



**Zakład Systemów Elektronicznych i Telekomunikacyjnych**  
**METODY NUMERYCZNE**

**1 – NARZĘDZIA PROGRAMOWE CZĘŚĆ 2**

**Nazwa roku:** ET0-DU-1(01)  
**Data wykonania:** 14.04.2015  
**Grupa laboratoryjna:** L02  
**Nr zespołu:** 3

**Skład zespołu:**  
1. Krzysztof Maj  
2. Witold Olechowski  
3. Damian Pchełka

## SPIS TREŚCI

<b>1. WSTĘP</b>	<b>3</b>
<b>2. REALIZACJA ĆWICZENIA LABORATORYJNEGO</b>	<b>4</b>
2.1. Wykresy 2D	4
2.1.1 Listing kodu źródłowego	4
2.1.2 Zadanie do przeanalizowania	5
2.1.3 Listingi kodów źródłowych	6
2.2. Wykresy 3D	7
2.2.1 Listing kodu źródłowego	8
2.2.2 Zadanie do przeanalizowania	8
2.2.3 Listing kodu źródłowego	9
<b>3. PĘTLE PROGRAMOWE I INSTRUKCJE WARUNKOWE</b>	<b>10</b>
3.1. Pierwsze zadanie do rozwiązania	10
3.1.1 Listingi kodów źródłowych	10
3.2. Drugie zadanie do rozwiązania	11
3.2.1 Listing kodu źródłowego	11
<b>4. WNIOSKI</b>	<b>12</b>

## 1. WSTĘP

Celem ćwiczenia laboratoryjnego było wykonanie wizualizacji obliczeń numerycznych, za pomocą wykresów w programie **Scilab**, oraz wykorzystywania typowych konstrukcji programistycznych, które w przyszłości ułatwią sformułowanie i realizację dowolnego problemu w środowisku programu **Scilab**.

## 2. REALIZACJA ĆWICZENIA LABORATORYJNEGO

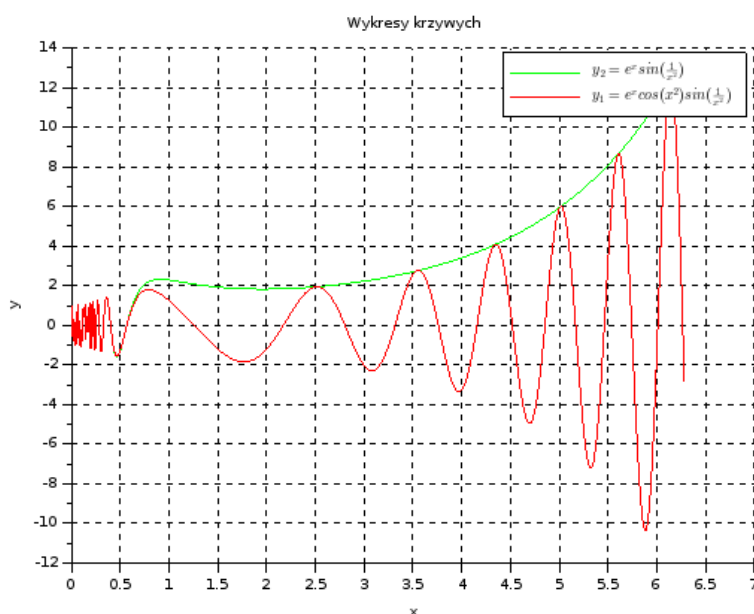
Środowisko **Scilab** umożliwia wizualizację wykonanych obliczeń numerycznych w wielu wariantach. Najpopularniejszym rozwiązaniem są wykresy dwu i trójwymiarowe realizowane za pomocą odpowiednich funkcji np. (**plot2d**, **plot3d**).

### 2.1. Wykresy 2D

Za pomocą funkcji **plot2d** zostały wykonane wykresy dwóch funkcji :

$$y_1 = e^x \cos(x^2) \sin\left(\frac{1}{x^2}\right) \quad \text{oraz} \quad y_2 = e^x \sin\left(\frac{1}{x^2}\right), \text{ dla } x \text{ z przedziału } [0, 2\pi].$$

Na Rys.1 przedstawiono wykresy wygenerowanych krzywych.



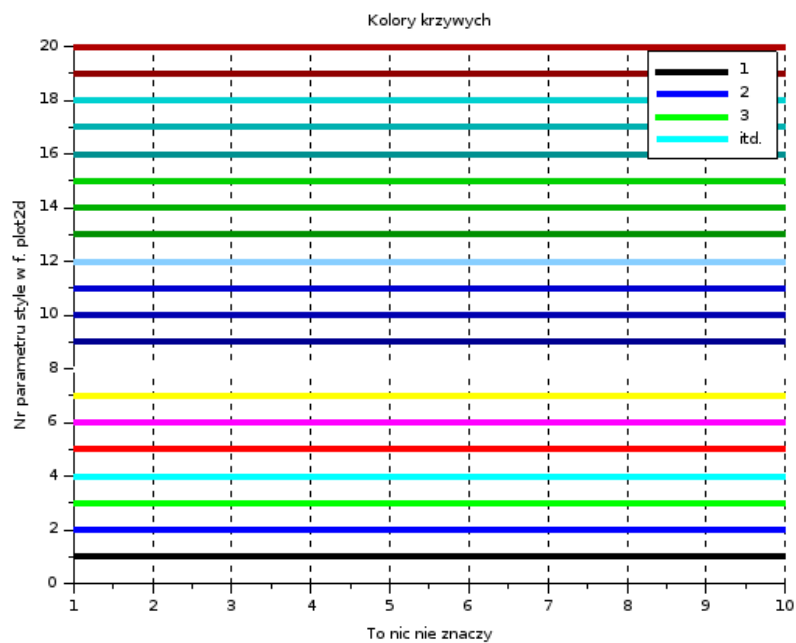
Rys. 1: Wykresy krzywych zadanych funkcji

#### 2.1.1 Listing kodu źródłowego

```
///Skrypt generujący wykres funkcji y1 i y2
x = linspace(%eps,2*%pi,1000)'; // wektor x stanowiący dziedzinę funkcji
y1 = exp(x).*cos(x.^2).*sin(x.^-2); // wektor wartości funkcji
y2 = exp(x).*sin(x.^-2);
xgrid() //siatka wykresu
xtitle('Wykresy krzywych','y','y1')
plot2d(x,y2, style=3)
plot2d(x,y1, style=5)
legend('$\textstyle y_2 = e^x \sin(\frac{1}{x^2})$', '$\textstyle y_1 = e^x \cos(x^2) \sin(\frac{1}{x^2})$') //legenda do wykresu
```

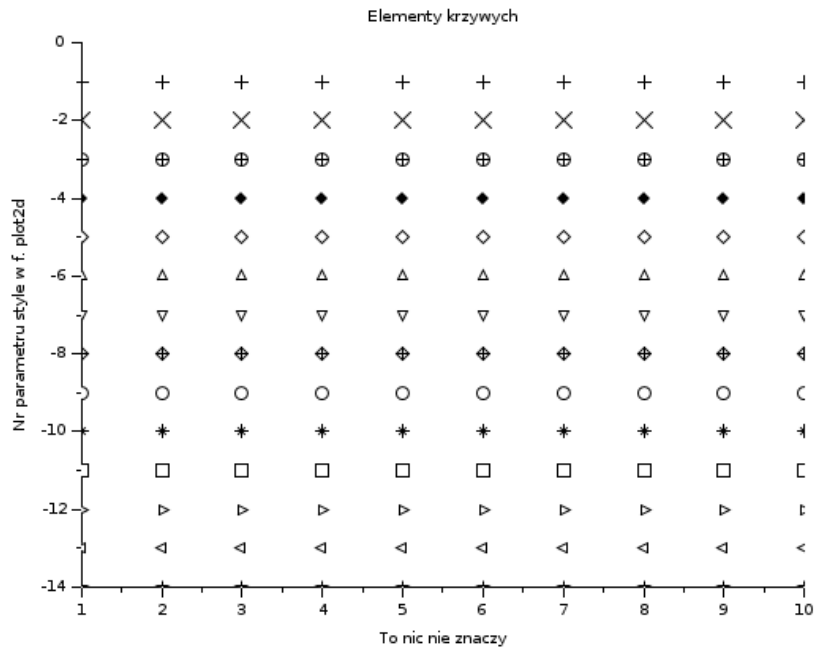
### 2.1.2 Zadanie do przeanalizowania

Wykorzystywana oprogramowanie umożliwia formatowanie wykresów w bardzo szerokim zakresie. Na Rys.2 przedstawiono kilka wariantów edytowania wykresów.



Rys. 2: Wynik realizacji skryptu

Na Rys.3 przedstawiono możliwości formatowania krzywych na wykresach.



Rys. 3: Realizacja skryptu 2

### 2.1.3 Listingi kodów źródłowych

//Program obrazujący kolorystykę krzywych na wykresach

```
nr=[1:20];
```

```
x=[1;10];
```

```
y=[1;1];
```

```
xset('thickness',4) //grubość linii = 4 pixele
```

```
plot2d(x,y*nr,style=nr)
```

```
xtitle('Kolory krzywych','To nic nie znaczy','Nr parametru style w f. plot2d')
```

```
xset('thickness',1)
```

```
xgrid() //siatka wykresu
```

```
legend('1','2','3','itd.') //legenda do wykresu
```

//Program obrazujący elementy krzywych na wykresach

```
nr=[1:14]; //UWAGA - istotny jest znak minus dla parametru style w f. plot2d
```

```
x=[1:10]';
```

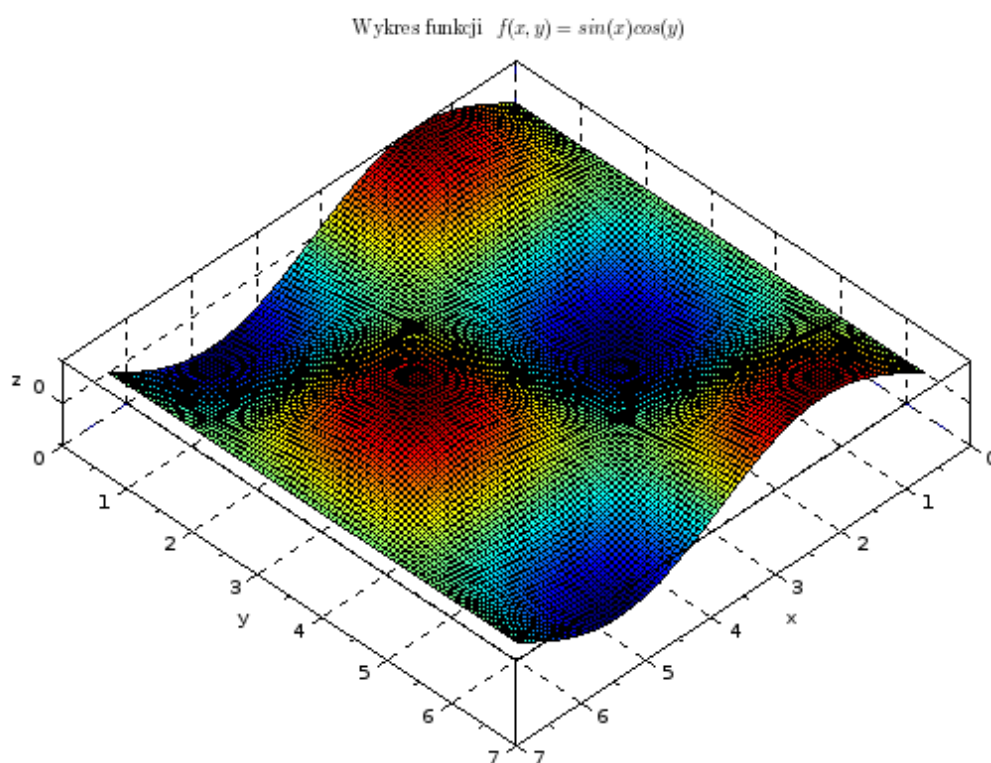
```
y=ones(x);
```

```
plot2d(x,y*nr,style=nr)
```

```
xtitle('Elementy krzywych','To nic nie znaczy','Nr parametru style w f. plot2d')
```

## 2.2. Wykresy 3D

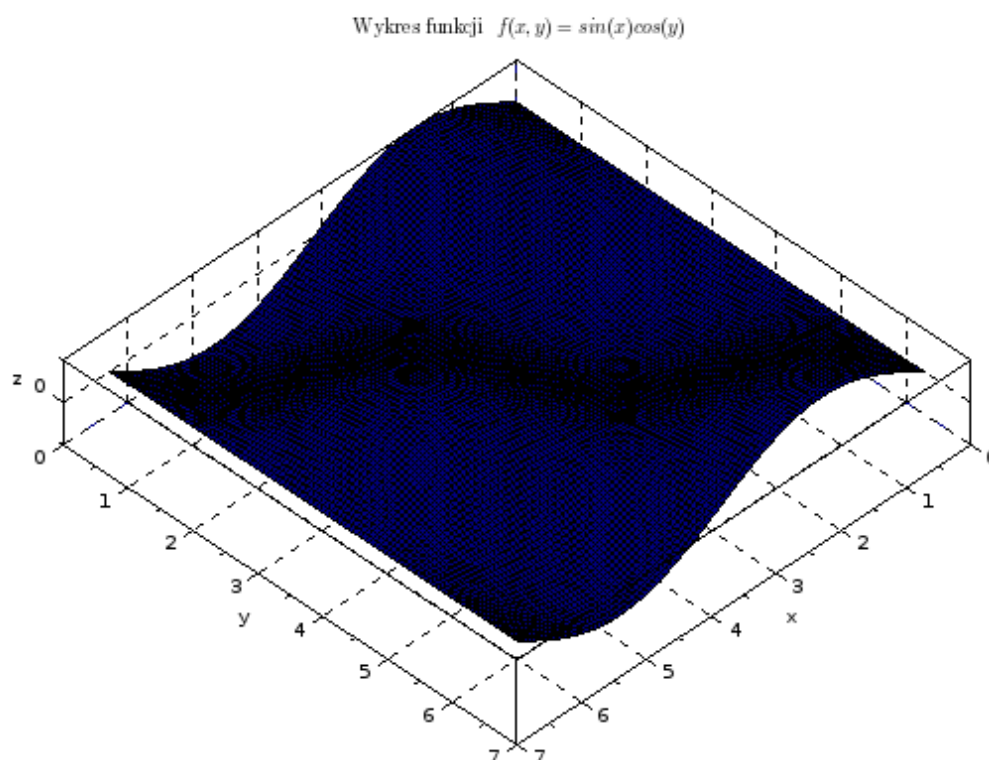
Korzystając z podstawowej funkcji do generowania wykresów trójwymiarowych **plot3d**, oraz **plot3d1**, wykreślono i sformatowano wykres funkcji:  $f(x, y) = \sin(x) \cdot \cos(y)$ , dla  $x$  i  $y$  z przedziału  $[0, 2\pi]$ .



Rys. 4: Wizualizacja przebiegu funkcji dwóch zmiennych za pomocą komendy **plot3d1**

W przypadku zastosowania **plot3d1** zmiany wartości wykreślanej funkcji są dodatkowo reprezentowane przez gradient barwy rysowanej powierzchni. Poprawia to w sposób znaczący czytelność wykresu.

W celu porównania działania komend służących do rysowania wykresów trójwymiarowych przebieg tej samej funkcji wykreślono za pomocą komendy **plot3d**.



Rys. 5: Wizualizacja przebiegu funkcji dwóch zmiennych za pomocą komendy **plot3d**

### 2.2.1 Listing kodu źródłowego

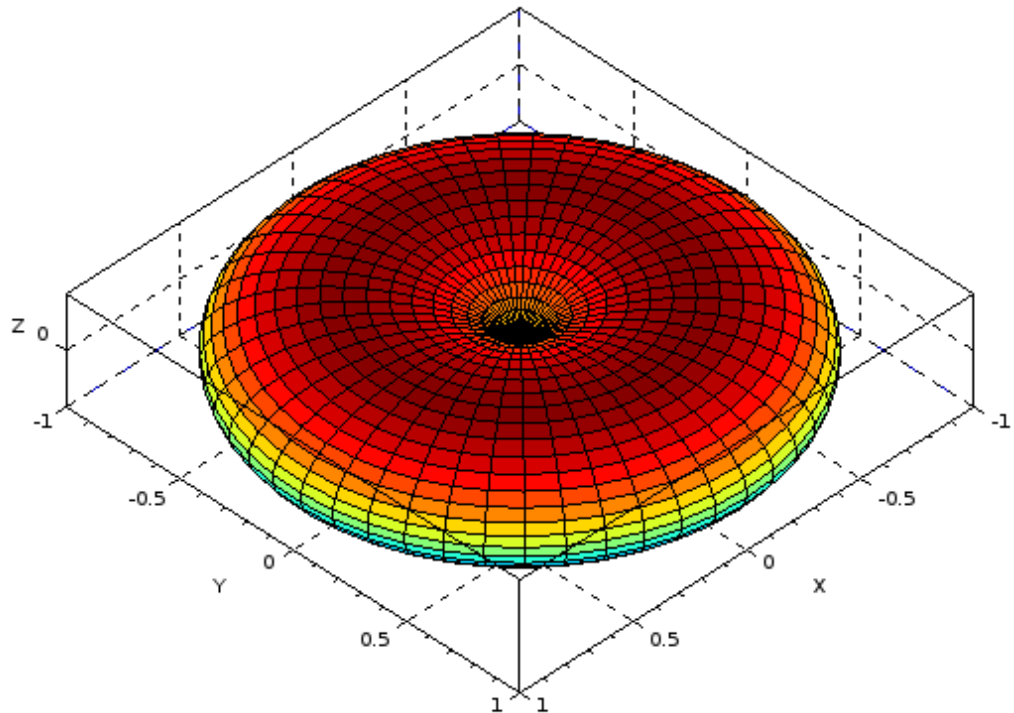
```
//Program obrazujący wykresy funkcji  $f(x,y)=\sin(x)\cos(y)$ 
x = linspace(%eps,2*%pi,100);
y = linspace(%eps,2*%pi,100);
z = sin(x)*cos(y);
xtitle('$\textstyle \mbox{Wykres funkcji } f(x,y)=\sin(x)\cos(y)$','x','y','z')
plot3d1(x,y,z) //plot3d(x,y,z) dla drugiego wykresu
xset('colormap',jetcolormap(1024))
xgrid
```

### 2.2.2 Zadanie do przeanalizowania

Korzystając z funkcji do generowania wykresów trójwymiarowych **plot3d1**, wykreślono i sformatowano wykres funkcji, który w układzie współrzędnych sferycznych prezentuje unormowaną charakterystykę mocy dla elementarnego źródła promieniowania, jakim jest dipol Herza:  $P_n(\theta) = \sin^2(\theta)$ , dla  $\theta$  z przedziału  $[0, \pi]$  oraz  $\Phi$  z przedziału  $[0, 2\pi]$ .



Na Rys.6 przedstawiono wizualizację danej funkcji.



Rys. 6: Wykres funkcji  $P_n(\theta) = \sin^2(\theta)$ , dla  $\theta$  z przedziału  $[0, \pi]$  oraz  $\Phi$  z przedziału  $[0, 2\pi]$

### 2.2.3 Listing kodu źródłowego

```
theta=linspace(0,%pi,50)';
phi=linspace(0,2*%pi,50)';
//definicja funkcji, reprezentującej charakterystykę mocy w ukł. wsp. sferycznych,
// i jednoczesna zamiana współrzędnych sferycznych na kartezjańskie
deff("[x,y,z]=scp(theta,phi)","x=(sin(theta)^2).*sin(theta).*sin(phi);..
"y=(sin(theta)^2).*sin(theta).*cos(phi);..
"z=(sin(theta)^2).*cos(theta)"]
//funkcja ułatwiająca konstrukcję powierzchni wykresu, opisanej (w przykładzie)
// definicją zmieniającą współrzędne układu sferycznego na kartezjański
[Xf,Yf,Zf]=eval3dp(scp,theta,phi);
//ustawienie mapy kolorów na wykresie
xset("colormap",jetcolormap(64))
//można jeszcze użyć innych profili kolorów: autumncolormap, bonecolormap,
// coolcolormap, coppercolormap, graycolormap, hotcolormap, hsvcolormap,
// jetcolormap, oceancolormap, pinkcolormap, rainbowcolormap,
//springcolormap, summercolormap, whitecolormap, wintercolormap
plot3d1(Xf,Yf,Zf)
xgrid()
```

## 3. Pętle programowe i instrukcje warunkowe

### 3.1. Pierwsze zadanie do rozwiązania

Wykorzystując operacje na macierzach, napisać 4 przykładowe skrypty demonstrujące działanie pętli programowych: **for** i **while** oraz instrukcji warunkowych: **if** oraz **select case**.

#### 3.1.1 Listingi kodów źródłowych

*//---przykład zastosowania instrukcji select-case-----*

```
x = 1:1:3;
for i = 1:1:3
    a = x(i);
    select a
        case 1
            disp("opcja 1");
        case 2
            disp("opcja 2");
        case 3
            disp("opcja 3");
    end
end
```

*//-----przykład zastosowania petli for-----*

```
E = zeros(100)

    for i = 1:1:10
        for j = 1:1:10
            E(i,j) = i*j;
        end
    end
```

*//---przykład zastosowania instrukcji if---*

```
A = zeros(11);
B = zeros(11);
C = zeros(11);

for i = 1:1:11
    if(i <= 4)
        A(i) = i;
    else
        B(i) = i;
    end
end
```

*//---przykład na instrukcje while---*

```
i = 1;
x = 0:1:15;
//D = zeros(11);
while(x(i) < 11)
    disp("x(i)");
    i = i + 1;
end
```

## 3.2. Drugie zadanie do rozwiązania

Napisać skrypt, w którym należy:

- utworzyć macierz elementów losowych o rozmiarze 5x5 i wartościach z przedziału [0,10);
- wyliczyć liczbę elementów: a) większych od 5; b) mniejszych lub równych 1;
- wyliczyć procent, jaki stanowią elementy: a) większe od 5; b) mniejszych lub równych

W Tabeli 1. przedstawiono przykładową wygenerowaną macierz elementów losowych.

Tabela 1: Macierz elementów losowych

1,76	6,90	2,74	8,43	6,84
5,52	2,07	3,47	9,84	8,77
2,43	6,73	9,66	9,51	1,44
2,17	2,86	7,92	4,95	9,97
6,05	4,52	8,89	3,13	3,02

**Przykładowy komunikat zwrotny wyświetlony w konsoli:**

*Liczb wiekszych od 5 jest: 13,000000, mniejszych badz rownych 1 jest:0,000000.*

*Liczyby wieksze od 5 stanowaia 52,000000 a mniejsze badz rowne 1 stanowia: 0,000000*

### 3.2.1 Listing kodu źródłowego

*//--przykład z macierzą liczb losowych---*

```
a= (10-%eps)*rand(5,5);
b = 0;
c=0;
d=0;
e=0;
for i=1:size(a,1)
    if a(i)< 1 | a(i) == 1 then
        b=b+1;
    elseif a(i) > 5 then
        c=c+1;
    end
end

d=(b/size(a,1))*100;
e=(c/size(a,1))*100;
```

```
printf("Liczb wiekszych od 5 jest: %f, mniejszych badz rownych 1 jest:%f.\nLiczyby wieksze
od 5 stanowaia %f a mniejsze badz rowne 1 stanowia: %f ",c,b,e,d)
```

## 4.Wnioski

Dzięki narzędziu Scilab mamy możliwość generowania wykresów dwu i trójwymiarowych. Program umożliwia zmianę stylów wykresów, prawie w nieograniczonym stopniu. Przy generowaniu wykresów należy zwrócić uwagę na poprawność operacji macierzowych, oraz skalarnych.

Dostępne w środowisku pętle i instrukcje umożliwiają realizację zadań w bardzo prosty sposób. Składnia programu jest podobna do składni języka C, dzięki czemu jest intuicyjna dla przeciętnego użytkownika.