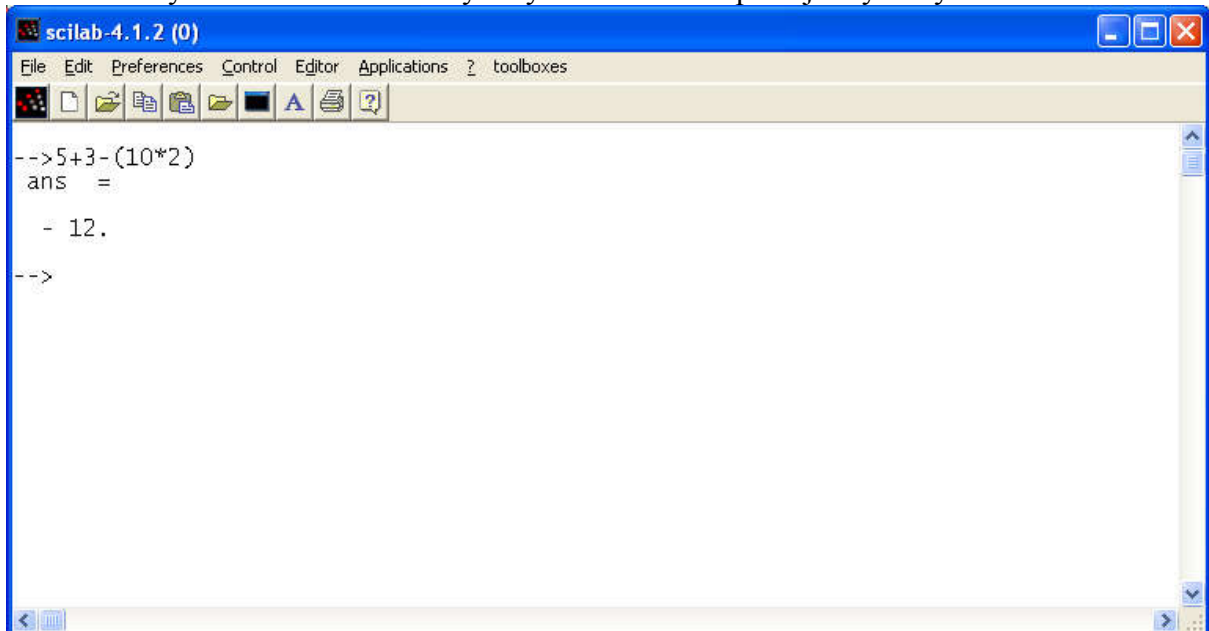
W przypadku, gdy na końcu linii nie wstawimy średnika wówczas wynik tej operacji zostanie wyświetlony na ekranie.

Dla przykładu na Rys. 2 przedstawiono utworzenie macierzy X o 3 wierszach i 2 kolumnach:



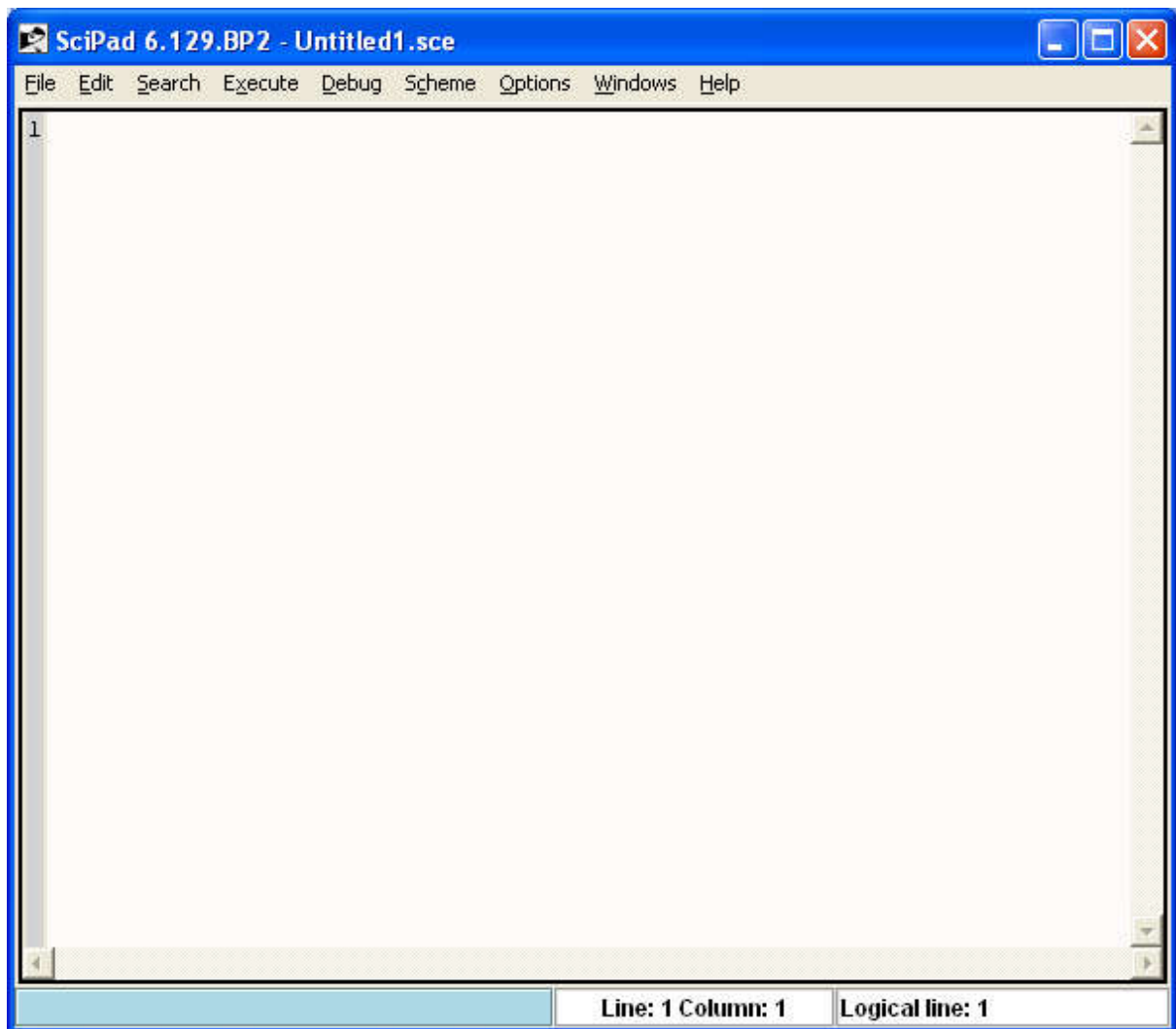
Rys. 2 – Przykład użycia instrukcji zeros

W oknie z Rys. 1 można również wykonywać wszelkie operacje arytmetyczne:



Powyżej widzimy obliczony wynik operacji arytmetycznej.

Aby tworzyć programy (tzw. Skrypty) należy wybrać opcję **Editor** z górnego Menu. Wówczas uruchomi się edytor programu SCILAB (patrz Rys.3), w którym możemy pisać kod programu.



Rys. 3 – Edytor programu SCILAB

Napiszmy teraz prosty program pobierający od użytkownika dwie liczby a następnie obliczający sumę tych liczb i wyświetlający ją na ekran (patrz Rys. 4).

```
1 //----- pobranie pierwszej liczby A
2 A=input('Podaj A:');
3 //----- pobranie drugiej liczby B
4 B=input('Podaj B:');
5 //----- dokonanie sumowania i przypisanie wyniku do zmiennej S
6 S=A+B;
7 //----- wyświetlenie wartości S na ekran
8 disp(S);
9
```

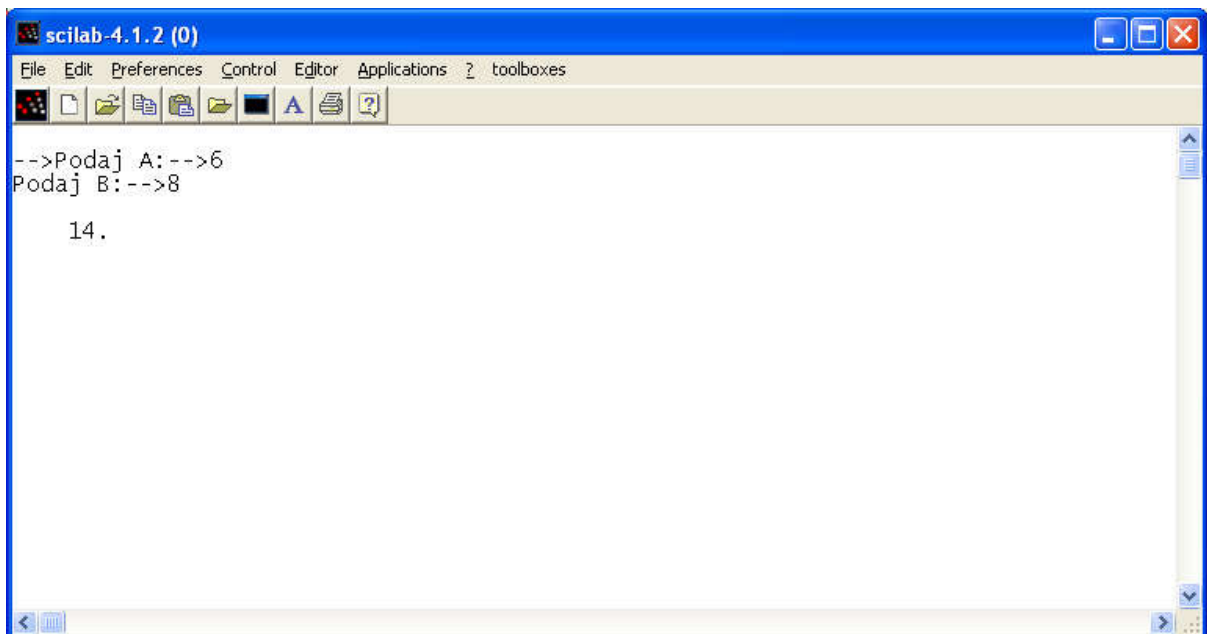
Rys. 4 – Prosty program w środowisku SCILAB

Teraz należy program zapisać na dysku. W nazwie programu należy wpisać ręcznie rozszerzenie *.sce. Aby zapisać program należy w oknie z Rys. 4 wybrać opcję:

File -> Save As

Założmy, że program zapisaaliśmy pod nazwą **suma.sce**.

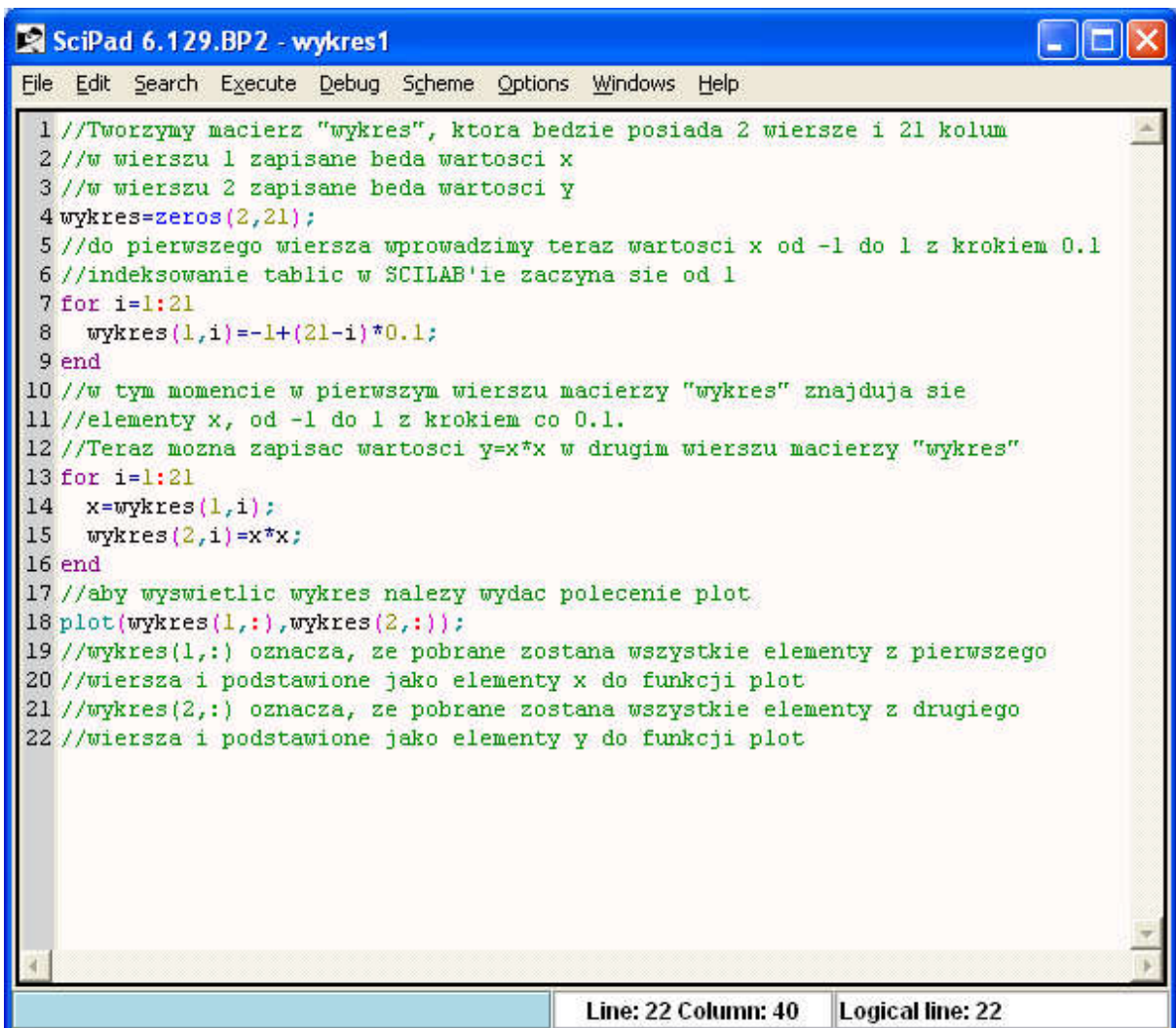
Aby program uruchomić należy wybrać z okna z Rys. 4 opcję: **Execute -> Load into Scilab**. Jeśli wszystko jest ok. program zostanie uruchomiony w oknie głównym SCILAB'a. Jeśli pojawią się błędy wówczas użytkownik zostanie o nich powiadomiony. Na Rys. 5 przedstawiono wynik uruchomienia programu z Rys. 4.



Rys. 5 – Efekt uruchomienia programu z Rys. 4

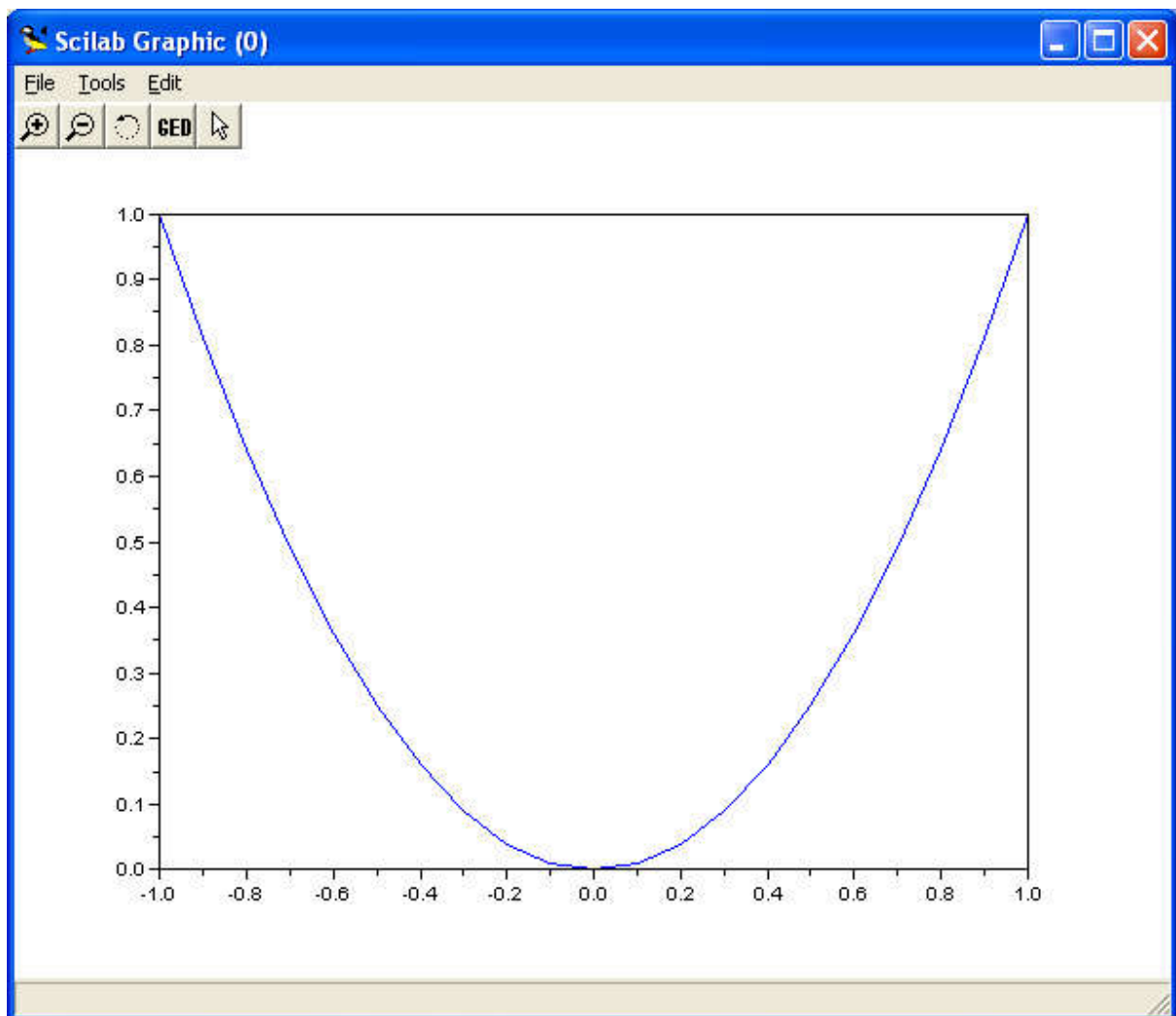
Wykreślenie wykresu 2D

Aby wykreślić dowolny wykres należy przygotować odpowiednie dane odpowiedzialne za wartości X oraz Y. Dla przykładu wykreślimy wykres funkcji $y=x^2$ w przedziale $[-1; 1]$. Przyjmijmy, że przedział $[-1; 1]$ ma mieć przyrost (dyskretyzację) co 0.1. Wówczas na osi „x” znajdzie się 21 elementów $(-1, -0.9, -0.8, \dots, 0, 0.8, 0.9, 1)$. Na Rys. 6 przedstawiono program demonstrujący wykreślenie wykresu $y=x^2$ dla x należącego $[-1; 1]$.



Rys. 6 – Wykreślanie wykresu 2D w jednym oknie

Po uruchomieniu programu z Rys. 6 otrzymamy:



Rys. 7 – Wykres funkcji $y=x*x$

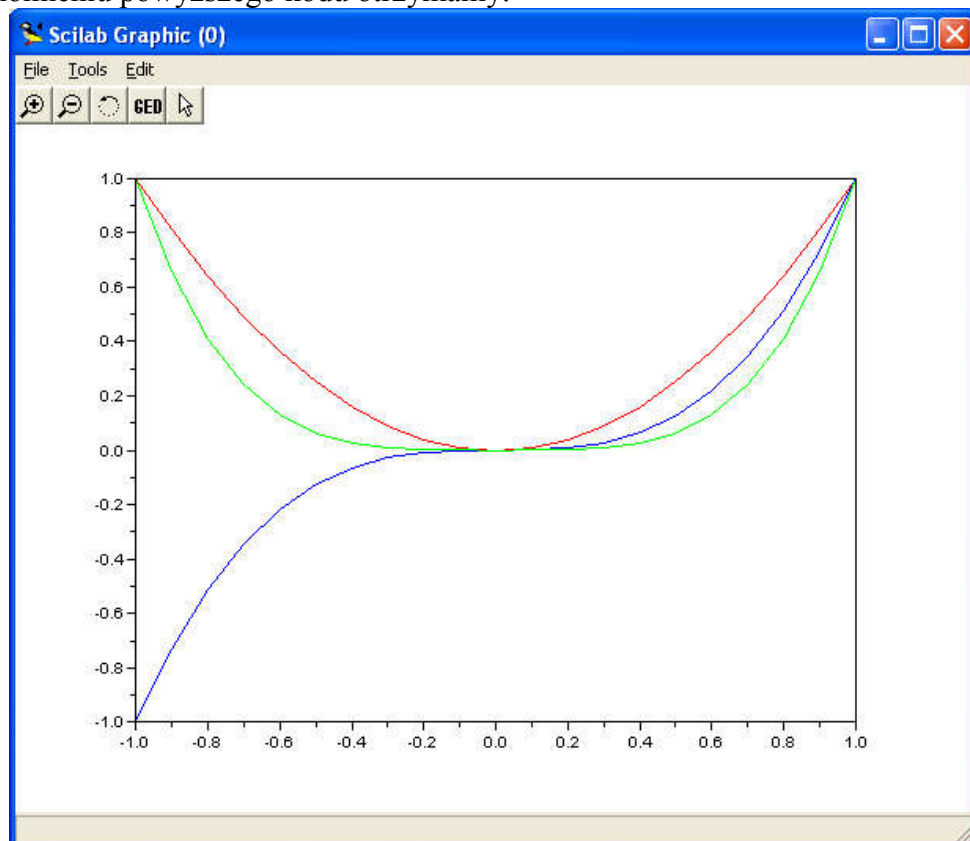
Wykreślenie kilku wykresów 2D w jednym oknie

Aby wykreślić kilka wykresów w jednym oknie należy użyć instrukcji `mtlb_hold on`; oraz `mtlb_hold off`. Teraz wykreślimy 3 wykresy: $y=x^2$; $y=x^3$; $y=x^4$ w jednym oknie. Instrukcja `mtlb_hold on` – blokuje okno; `mtlb_hold off` – odblokowuje okno;

```
SciPad 6.129.BP2 - wykres1 (modified)
File Edit Search Execute Debug Scheme Options Windows Help
1 //Tworzymy macierz "wykres", która będzie posiadała 4 wiersze i 21 kolumn
2 //w wierszu 1 zapisane będą wartości x
3 //w wierszu 2 zapisane będą wartości y=x^2;
4 //w wierszu 3 zapisane będą wartości y=x^3;
5 //w wierszu 4 zapisane będą wartości y=x^4;
6 wykres=zeros(4,21);
7 //do pierwszego wiersza wprowadzimy teraz wartości x od -1 do 1 z krokiem 0.1
8 //indeksowanie tablic w SCILAB'ie zaczyna się od 1
9 for i=1:21
10  wykres(1,i)=-1+(21-i)*0.1;
11 end
12 //w tym momencie w pierwszym wierszu macierzy "wykres" znajdują się
13 //elementy x, od -1 do 1 z krokiem co 0.1.
14 //Teraz można zapisać wartości y=x^2 w drugim wierszu macierzy "wykres"
15 //w trzecim wierszu macierzy "wykres" zapisujemy y=x^3
16 //w czwartym wierszu macierzy "wykres" zapisujemy y=x^4
17 for i=1:21
18  x=wykres(1,i);
19  wykres(2,i)=x^2;
20  wykres(3,i)=x^3;
21  wykres(4,i)=x^4;
22 end
23 //aby wyświetlić wykres y=x^2 w kolorze czerwonym należy wydać polecenie plot
24 plot(wykres(1,:),wykres(2:4,:), 'r');
25 //teraz wykres należy zamrozić przy pomocy instrukcji mtlb_hold on;
26 mtlb_hold on;
27 //a następnie narysujemy pozostałe wykresy na zamrożonym oknie
28 plot(wykres(1,:),wykres(3:4,:), 'b'); //w kolorze niebieskim
29 plot(wykres(1,:),wykres(4,:), 'g'); //w kolorze zielonym
```

Line: 29 Column: 56 Logical line: 29

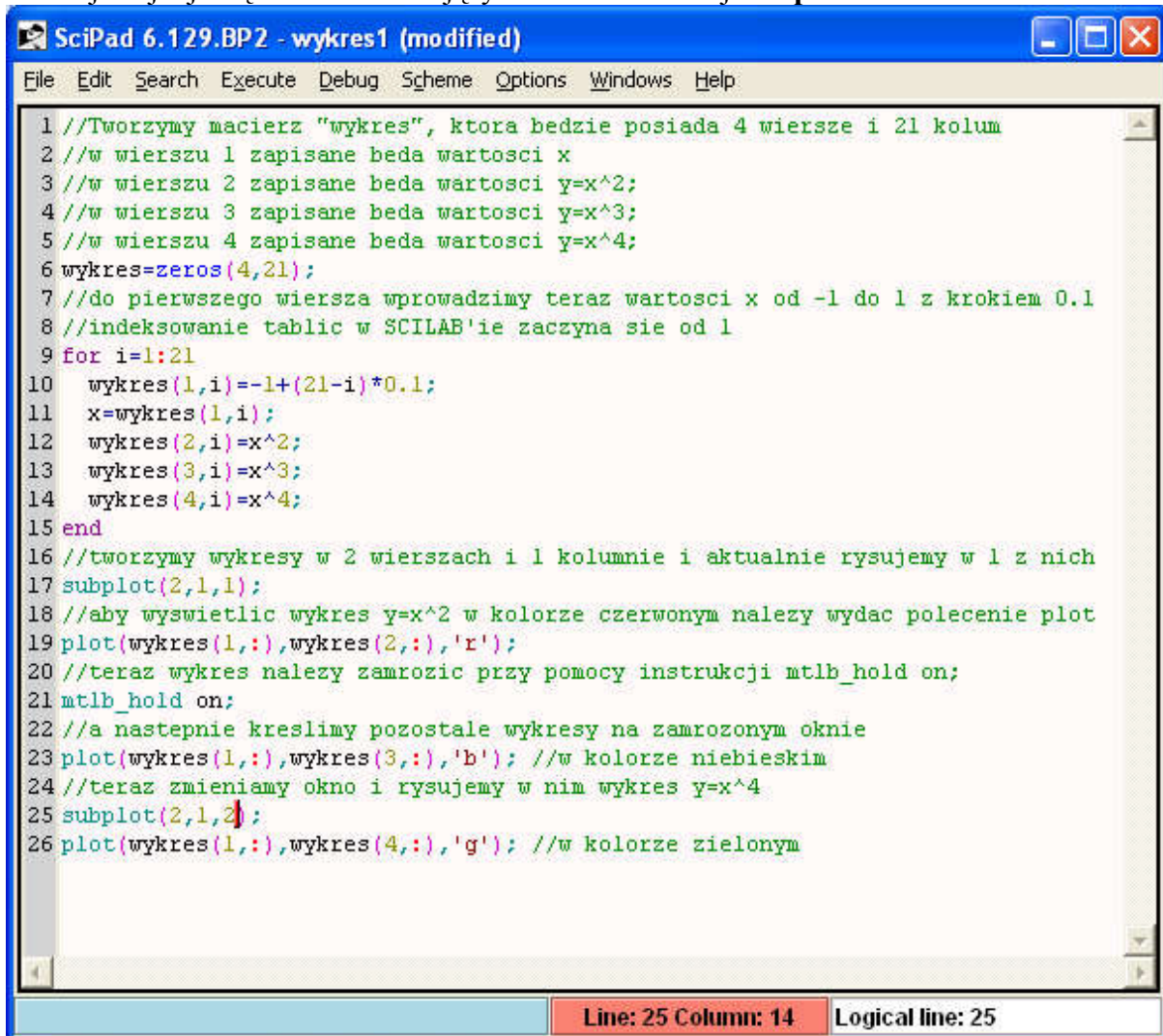
Po uruchomieniu powyższego kodu otrzymamy:



Aby wykreślić np. 2 wykresy ($y=x^2$ i $y=x^3$) w jednym oknie, a wykres $y=x^4$ w oknie drugim należy użyć polecenia **subplot(x,y,z)**;

x – oznacza liczbę wierszy z wykresami
y – oznacza liczbę kolumn z wykresami
z – oznacza numer okna w którym rysujemy.

Poniżej znajduje się kod demonstrujący działanie instrukcji **subplot**



```
1 //Tworzymy macierz "wykres", która będzie posiadała 4 wiersze i 21 kolumn
2 //w wierszu 1 zapisane będą wartości x
3 //w wierszu 2 zapisane będą wartości y=x^2;
4 //w wierszu 3 zapisane będą wartości y=x^3;
5 //w wierszu 4 zapisane będą wartości y=x^4;
6 wykres=zeros(4,21);
7 //do pierwszego wiersza wprowadzimy teraz wartości x od -1 do 1 z krokiem 0.1
8 //indeksowanie tablic w SCILAB'ie zaczyna się od 1
9 for i=1:21
10  wykres(1,i)=-1+(21-i)*0.1;
11  x=wykres(1,i);
12  wykres(2,i)=x^2;
13  wykres(3,i)=x^3;
14  wykres(4,i)=x^4;
15 end
16 //tworzymy wykresy w 2 wierszach i 1 kolumnie i aktualnie rysujemy w 1 z nich
17 subplot(2,1,1);
18 //aby wyświetlić wykres y=x^2 w kolorze czerwonym należy wydać polecenie plot
19 plot(wykres(1,:),wykres(2:3,:), 'r');
20 //teraz wykres należy zamrozić przy pomocy instrukcji mtlb_hold on;
21 mtlb_hold on;
22 //a następnie kreślimy pozostałe wykresy na zamrożonym oknie
23 plot(wykres(1,:),wykres(3:4,:), 'b'); //w kolorze niebieskim
24 //teraz zmieniamy okno i rysujemy w nim wykres y=x^4
25 subplot(2,1,2);
26 plot(wykres(1,:),wykres(4,:), 'g'); //w kolorze zielonym
```

Line: 25 Column: 14 Logical line: 25

Po uruchomieniu powyższego kodu otrzymamy:

