Part 1: Problems from Gilat, Ch. 3.9

1. For the function

$$y = \frac{(2x^2 - 5x + 4)^3}{x^2},$$

calculate the value of y for the following values of x: $-2, -1, 0, 1, 2, 3, 4, 5$ using element-by-element operations.

```
INPUT

x = -2:1:5
y = (2.*x.^2 - 5.*x + 4).^3 ./ x.^2

OUTPUT

x =

  -2  -1   0   1   2   3   4   5

y =

 Columns 1 through 6:

   2662.0000   1331.0000         Inf      1.0000      2.0000     38.1111

 Columns 7 and 8:

    256.0000    975.5600
```

2. For the function

$$y = 5\sqrt{t} - \frac{(t+2)^2}{0.5(t+1)} + 8,$$

calculate the value of y for the following values of t: $0, 1, 2, 3, 4, 5, 6, 7, 8$ using element-by-element operations.

```
INPUT

t = 0:1:8
y = 5.*sqrt(t) - (t+2).^2 ./ (0.5.*(t + 1)) + 8

OUTPUT

t =

  0  1  2  3  4  5  6  7  8

y =

 Columns 1 through 8:

   0.00000   4.00000   4.40440   4.16025   3.60000   2.84701   1.96173   0.97876

 Column 9:

  -0.08009
```

6. The position as a function of time $(x(t), y(t))$ of a projectile fired with a speed of v_0 at an angle θ is given by

$$\begin{aligned} x(t) &= v_0 \cos\theta \cdot t \\ y(t) &= v_0 \sin\theta \cdot t - \frac{1}{2}gt^2 \end{aligned}$$

where $g = 9.81\text{m/s}^2$ is the gravitation of the Earth. The distance r to the projectile at time at time t can be calculated by $r(t) = \sqrt{x(t)^2 + y(t)^2}$. Consider the case where $v_0 = 100\,\text{m/s}$ and $\theta = 79°$. Determine the distance r to the projectile for $t = 0, 2, 4, \ldots, 20$ s.

```
INPUT

v0 = 100;
g = 9.81;
thetad = 79;

t = 0:2:20;

x = v0.*cosd(thetad).*t;
y = v0.*sind(thetad).*t - (0.5*g).*t.^2;

r = sqrt(x.^2 + y.^2)

OUTPUT

r =

 Columns 1 through 6:

     0.00000    180.77924    323.30888    427.99255    495.48150    526.89086

 Columns 7 through 11:

   524.27565    491.77701    438.61312    386.70741    381.62005
```

8. Define x and y as the vectors $x = 2, 4, 6, 8, 10$ and $y = 3, 6, 9, 12, 15$. Then use them in the following expression to calculate z using element-by-element calculations.

$$z = \left(\frac{y}{x}\right)^2 + (x + y)^{\left(\frac{y-x}{x}\right)}$$

```
INPUT

x = 2:2:10
y = 3:3:15
z = (y./x).^2 + (x + y).^((y - x)./x)

OUTPUT

x =

   2   4   6   8  10

y =

   3   6   9  12  15

z =

   4.4861   5.4123   6.1230   6.7221   7.2500
```

10. Show that

$$\lim_{x \to 0} \frac{e^x - 1}{x} = 1$$

Do this by first creating a vector x that has the elements: $1, 0.5, 0.1, 0.01, 0.001, 0.00001$ and 0.0000001. Then create a new vector y in which each element is determined from the elements of x by $(e^x - 1)/x$. Compare the elements of y with the value 1 (use `format long` to display the numbers).

```
INPUT

format long

x = [1 0.5 0.1 0.01 0.001 0.00001 0.0000001]
y = (exp(x) - 1) ./ x

y - 1
```

```
OUTPUT

x =

 Columns 1 through 3:

   1.00000000000000e+00   5.00000000000000e-01   1.00000000000000e-01

 Columns 4 through 6:

   1.00000000000000e-02   1.00000000000000e-03   1.00000000000000e-05

 Column 7:

   1.00000000000000e-07

y =

 Columns 1 through 4:

   1.71828182845905   1.29744254140026   1.05170918075648   1.00501670841679

 Columns 5 through 7:

   1.00050016670838   1.00000500000696   1.00000004943368

ans =

 Columns 1 through 3:

   7.18281828459045e-01   2.97442541400256e-01   5.17091807564771e-02

 Columns 4 through 6:

   5.01670841679491e-03   5.00166708384597e-04   5.00000696490588e-06

 Column 7:

   4.94336802603357e-08
```

12. Use `octave` to show that the sum of the infinite series

$$\sum_{n=0}^{\infty} \frac{1}{(2n+1)(2n+2)}$$

converges to ln 2. Show this by computing the sum for

1. $n = 50$
2. $n = 500$
3. $n = 5000$

For each part, create a vector n in which the first element is 0, the increment is 1 and the last term is 50, 500 or 5000. Then use element-by-element calculation to create a vector in which the elements are

$$\frac{1}{(2n+1)(2n+2)}$$

Finally, use the function `sum` to add the terms in the series. Compare the values obtained in parts 1, 2 and 3 to ln 2.

```
INPUT

n = 0:50;
ans1 = sum(1./((2*n + 1) .* (2*n + 2))) - log(2)

n = 0:500;
ans2 = sum(1./((2*n + 1) .* (2*n + 2))) - log(2)

n = 0:5000;
ans3 = sum(1./((2*n + 1) .* (2*n + 2))) - log(2)

OUTPUT

ans1 = -0.0048779
ans2 = -4.9875e-04
ans3 = -4.9988e-05
```

18. Solve the following system of five linear equations:

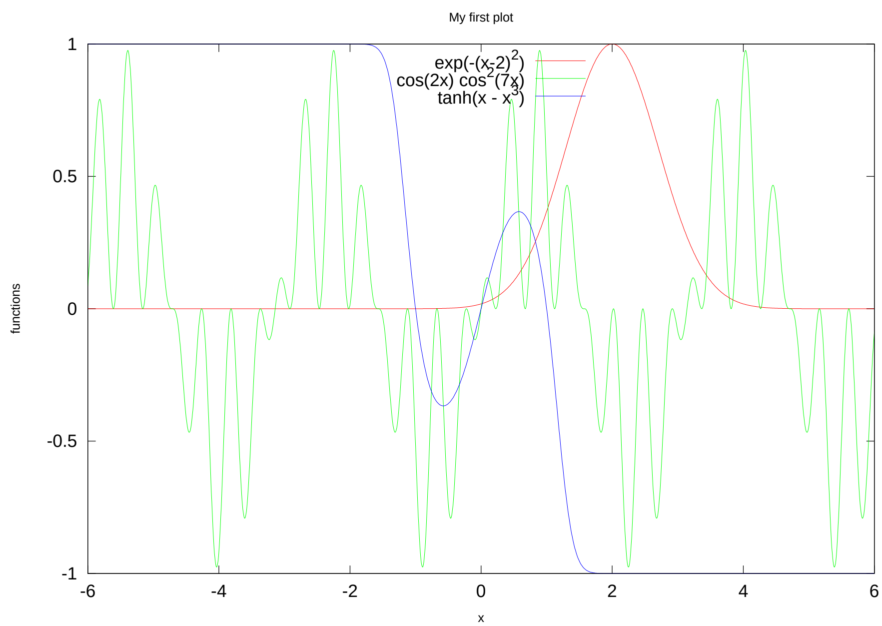
$$\begin{aligned}
1.5x - 2y + z + 3u + 0.5w &= 7.5 \\
3x + y - z + 4u - 3w &= 16 \\
2x + 6y - 3z - u + 3w &= 78 \\
5x + 2y + 4z - 2u + 6w &= 71 \\
-3x + 3y + 2z + 5u + 4w &= 54
\end{aligned}$$

```
INPUT

A = [1.5 -2 1 3 0.5; 3 1 -1 4 -3; 2 6 -3 -1 3; 5 2 4 -2 6; -3 3 2 5 4];
B = [7.5; 16; 78; 71; 54];

X = A \ B

OUTPUT

X =

   5.0000
   7.0000
  -2.0000
   4.0000
   8.0000
```

Part 2: Basic 2D plotting with `octave`

Sample implementation of `myplot.m`

```
clf
hold on
x = linspace(-6, 6, 2000);

plot(x, exp(-1 * (x - 2).^2), '-r');
plot(x, sin(2*x) .* cos(7*x).^2, '-g');
plot(x, tanh(x - x.^3), '-b');

xlabel('x');
ylabel('functions');

title('My first plot');

legend('exp(-(x-2)^2)', 'cos(2x) cos^2(7x)', 'tanh(x - x^3)', ...
'location' , 'north');

print('myplot.ps','-depsc')
```



Part 3: (Pseudo)-Random Numbers

1. Demonstrate that the mean value of the random numbers generated by `rand` approaches 0.5 as the length, n, of the random number sequence approaches ∞. Do this by computing the mean value for sequences of length $n = 10, 10^2, 10^3, 10^4, 10^5, 10^6$ and 10^7.

INPUT

```
for n = [10 100 1000 10000 100000 1000000 10000000]
   mean(rand(1,n))
end
```

OUTPUT

```
ans =  0.64234
ans =  0.44700
ans =  0.52028
ans =  0.50103
ans =  0.50141
ans =  0.49962
ans =  0.49997
```

2a. Demonstrate that the mean value of the random numbers generated by `randn` approaches 0.0, and the standard deviation approaches 1.0, as the length, n, of the random number sequence approaches ∞. Do this by computing the mean value and standard deviation for sequences of length $n = 10, 10^2, 10^3, 10^4, 10^5, 10^6$ and 10^7.

INPUT

```
for n = [10 100 1000 10000 100000 1000000 10000000]
   mean(randn(1,n))
   std(randn(1,n))
disp('')
end
```

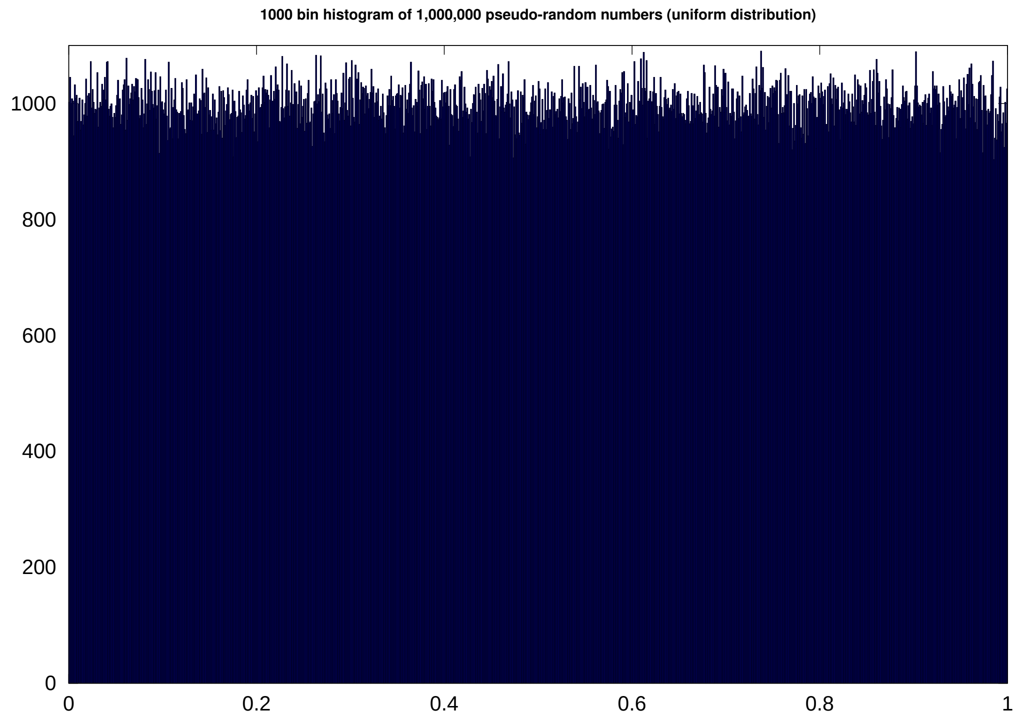
OUTPUT

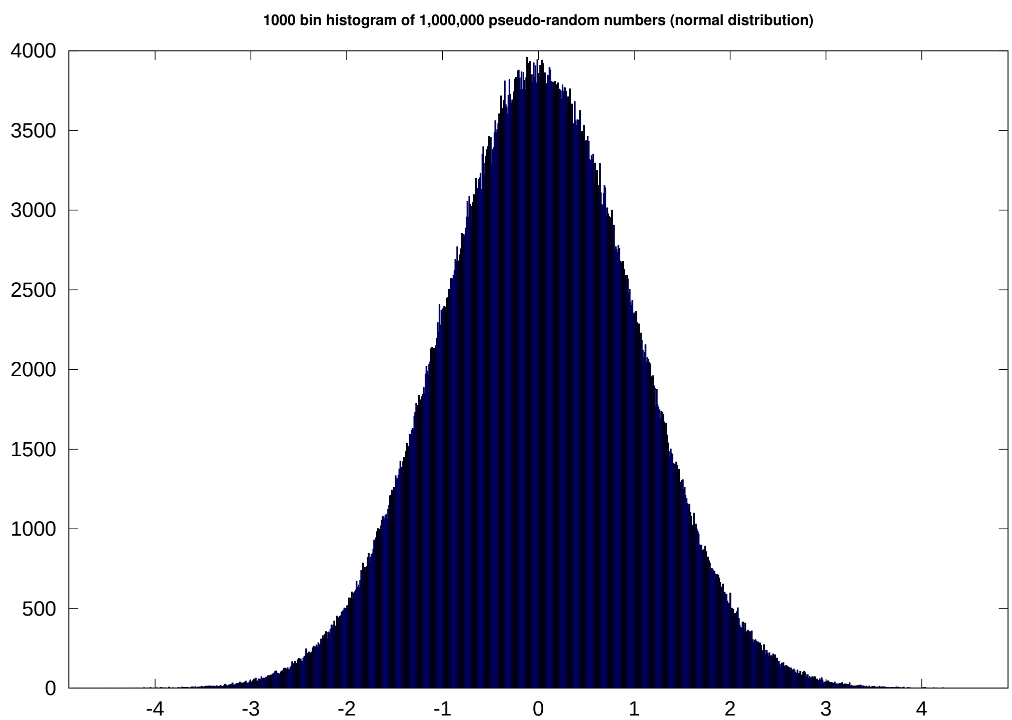
```
ans =  0.036533
ans =  1.3290

ans = -0.036698
ans =  1.0833

ans =  0.047042
ans =  0.98887

ans =  0.018344
ans =  0.99587

ans =  7.4008e-05
ans =  0.99610

ans = -3.6768e-07
ans =  1.0001

ans =  5.5271e-05
ans =  1.0004
```

One notes that the convergence towards the expected mean of 0 and standard deviation of 1 is not as systematic as might be expected for $n = 10^5$, 10^6 and 10^{-7}. It may be that limitations in `octave`'s algorithm for generating normally distributed pseudo-random numbers are being reached, but this would require more detailed study.

2b. Use `octave`'s `hist` function (type `doc hist` for usage information) to plot histograms with 1000 bars for the case of a million random numbers generated by `rand` and `randn` respectively.

```
clf
hold on
hist(rand(1,1000000),1000);
title('{\bf 1000 bin histogram of 1,000,000 pseudo-random numbers (uniform distribution)}');
print('hist-rand.ps','-depsc');

clf
hold on
hist(randn(1,1000000),1000)
title('{\bf 1000 bin histogram of 1,000,000 pseudo-random numbers (normal distribution)}');
print('hist-randn.ps','-depsc');
```


1000 bin histogram of 1,000,000 pseudo-random numbers (uniform distribution)
0
200
400
600
800
1000
0
0.2
0.4
0.6
0.8
1


1000 bin histogram of 1,000,000 pseudo-random numbers (normal distribution)
0
500
1000
1500
2000
2500
3000
3500
4000
-4
-3
-2
-1
0
1
2
3
4