

POLITECHNIKA CZĘSTOCHOWSKA

Wydział Inżynierii Mechanicznej i Informatyki

**Metody numeryczne**

Temat: Wprowadzenie do GNU Octave

Adrian Dobosz

Studia stacjonarne /I stopień

Kierunek: Mechatronika

## 1. Pomoc i czyszczenie terminala

1.1. Wyświetlić informacje o funkcjach sqrt i nthroot. Do czego one służą?

```
clear;  
help sqrt;  
help nthroot;
```

1.2. Wyczyścić ekran.

```
clc;
```

## 2. Proste operacje matematyczne

```
a=sqrt(32^(3/5)+0.0625*8+1/2);
```

```
a
```

```
a = 3
```

## 3. Funkcje matematyczne, potęgi i logarytmy, funkcje trygonometryczne

```
b=sqrt(32^(3/5)+sin(pi/2))
```

```
b = 3
```

## 4. Zmienne

4.1. Przyprostokątne trójkąta prostokątnego wynoszą: a=9, b=12. Obliczyć przeciwprostokątną c korzystając ze zmiennych a i b.

```
a=9
```

```
b=12
```

```
c=sqrt(a^2+b^2)
```

```
a = 9
```

```
b = 12
```

```
c = 15
```

4.2. Jeśli a i b będą długościami podstaw trapezu, a odległość między podstawami będzie siódmą częścią sumy ich długości, to jakie będzie jego pole? Pole zapisać w zmiennej Pt.

```
Pt=((a+b)/2)*(a+b)/7
```

```
Pt = 31.500
```

4.3. W zmiennej Pk zapisać pole koła, którego średnica równa się sumie a i b.

```
Pk= pi*((a+b)/2)^2
```

```
Pk = 346.36
```

## 5. Wektory i macierze

### 5.1. Utworzyć dwie macierze:

$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \end{bmatrix}$

$B = \begin{bmatrix} 1 & 4 \\ 0 & 3 \\ 2 & 2 \end{bmatrix}$

$A =$

1 2 3

4 5 0

$B =$

1 4

0 3

2 2

### 5.2. Utworzyć macierze C i D o wartościach: $C=A*B$ , $D=B*A$ . Jakie są ich rozmiary? Jak sprawdzić ich rozmiary przynajmniej na dwa sposoby?

$C=A*B$

$D=B*A$

$C =$

7 16

4 31

$D =$

17 22 3

12 15 0

10 14 6

### 5.3. Zwiększyć wartości elementów macierzy D dwukrotnie.

$D=D*2$

$D =$

34 44 6

24 30 0

20 28 12

### 5.4. Utworzyć macierz E o rozmiarze 3x4 zawierającą same zera. Wartości pierwszych trzech kolumn macierzy E przypisać z macierzy D.

$E=\text{zeros}(3,4)$

$D=E(:,1:3)$

$E =$

0 0 0 0

0 0 0 0

0 0 0 0

```
D =
    0    0    0
    0    0    0
    0    0    0
```

**5.5. Utworzyć macierz F, której wartości będą iloczynami poszczególnych wartości transponowanej macierzy A i macierzy B (nie mnożymy macierzy, ale ich poszczególne wartości). Jakie rozmiary będzie miała macierz F?**

**F=A'.\*B**

```
F =
    1    16
    0    15
    6     0
```

## Wektory

**5.6. Utworzyć wektor u zawierający wartości: u=[0 pi/2 pi 3 2 pi 2pi]**

**u=[0 pi/2 pi pi\*3/2 2\*pi]**

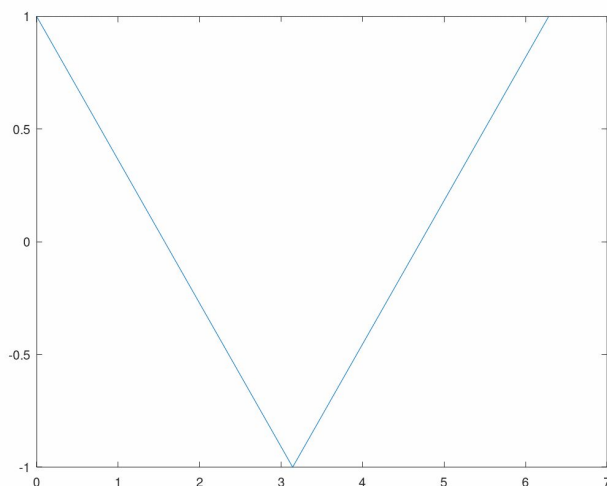
```
u =
    0.00000    1.57080    3.14159    4.71239    6.28319
```

**5.7. Utworzyć wektor v zawierający wartości cos(u). Narysować na osi współrzędnych punkty złożone z par (ui,vi) dla utworzonych wektorów. Co zdaje się przypominać wykres? Jak narysować bardziej wierny wykres funkcji cosinus?**

**v=[cos(u(1)) cos(u(2)) cos(u(3)) cos(u(4)) cos(u(5))]**

**plot(u,v)**

```
v =
    1.0000e+00    6.1230e-17   -1.0000e+00   -1.8369e-16    1.0000e+00
```



**5.8. Utworzyć wektor x zawierający wartości od 0 do  $2\pi$  z krokiem co 0.2.**

**Odczytać rozmiar wektora.**

```
x=[0:0.2:2*pi]
```

```
xsize=size(x)
```

```
x =
```

*Columns 1 through 11:*

```
0.00000 0.20000 0.40000 0.60000 0.80000 1.00000 1.20000 1.40000 1.60000  
1.80000 2.00000
```

*Columns 12 through 22:*

```
2.20000 2.40000 2.60000 2.80000 3.00000 3.20000 3.40000 3.60000 3.80000  
4.00000 4.20000
```

*Columns 23 through 32:*

```
4.40000 4.60000 4.80000 5.00000 5.20000 5.40000 5.60000 5.80000 6.00000  
6.20000
```

```
xsize =
```

```
1 32
```

**5.9. Utworzyć wektor y zawierający elementy równe  $\sin(x)$ . Narysować wykres funkcji  $\sin(x)$ .**

```
y=sin(x(1:xsize(2)))
```

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

```
y =
```

*Columns 1 through 11:*

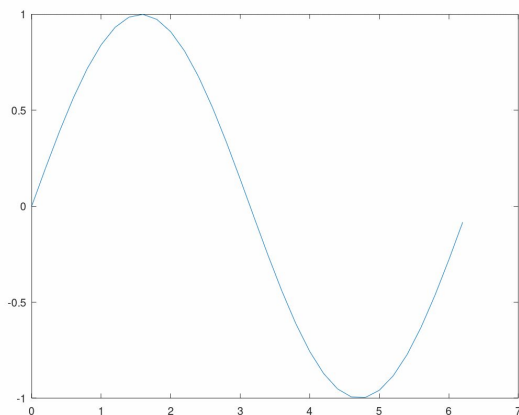
```
0.00000 0.19867 0.38942 0.56464 0.71736 0.84147 0.93204 0.98545 0.99957  
0.97385 0.90930
```

*Columns 12 through 22:*

```
0.80850 0.67546 0.51550 0.33499 0.14112 -0.05837 -0.25554 -0.44252 -0.61186  
-0.75680 -0.87158
```

*Columns 23 through 32:*

```
-0.95160 -0.99369 -0.99616 -0.95892 -0.88345 -0.77276 -0.63127 -0.46460 -0.27942  
-0.08309
```



**5.10. Utworzyć macierz BIG będącą wynikiem pomnożenia transponowanego wektora y przez jego samego. Wynik powinien być macierzą kwadratową. Można narysować: surf(BIG)**

**BIG=y'\*y**

**surf(BIG)**

**BIG =**

*Columns 1 through 11:*

```

0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000
0.00000 0.03947 0.07737 0.11218 0.14252 0.16717 0.18517 0.19578 0.19858
0.19347 0.18065
0.00000 0.07737 0.15165 0.21988 0.27935 0.32768 0.36295 0.38375 0.38925
0.37923 0.35410
0.00000 0.11218 0.21988 0.31882 0.40505 0.47513 0.52627 0.55643 0.56440
0.54988 0.51343
0.00000 0.14252 0.27935 0.40505 0.51460 0.60363 0.66860 0.70692 0.71705
0.69860 0.65229
0.00000 0.16717 0.32768 0.47513 0.60363 0.70807 0.78428 0.82923 0.84111
0.81946 0.76515
0.00000 0.18517 0.36295 0.52627 0.66860 0.78428 0.86870 0.91848 0.93164
0.90766 0.84750
0.00000 0.19578 0.38375 0.55643 0.70692 0.82923 0.91848 0.97111 0.98503
0.95968 0.89607
0.00000 0.19858 0.38925 0.56440 0.71705 0.84111 0.93164 0.98503 0.99915
0.97343 0.90891
0.00000 0.19347 0.37923 0.54988 0.69860 0.81946 0.90766 0.95968 0.97343
0.94838 0.88552
0.00000 0.18065 0.35410 0.51343 0.65229 0.76515 0.84750 0.89607 0.90891
0.88552 0.82682
0.00000 0.16062 0.31484 0.45651 0.57998 0.68033 0.75355 0.79673 0.80815
0.78735 0.73516
0.00000 0.13419 0.26304 0.38140 0.48455 0.56838 0.62956 0.66564 0.67518
0.65780 0.61420

```

0.00000	0.10241	0.20075	0.29107	0.36980	0.43378	0.48047	0.50800	0.51528
0.50202	0.46874							
0.00000	0.06655	0.13045	0.18915	0.24031	0.28188	0.31222	0.33011	0.33485
0.32623	0.30460							
0.00000	0.02804	0.05495	0.07968	0.10123	0.11875	0.13153	0.13907	0.14106
0.13743	0.12832							
0.00000	-0.01160	-0.02273	-0.03296	-0.04188	-0.04912	-0.05441	-0.05752	-0.05835
-0.05685	-0.05308							
0.00000	-0.05077	-0.09951	-0.14429	-0.18331	-0.21503	-0.23817	-0.25182	-0.25543
-0.24886	-0.23236							
0.00000	-0.08792	-0.17233	-0.24987	-0.31744	-0.37237	-0.41245	-0.43608	-0.44233
-0.43095	-0.40238							
0.00000	-0.12156	-0.23827	-0.34548	-0.43892	-0.51486	-0.57028	-0.60296	-0.61160
-0.59586	-0.55636							
0.00000	-0.15035	-0.29471	-0.42732	-0.54290	-0.63683	-0.70537	-0.74579	-0.75648
-0.73701	-0.68816							
0.00000	-0.17316	-0.33941	-0.49213	-0.62523	-0.73341	-0.81234	-0.85889	-0.87120
-0.84878	-0.79252							
0.00000	-0.18905	-0.37057	-0.53731	-0.68264	-0.80075	-0.88693	-0.93776	-0.95120
-0.92672	-0.86529							
0.00000	-0.19742	-0.38696	-0.56108	-0.71283	-0.83616	-0.92616	-0.97923	-0.99327
-0.96770	-0.90356							
0.00000	-0.19791	-0.38792	-0.56248	-0.71460	-0.83824	-0.92846	-0.98167	-0.99574
-0.97011	-0.90581							
0.00000	-0.19051	-0.37342	-0.54145	-0.68789	-0.80691	-0.89375	-0.94497	-0.95852
-0.93385	-0.87195							
0.00000	-0.17552	-0.34403	-0.49884	-0.63375	-0.74340	-0.82341	-0.87060	-0.88308
-0.86035	-0.80332							
0.00000	-0.15352	-0.30093	-0.43634	-0.55435	-0.65026	-0.72025	-0.76152	-0.77243
-0.75255	-0.70267							
0.00000	-0.12541	-0.24583	-0.35644	-0.45284	-0.53119	-0.58837	-0.62208	-0.63100
-0.61476	-0.57401							
0.00000	-0.09230	-0.18092	-0.26233	-0.33329	-0.39095	-0.43303	-0.45784	-0.46440
-0.45245	-0.42246							
0.00000	-0.05551	-0.10881	-0.15777	-0.20044	-0.23512	-0.26043	-0.27535	-0.27930
-0.27211	-0.25407							
0.00000	-0.01651	-0.03236	-0.04692	-0.05960	-0.06992	-0.07744	-0.08188	-0.08305
-0.08092	-0.07555							
Columns 12 through 22:								
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000							
0.16062	0.13419	0.10241	0.06655	0.02804	-0.01160	-0.05077	-0.08792	-0.12156
-0.15035	-0.17316							
0.31484	0.26304	0.20075	0.13045	0.05495	-0.02273	-0.09951	-0.17233	-0.23827
-0.29471	-0.33941							

0.45651 0.38140 0.29107 0.18915 0.07968 -0.03296 -0.14429 -0.24987 -0.34548  
-0.42732 -0.49213  
0.57998 0.48455 0.36980 0.24031 0.10123 -0.04188 -0.18331 -0.31744 -0.43892  
-0.54290 -0.62523  
0.68033 0.56838 0.43378 0.28188 0.11875 -0.04912 -0.21503 -0.37237 -0.51486  
-0.63683 -0.73341  
0.75355 0.62956 0.48047 0.31222 0.13153 -0.05441 -0.23817 -0.41245 -0.57028  
-0.70537 -0.81234  
0.79673 0.66564 0.50800 0.33011 0.13907 -0.05752 -0.25182 -0.43608 -0.60296  
-0.74579 -0.85889  
0.80815 0.67518 0.51528 0.33485 0.14106 -0.05835 -0.25543 -0.44233 -0.61160  
-0.75648 -0.87120  
0.78735 0.65780 0.50202 0.32623 0.13743 -0.05685 -0.24886 -0.43095 -0.59586  
-0.73701 -0.84878  
0.73516 0.61420 0.46874 0.30460 0.12832 -0.05308 -0.23236 -0.40238 -0.55636  
-0.68816 -0.79252  
0.65367 0.54611 0.41678 0.27084 0.11410 -0.04720 -0.20660 -0.35778 -0.49468  
-0.61187 -0.70467  
0.54611 0.45625 0.34820 0.22627 0.09532 -0.03943 -0.17261 -0.29891 -0.41329  
-0.51119 -0.58872  
0.41678 0.34820 0.26574 0.17269 0.07275 -0.03009 -0.13173 -0.22812 -0.31541  
-0.39013 -0.44930  
0.27084 0.22627 0.17269 0.11222 0.04727 -0.01955 -0.08560 -0.14824 -0.20497  
-0.25352 -0.29197  
0.11410 0.09532 0.07275 0.04727 0.01991 -0.00824 -0.03606 -0.06245 -0.08635  
-0.10680 -0.12300  
-0.04720 -0.03943 -0.03009 -0.01955 -0.00824 0.00341 0.01492 0.02583 0.03572  
0.04418 0.05088  
-0.20660 -0.17261 -0.13173 -0.08560 -0.03606 0.01492 0.06530 0.11308 0.15635  
0.19339 0.22272  
-0.35778 -0.29891 -0.22812 -0.14824 -0.06245 0.02583 0.11308 0.19582 0.27076  
0.33490 0.38569  
-0.49468 -0.41329 -0.31541 -0.20497 -0.08635 0.03572 0.15635 0.27076 0.37437  
0.46306 0.53328  
-0.61187 -0.51119 -0.39013 -0.25352 -0.10680 0.04418 0.19339 0.33490 0.46306  
0.57275 0.65961  
-0.70467 -0.58872 -0.44930 -0.29197 -0.12300 0.05088 0.22272 0.38569 0.53328  
0.65961 0.75964  
-0.76937 -0.64277 -0.49055 -0.31878 -0.13429 0.05555 0.24317 0.42110 0.58225  
0.72017 0.82939  
-0.80340 -0.67120 -0.51225 -0.33287 -0.14023 0.05801 0.25393 0.43973 0.60800  
0.75203 0.86608  
-0.80540 -0.67287 -0.51352 -0.33370 -0.14058 0.05815 0.25456 0.44082 0.60951  
0.75390 0.86823  
-0.77529 -0.64772 -0.49433 -0.32123 -0.13532 0.05598 0.24504 0.42434 0.58673  
0.72572 0.83578

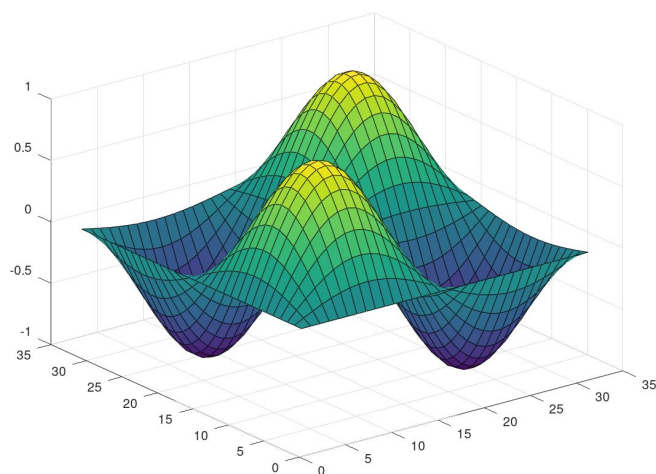


```

-0.71427 -0.59674 -0.45542 -0.29595 -0.12467 0.05157 0.22576 0.39095 0.54055
0.66860 0.77000
-0.62478 -0.52197 -0.39836 -0.25887 -0.10905 0.04511 0.19747 0.34196 0.47282
0.58483 0.67352
-0.51038 -0.42640 -0.32542 -0.21147 -0.08908 0.03685 0.16131 0.27935 0.38625
0.47774 0.55020
-0.37563 -0.31382 -0.23950 -0.15564 -0.06556 0.02712 0.11872 0.20560 0.28427
0.35161 0.40494
-0.22591 -0.18873 -0.14404 -0.09360 -0.03943 0.01631 0.07140 0.12365 0.17096
0.21146 0.24353
-0.06718 -0.05612 -0.04283 -0.02783 -0.01173 0.00485 0.02123 0.03677 0.05084
0.06288 0.07242
Columns 23 through 32:
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000
-0.18905 -0.19742 -0.19791 -0.19051 -0.17552 -0.15352 -0.12541 -0.09230 -0.05551
-0.01651
-0.37057 -0.38696 -0.38792 -0.37342 -0.34403 -0.30093 -0.24583 -0.18092 -0.10881
-0.03236
-0.53731 -0.56108 -0.56248 -0.54145 -0.49884 -0.43634 -0.35644 -0.26233 -0.15777
-0.04692
-0.68264 -0.71283 -0.71460 -0.68789 -0.63375 -0.55435 -0.45284 -0.33329 -0.20044
-0.05960
-0.80075 -0.83616 -0.83824 -0.80691 -0.74340 -0.65026 -0.53119 -0.39095 -0.23512
-0.06992
-0.88693 -0.92616 -0.92846 -0.89375 -0.82341 -0.72025 -0.58837 -0.43303 -0.26043
-0.07744
-0.93776 -0.97923 -0.98167 -0.94497 -0.87060 -0.76152 -0.62208 -0.45784 -0.27535
-0.08188
-0.95120 -0.99327 -0.99574 -0.95852 -0.88308 -0.77243 -0.63100 -0.46440 -0.27930
-0.08305
-0.92672 -0.96770 -0.97011 -0.93385 -0.86035 -0.75255 -0.61476 -0.45245 -0.27211
-0.08092
-0.86529 -0.90356 -0.90581 -0.87195 -0.80332 -0.70267 -0.57401 -0.42246 -0.25407
-0.07555
-0.76937 -0.80340 -0.80540 -0.77529 -0.71427 -0.62478 -0.51038 -0.37563 -0.22591
-0.06718
-0.64277 -0.67120 -0.67287 -0.64772 -0.59674 -0.52197 -0.42640 -0.31382 -0.18873
-0.05612
-0.49055 -0.51225 -0.51352 -0.49433 -0.45542 -0.39836 -0.32542 -0.23950 -0.14404
-0.04283
-0.31878 -0.33287 -0.33370 -0.32123 -0.29595 -0.25887 -0.21147 -0.15564 -0.09360
-0.02783
-0.13429 -0.14023 -0.14058 -0.13532 -0.12467 -0.10905 -0.08908 -0.06556 -0.03943
-0.01173

```

0.05555	0.05801	0.05815	0.05598	0.05157	0.04511	0.03685	0.02712	0.01631
0.00485								
0.24317	0.25393	0.25456	0.24504	0.22576	0.19747	0.16131	0.11872	0.07140
0.02123								
0.42110	0.43973	0.44082	0.42434	0.39095	0.34196	0.27935	0.20560	0.12365
0.03677								
0.58225	0.60800	0.60951	0.58673	0.54055	0.47282	0.38625	0.28427	0.17096
0.05084								
0.72017	0.75203	0.75390	0.72572	0.66860	0.58483	0.47774	0.35161	0.21146
0.06288								
0.82939	0.86608	0.86823	0.83578	0.77000	0.67352	0.55020	0.40494	0.24353
0.07242								
0.90555	0.94560	0.94795	0.91251	0.84070	0.73536	0.60071	0.44212	0.26589
0.07907								
0.94560	0.98742	0.98988	0.95287	0.87788	0.76789	0.62728	0.46167	0.27765
0.08257								
0.94795	0.98988	0.99234	0.95525	0.88007	0.76980	0.62885	0.46282	0.27834
0.08277								
0.91251	0.95287	0.95525	0.91954	0.84717	0.74102	0.60534	0.44552	0.26794
0.07968								
0.84070	0.87788	0.88007	0.84717	0.78049	0.68270	0.55770	0.41045	0.24685
0.07341								
0.73536	0.76789	0.76980	0.74102	0.68270	0.59716	0.48782	0.35903	0.21592
0.06421								
0.60071	0.62728	0.62885	0.60534	0.55770	0.48782	0.39850	0.29329	0.17639
0.05245								
0.44212	0.46167	0.46282	0.44552	0.41045	0.35903	0.29329	0.21586	0.12982
0.03860								
0.26589	0.27765	0.27834	0.26794	0.24685	0.21592	0.17639	0.12982	0.07807
0.02322								
0.07907	0.08257	0.08277	0.07968	0.07341	0.06421	0.05245	0.03860	0.02322
0.00690								



## 6. Rysowanie wykresów

6.1. Narysować na jednym rysunku wykresy funkcji  $y=x$ ,  $y=x^2$ ,  $y=x^3$  dla przedziału argumentów  $[-1, 1]$ .

```
x=[-1:0.1:1]
```

```
xsize=size(x)
```

```
y1=x(1:xsize(2));
```

```
y2=x(1:xsize(2)).^(3/2);
```

```
y3=x(1:xsize(2)).^2;
```

```
y4=x(1:xsize(2)).^3
```

```
plot(y1,x,y2,x,y3,x,y4,x)
```

```
x =
```

```
Columns 1 through 11:
```

```
-1.00000 -0.90000 -0.80000 -0.70000 -0.60000 -0.50000 -0.40000 -0.30000 -0.20000  
-0.10000 0.00000
```

```
Columns 12 through 21:
```

```
0.10000 0.20000 0.30000 0.40000 0.50000 0.60000 0.70000 0.80000 0.90000  
1.00000
```

```
xsize =
```

```
1 21
```

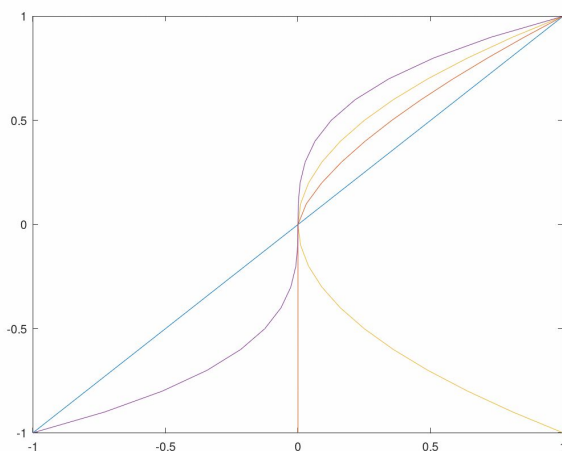
```
y4 =
```

```
Columns 1 through 11:
```

```
-1.00000 -0.72900 -0.51200 -0.34300 -0.21600 -0.12500 -0.06400 -0.02700 -0.00800  
-0.00100 0.00000
```

```
Columns 12 through 21:
```

```
0.00100 0.00800 0.02700 0.06400 0.12500 0.21600 0.34300 0.51200 0.72900  
1.00000
```



## Ćwiczenia – uruchamianie prostych skryptów

1.1. Napisać skrypt test.m, w którym utworzymy macierz  $A=[1 \ 2 \ 3; \ 4 \ 7 \ 0]$ . Do zmiennej (wektor dwuelementowy) rozA przypiszemy jej rozmiar. Po uruchomieniu skrypt powinien wyświetlić:

```

A=[1 2 3; 4 7 0];
rozA=size(A);
disp("Maciez A rowna sie:")
disp(A)
printf("ma rozmiar:")
disp(rozA)
Maciez A rowna sie:
  1  2  3
  4  7  0
ma rozmiar:  2  3

```

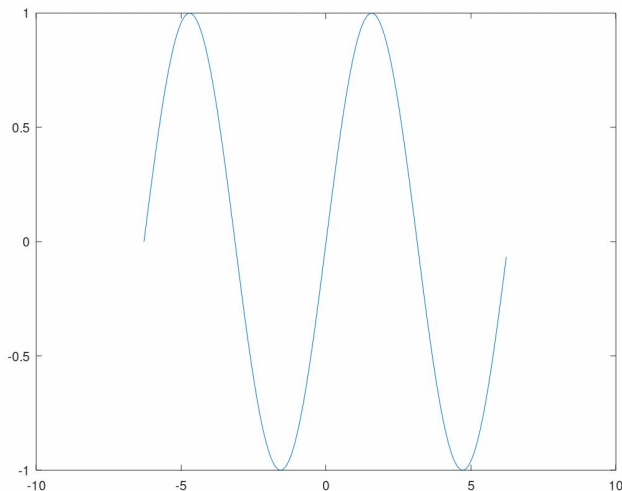
**1.2. Do skryptu test.m dodać instrukcję, która narysuje wykres funkcji sinus dla  $x = [-2\pi, 2\pi]$ . Wyświetlanie informacji o macierzy A zakomentować (znak % na początku linii). Przetestować.**

```

x=[-pi*2:0.1:pi*2];
xsize=size(x)
y=[y=sin(x(1:xsize(2)))]
plot(x,y)
x =
Columns 1 through 10:
-6.283185 -5.983185 -5.683185 -5.383185 -5.083185 -4.783185 -4.483185 -4.183185
-3.883185 -3.583185
Columns 11 through 20:
-3.283185 -2.983185 -2.683185 -2.383185 -2.083185 -1.783185 -1.483185 -1.183185
-0.883185 -0.583185
Columns 21 through 30:
-0.283185 0.016815 0.316815 0.616815 0.916815 1.216815 1.516815 1.816815
2.116815 2.416815
Columns 31 through 40:
2.716815 3.016815 3.316815 3.616815 3.916815 4.216815 4.516815 4.816815
5.116815 5.416815
Columns 41 and 42:
5.716815 6.016815
xsize =
  1  42
y =
Columns 1 through 8:
2.4492e-16 2.9552e-01 5.6464e-01 7.8333e-01 9.3204e-01 9.9749e-01
9.7385e-01 8.6321e-01
Columns 9 through 16:
6.7546e-01 4.2738e-01 1.4112e-01 -1.5775e-01 -4.4252e-01 -6.8777e-01
-8.7158e-01 -9.7753e-01
Columns 17 through 24:
-9.9616e-01 -9.2581e-01 -7.7276e-01 -5.5069e-01 -2.7942e-01 1.6814e-02
3.1154e-01 5.7844e-01
Columns 25 through 32:

```

7.9367e-01 9.3800e-01 9.9854e-01 9.6989e-01 8.5460e-01 6.6297e-01  
 4.1212e-01 1.2445e-01  
 Columns 33 through 40:  
 -1.7433e-01 -4.5754e-01 -6.9987e-01 -8.7970e-01 -9.8094e-01 -9.9455e-01  
 -9.1933e-01 -7.6198e-01  
 Columns 41 and 42:  
 -5.3657e-01 -2.6323e-01



**1.3. Napisać skrypt trojkat.m, w którym przypiszemy trzem zmiennym (a, b i c) wartości 3.17, 4.5, i 5. Z wzoru Herona proszę wyliczyć pole (warto stworzyć sobie zmienną pomocniczą p). Skrypt powinien wyświetlać informację na dwa sposoby (korzystając z funkcji disp oraz printf):**

**Trójkąt o bokach a=3.17, b=4.5, c=5 ma pole P=7.0084**

**Trójkąt o bokach a=3.17, b=4.500000, c=5.000000 ma pole P=7.0084**

**a=3.17;**

**b=4.5;**

**c=5;**

**p=1/2\*(a+b+c);**

**P=sqrt(p\*(p-a)\*(p-b)\*(p-c));**

**printf("Trojkat o bokach a=%f, b=%f, c=%f ma pole P=%f\n",a,b,c,P)**

**disp("Trojkat o bokach a=")**

**disp(a)**

**disp(", b=")**

**disp(b)**

**disp(", c=")**

**disp(c)**

**disp("ma pole P=")**

**disp(P)**

*Trojkat o bokach a=3.170000, b=4.500000, c=5.000000 ma pole P=7.008399*

*Trojkat o bokach a=*

*3.1700*

*, b=*

4.5000  
, c=  
5  
ma pole P=  
7.0084

## Ćwiczenia – instrukcja warunkowa

**2.1. W skrypcie trojkat2.m utworzyć trzy zmienne (a, b, c) mające dowolne długości, reprezentujące boki trójkąta. Wykorzystać instrukcję warunkową do sprawdzenia, czy z boków o podanych długościach można zbudować trójkąt. Dla obu możliwych rozwiązań wyświetlić właściwą informację.**

```
a=randi(10,1)
b=randi(10,1)
c=randi(10,1)
if(a+b>c && a+c>b && b+c>a)
disp("trojkant mozna utworzyc");
else
disp("trojkanta nie mozna utworzyc");
endif
a = 10
b = 10
c = 6
trojkant mozna utworzyc
```

## Ćwiczenia – pętle (for, while, ...) i macierze

**3.1. Korzystając z pętli for albo while wyświetlić 200 razy napis "n. Będę się pilnie uczył.", gdzie zamiast n, powinien być numer wyświetlanej linii (od 1 do 200).**

```
for i=1:200
    printf("%0f Bede sie pilnie uczyl\n",i);
endfor
1 Bede sie pilnie uczyl
2 Bede sie pilnie uczyl
3 Bede sie pilnie uczyl
...
198 Bede sie pilnie uczyl
199 Bede sie pilnie uczyl
200 Bede sie pilnie uczyl
```

### 3.2. Korzystając z pętli for i funkcji printf() wyświetlić tabliczkę mnożenia, mniej więcej taką:

Pierwsza wersja.

```
for i=1:10
    for j=1:10
        a=i*j;
        printf("%d ",a);
    endfor
    printf("\n");
endfor
```

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

Druga wersja.

```
for i=1:10
    for j=1:10
        A(i,j)=i*j;
    endfor
endfor
disp(A)
```

```
1  2  3  4  5  6  7  8  9 10
2  4  6  8 10 12 14 16 18 20
3  6  9 12 15 18 21 24 27 30
4  8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

**3.3. Korzystając z pętli for, utworzyć macierz B zwiększając każdy element macierzy A o 3. Porównaj macierz B wynikiem operacji  $A+3$ .**

```
A=[
  1 8 -3;
  8 1 0;
  9 0 8]
for i=1:3
  for j=1:3
    B(i,j)=A(i,j)+3;
  endfor
endfor
disp(A)
disp(B)
```

1	8	-3
8	1	0
9	0	8

4	11	0
11	4	3
12	3	11

**3.4. Korzystając z pętli for, utworzyć macierz C podnosząc do potęgi 2 wszystkie elementy leżące na głównej przekątnej (taki sam indeks wiersza i kolumny). Sprawdzić wynik**

```
C=A;
for i=1:3
  C(i,i)=C(i,i)*C(i,i);
endfor
disp(A)
disp(C)
```

1	8	-3
8	1	0
9	0	8

1	8	-3
8	1	0
9	0	64

**3.5. Korzystając z pętli for utworzyć macierz D równą sumie macierzy B i C. Porównać macierz B z wynikiem operacji  $B+C$ .**

```
D=B+C;
for i=1:3
  C(i,i)=C(i,i)*C(i,i);
endfor
disp(B)
disp(D)
```



```
4 11 0
11 4 3
12 3 11
```

```
5 19 -3
19 5 3
21 3 75
```

**3.6. Korzystając z pętli, instrukcji warunkowej i funkcji isprime(), sprawdzającej, czy podana liczba jest liczbą pierwszą, wyświetlić wszystkie liczby pierwsze mniejsze od 100. Zadanie rozwiązać na dwa sposoby, za pomocą pętli for i pętli while. Porównaj wynik z poleceniem: primes(100).**

```
a=1;
for i=1:100
    if(isprime(i)==1)
        p(a)=i;
        a++;
    endif
endfor
```

```
disp(p)
```

```
a=1;
while(i<100)
    if(isprime(i)==1)
        p(a)=i;
        a++;
    endif
    i++;
endwhile
```

```
disp(p)
```

```
primes(100)
```

```
Columns 1 through 23:
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83
```

```
Columns 24 and 25:
```

```
89 97
```

```
Columns 1 through 23:
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83
```

```
Columns 24 and 25:
```

```
89 97
```

```
ans =
```

```
Columns 1 through 23:
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
79 83
```

```
Columns 24 and 25:
```

```
89 97
```