

Piotr Chojnacki:

MATLAB

zajęcia 2:

{Poniżej znajduje się zrzut ekranowy pola gdzie wpisuje się komendy. Nie wszystkie komendy są opisane. Wynik działania każdej z nich widać linijkę niżej. Znajdziesz tutaj podstawowe komendy graficzne. Rysowanie wykresów oraz grafika trójwymiarowa z przykładami. Wykresy poziomicowe. Komendy należy wpisywać w Workspace i potwierdzając enterem.}

zajecia3.mat

W =

1 2 5 7 8 5 -2 -9 6

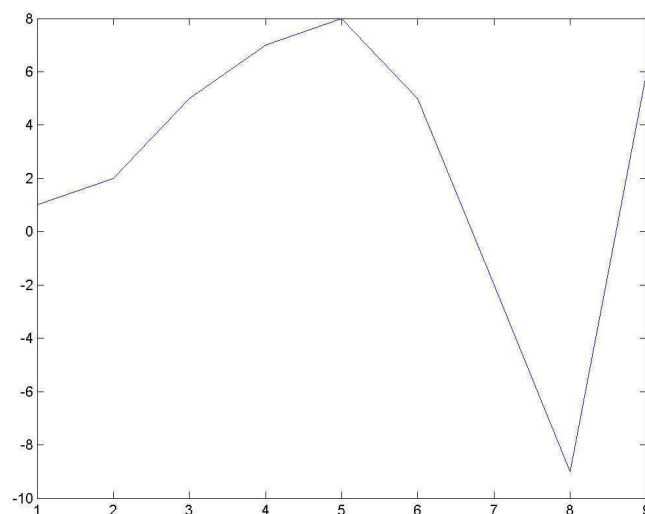
>> E=[5,7,0,-5,-10,4,3,5,1]

E =

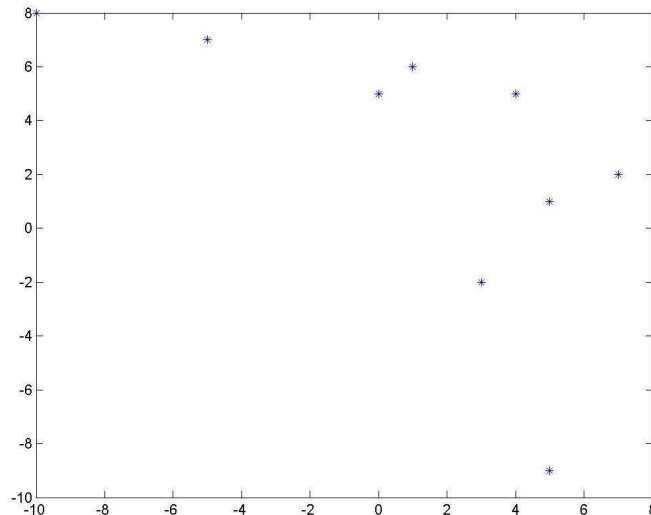
5 7 0 -5 -10 4 3 5 1

>> **plot(E)** {komenda rysująca wykres składający się z punktów E, wykres zostaje otworzony w nowym okienku Figure}

>> **plot(W)** {komenda rysująca wykres w postaci linii W w tym samym okienku. Poniżej zamieszczony jest wykres W}



```
>> plot(W)
>> hold on {komenda powodująca, że wszystkie następne komendy będą dodawane do tego samego okienka}
>> plot(W,E, '*') {na wykresie zostaną zaznaczone punkty w postaci gwiazdek dwóch wektorów W i E. Mogą to być również inne znaki np. +,x,o itp.}
```



```
>> plot(W, '*')
>> hold off
>> hold on
>> plot(W, '*')
>> plot(W)
>> hold on
>> plot(W, '*')
```

```
>> plot(W)
```

```
>> plot(E,W) {naniesienie dwóch wykresów na jeden rysunek. Trzeba pamiętać, że wektory muszą być jednakowej długości}
```

```
>>
>> plot(E,W)
```

```
>> grid on {komenda włączająca siatkę na rysunku. Standardowo jest ona w kolorze białym, więc musimy zmienić tło rysunku, żeby ją zobaczyć. Wszystkie parametry rysunku takie jak kolory tła, linii, tytuł wykresu itp. Można zmienić klikając (w okienku rysunku) kolejno: edit > axes properties, lub w menu insert możemy kolejno dodawać tytuły, legendę lub też tekst na wykresie.}
```

```
>> text(3,4,'ererer') {dodanie z klawiatury tekstu na wykres, gdzie liczby oznaczają kolejno x i y na wykresie, czyli położenie}
```

```
>> gtext('ererer') {tekst umieszczany za pomocą myszki}
```

```
>> save zajecia3
```

```
>> hold on
```

```
>> line([2,4],[7,8]) {komenda rysująca linie na wykresie. Musimy podać położenie początkowe x1 i y1 jak i końcowe x2 i y2}
```

```
>> line([-2,0],[7,8])
>> line([-8,-8],[6,6])
>> figure {komenda otwierająca kolejne okienko Figure}
>> figure
>> fplot('sin(x)',[-10,10]) {komenda rysująca funkcję sin w przedziale od -10 do 10}
>> figure
>> fplot('cos(x)',[-10,10])
```

{poniżej znajduje się zestaw komend rysujących różne funkcje. Są to tylko przykłady, co można zrobić}

```
>> fplot('sin(x)*cos(2*x)',[-10,10])
>> fplot('sin(x)*cos(2*x)*sin(0.5*x)',[-10,10])
>> fplot('sin(x)*cos(2*x)*sin(0.5*x)*cos(0.2*4)',[-10,10])
>> fplot('sin(x-2)*cos(2*x)*sin(0.5*x)*cos(0.2*4)',[-10,10])
>> fplot('sin(x-2)*cos(2*x)*sin(0.5*x)*(cos(0.2*4)^2)',[-10,10])
>> fplot('sin(x+22)*cos(2*x)*sin(0.5*x)*(cos(0.2*4)^2)',[-10,10])
>> fplot('sin(x+22)*cos(2*x)*sin(0.5*x)*(cos(0.2*4)^0.5)',[-10,10])
>> save zajecia3
>> subplot(2,2,2) {komanda dzieląca okienko na cztery małe wykresy. Kolejno dwa wiersze, dwie kolumny. Natomiast trzecia oznacza rysunek, który aktualnie edytujemy.
```

```
>> subplot(2,2,1)
>> fun=inline('sin(x)')
```

fun =

Inline function:
fun(x) = sin(x)

```
>> fplot('fun(x)',[-2,2])
>> figure
>> fplot('fun(x)',[-2,2])
>> fplot('fun(x)',[-2,2])
>> save zajecia3
>> fplot('fun(x)',[-2,5])
>> figure
>> fplot('fun(x)',[-2,5])
>> figure
```

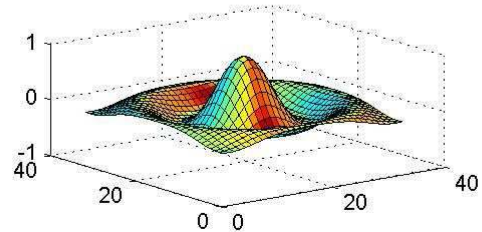
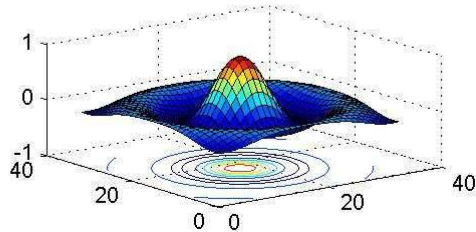
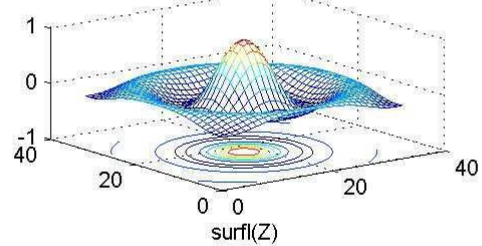
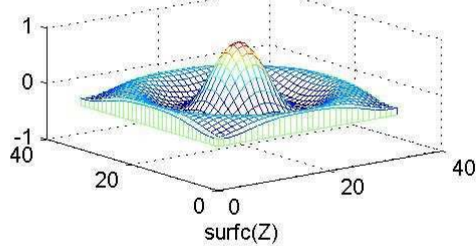
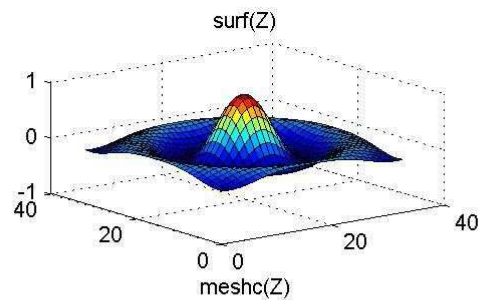
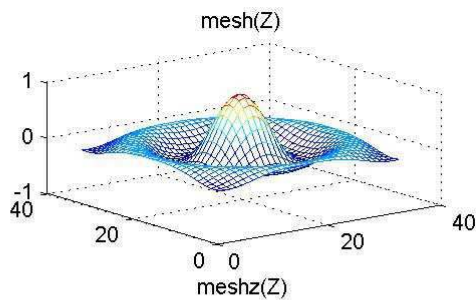
{grafika trójwymiarowa z przykładami wykresów}

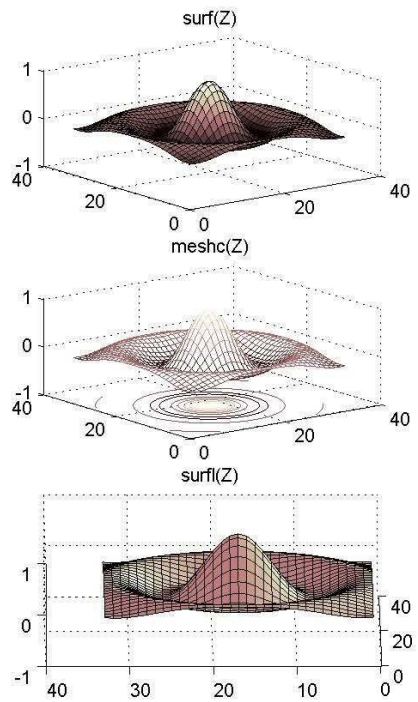
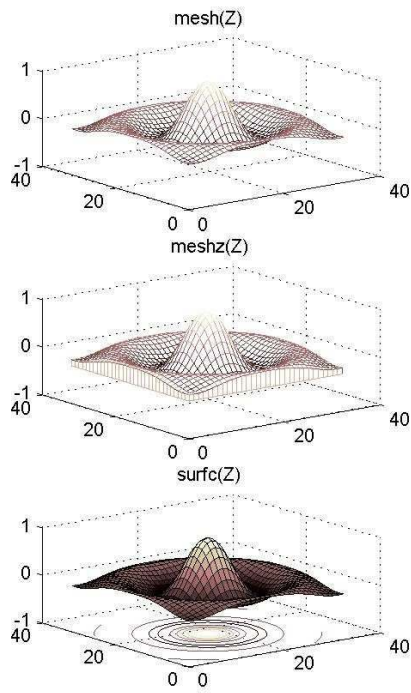
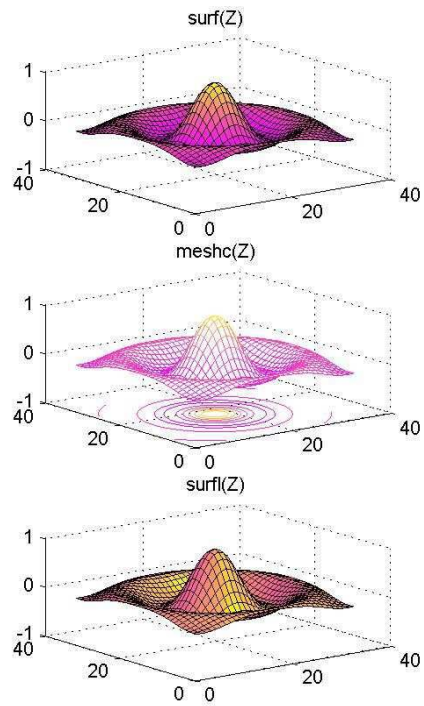
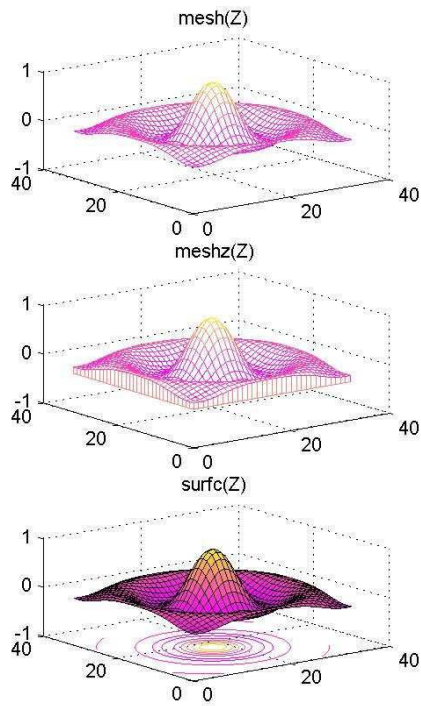
{po zdefiniowaniu:}

```
>> [W E]=meshgrid(-8:0.5:8);
>> R=sqrt(W.^2+E.^2)+eps;
>> Z=sin(R)./R;
```

{żeby zobaczyć wynik, każdej z pogrubionych komend najlepiej zobaczyć na własne oczy, co się dzieje. Pod komendami zamieszczone są trzy zdjęcia wyników. W przypadku tych wykresów również można się pobawić kolorami. Zamieściłem również dwa przykłady}

```
>> mesh(Z)
>> subplot(3,2,2)
>> surf(Z)
>> subplot(3,2,3)
>> meshz(Z)
>> subplot(3,2,4)
>> meshc(Z)
>> subplot(3,2,5)
>> surfc(Z)
>> subplot(3,2,6)
>> surf1(Z)
>> subplot(3,3,3)
>> subplot(3,2,6)
>> subplot(3,2,2)
>> surf(Z)
```





zajecia4.mat

```
>> Z=[2,4,7,8,25,47]
```

```
Z =
```

```
2 4 7 8 25 47
```

```
>> Z=peaks {poniższe komendy rysują wykresy poziomicowe}
```

```
Z =
```

```
Columns 1 through 7
```

```
0.0001 0.0001 0.0002 0.0004 0.0007 0.0011 0.0017  
0.0001 0.0002 0.0004 0.0006 0.0010 0.0017 0.0026
```

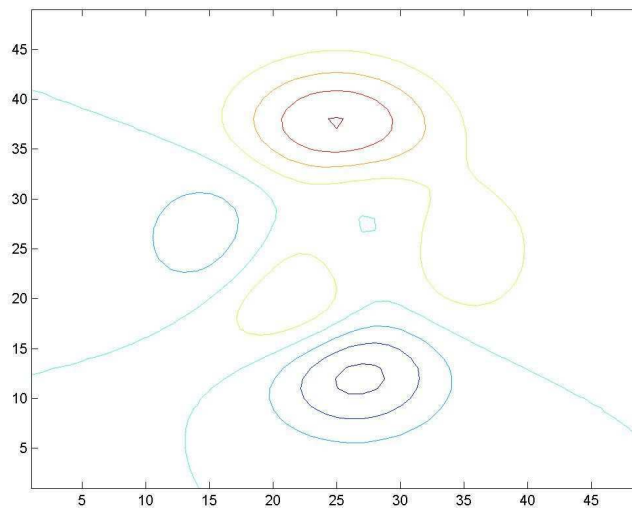
```
.....
```

{szereg danych wypisywanych przez program}

```
.....
```

```
>> contour(Z) {rysowanie konturów}
```

```
>>
```



```
[C,h]=contour(Z,10) {zaznaczenie wartości konturów na wykresie}
```

C =

Columns 1 through 7

```
-5.2174 30.0000 29.8363 29.0507 29.0000 28.0000 27.0000  
25.0000 12.0135 13.0000 14.0000 14.0354 14.4032 14.4145
```

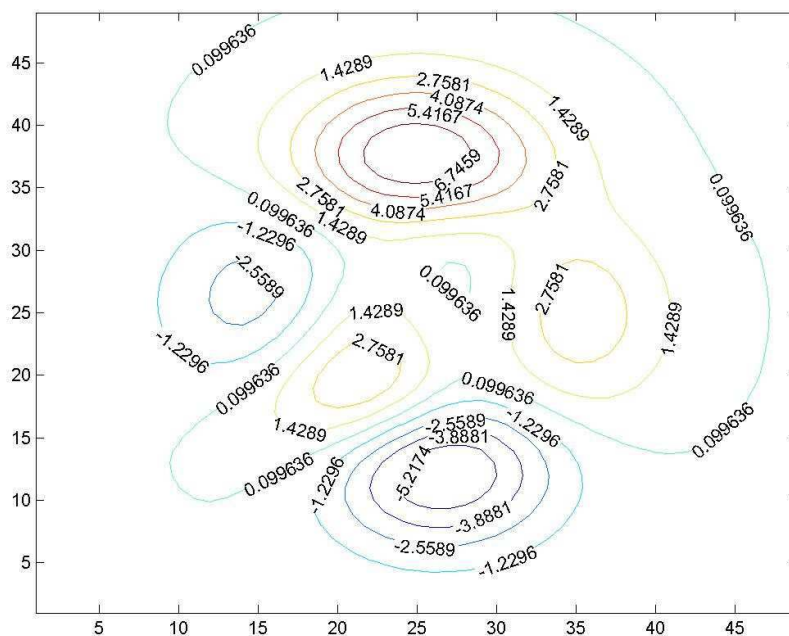
.....

{szereg danych wypisywanych przez program}

.....

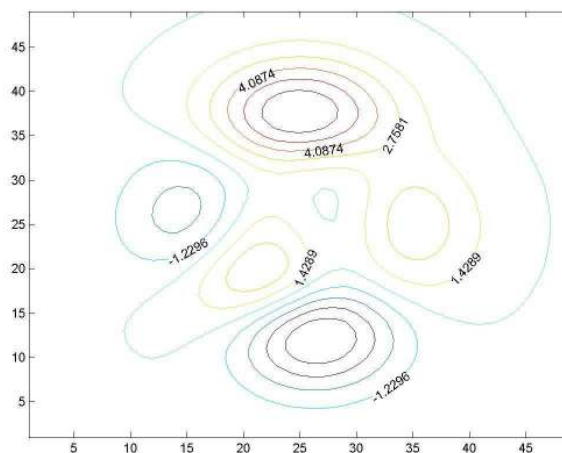
>> clabel(C,h)

>>

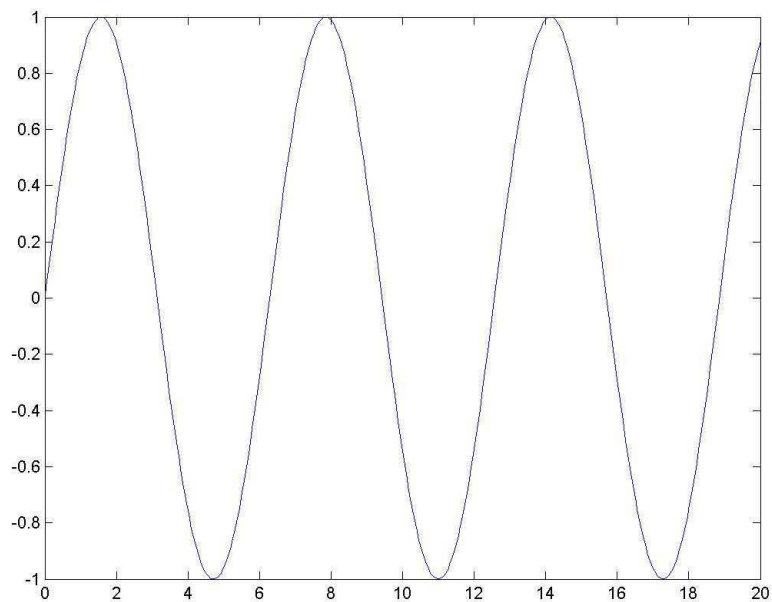


>> clabel(C,h,'manual') {komenda pozwalająca na ręczne zaznaczenie tylko wybranych konturów}

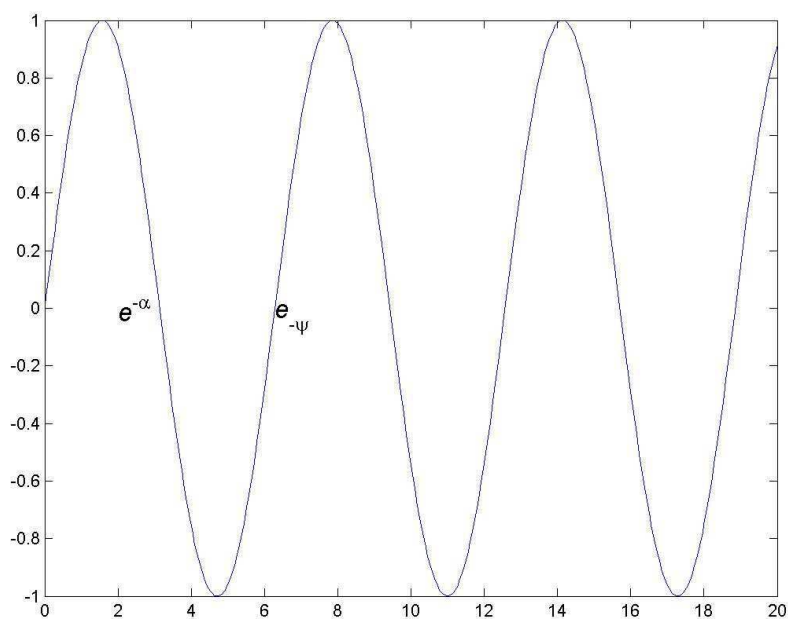
>>



```
>> fplot('sin(x)',[0 20]) {komenda rysująca wykres dowolnej funkcji}
>>
```



```
>> text(2,0,'{\ite}^{\-\alpha}') {wprowadzenie tekstu na wykres – indeks górny}
>> text(6,0,'{\ite}_{-\psi}') {wprowadzenie tekstu na wykres - indeks dolny}
>>
```




```
>> [b,v]=bucky {komendy rysujące cząsteczkę C60}
```

```
b =
```

```
(2,1)    1
```

```
.....
```

```
{szereg danych wypisywanych przez program}
```

```
.....
```

```
>>>> gplot(b,v)
```

```
>> axis('square')
```

```
>>
```

