+3 for presentation

CS 375, HW 1

Ryan Scherbarth, University of New Mexico

August 26, 2024

1. Suppose z = [10, 40, 70, 90, 20, 30, 50, 60]. What does this vector look like after each of these commands? Assume that the commands are done sequentially. You do not need to submit any code for this problem.

   (a) z(1:3:7) = zeros(1,3)

   $\boxed{\textbf{z=[0 40 70 0 20 30 0 60]}}$

   (b) z([3 4 1]) = []

   $\boxed{\textbf{z=[40 20 30 0 60]}}$

2. (a) Use the linspace function to create vectors identitical to the following created with colon notation:
   i. t = 1:4:25
   t = [1 5 9 13 17 21 25]

   $\boxed{\textbf{linspace(1, 25, 7)}}$

   ii. x = -11:1
   x = [-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1]

   $\boxed{\textbf{linspace(-11, 1, 13)}}$

   (b) Use colon notation to create vectors identitical to the following created with the linspace function:
   i. v = linspace(-10, -8, 6)
   v = [-10.0000 -9.6000 -9.2000 -8.8000 -8.4000 -8.0000]

   $\boxed{\textbf{v = -10:0.4:-8}}$

ii. r = linspace(0, 1, 5)
r = [0 0.2500 0.5000 0.7500 1.0000]

$$\boxed{\mathbf{r = 0{:}0.25{:}1}}$$

3. Given that t=0:0.1:1; y=sin(π*t); write a single-line matlab code that returns each of the following:
(a) $\sum_{k=1}^{N} t_k$

$$\boxed{\textbf{sum(t)}}$$

(b) $\sum_{k=1}^{N} t_k y_k$

$$\boxed{\textbf{sum(t .* y)}}$$

(c) $\sum_{k=1}^{N} t_k^2$
I cannot figure out how to format this in latek. It should be . ˆ inside of the sum. sum(t .(up arrow) 2)

$$\boxed{\textbf{sum(t . }\hat{\mathbf{2}}\textbf{)}}$$

4. Write a matlab script to plot the four functions x, exp(x), x^2, x^3, over the interval 0 ≤ x ≤ 1 using plot, semiology, semilogx, and loglog

Script:

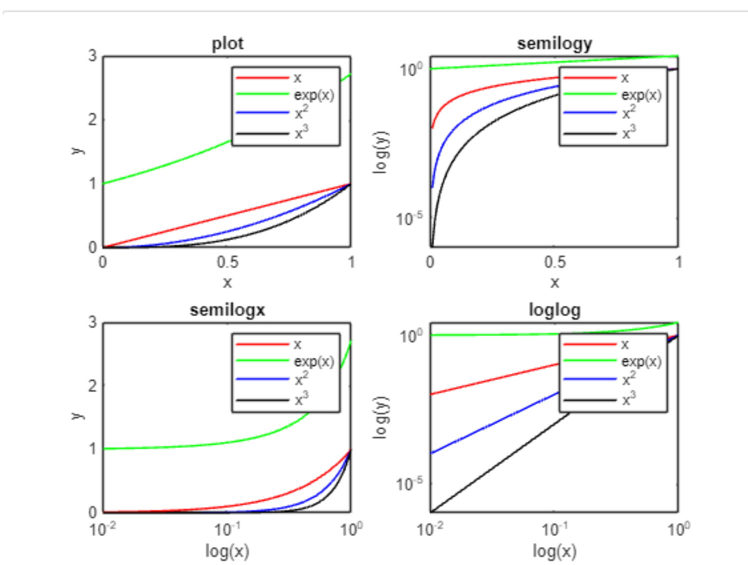
```
x = linspace(0, 1, 100);
f1 = x;
f2 = exp(x);
f3 = x.^2;
f4 = x.^3;

figure;
subplot(2,2,1);
plot(x, f1, '-r', x, f2, '-g', x, f3, '-b', x, f4, '-k');
title('plot');
xlabel('x');
ylabel('y');
legend('x', 'exp(x)', 'x^2', 'x^3');
```

```
subplot(2,2,2);
semilogy(x, f1, '-r', x, f2, '-g', x, f3, '-b', x, f4, '-k');
title('semilogy');
xlabel('x');
ylabel('log(y)');
legend('x', 'exp(x)', 'x^2', 'x^3');

subplot(2,2,3);
semilogx(x, f1, '-r', x, f2, '-g', x, f3, '-b', x, f4, '-k');
title('semilogx');
xlabel('log(x)');
ylabel('y');
legend('x', 'exp(x)', 'x^2', 'x^3');

subplot(2,2,4);
loglog(x, f1, '-r', x, f2, '-g', x, f3, '-b', x, f4, '-k');
title('loglog');
xlabel('log(x)');
ylabel('log(y)');
legend('x', 'exp(x)', 'x^2', 'x^3');
```



The different types of graphs help us identify different rates in which

the functions are increasing quickest. The log linear graph, with $\log(x)$ as logarithmic terms, helps us identify if a line has logarithmic growth. If so, it will show as a straight line in this case.
The inverse of the previous with only the y axis logarithmic helps us determine exponential growth, in teh same way, an exponential function on this graph will be a straight line.
The log log graph is where we make both axis logarithmic, in this case, power law functions ($y = kx^n$), will appear as straight lines: $O(n^k)$.

5. Write a matlab function called my mean which takes four arguments, a function name fun, number a, number b, s.t. $a \leq b$, and number N s.t. $N > 0$. Then return ...

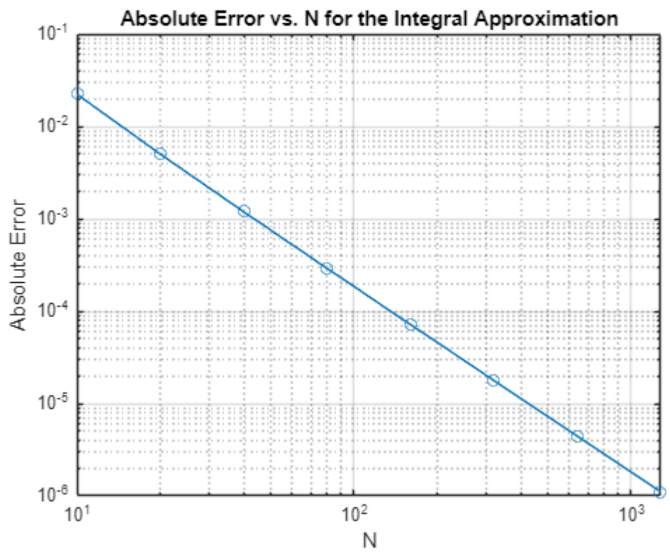
(a)

```
function I = my_mean(fun, a, b, N)
    h = (b - a) / (N - 1);
    x = linspace(a, b, N);
    I = (h / 2) * (fun(x(1)) + fun(x(N))) + h * sum(fun(x(2:N-1)));
end
```

(b)

```
function y = my_fun(x)
    y = x .* exp(x);
end
```

(c)



N	Approximation of M	Absolute Error
10	0.758097295839530	0.022338413496646
20	0.740777065435559	0.005018183092675
40	0.736950230001999	0.001191347659114
80	0.736049244751231	0.000290362408346
160	0.735830563802275	0.000071681459391
320	0.735776690596261	0.000017808253377
640	0.735763320486732	0.000004438143848
1280	0.735759990144773	0.000001107801888

We see that as we increase the N, number of rectangles under the curve, we increase the accuracy. The approximation gets closer to the actual value and the absolute errors gets smaller. The error converges at a second-order rate.