

COEN 45
Winter Quarter, 2011
Homework #7
Due Tues Mar. 1

1. There are various ways to “average” a set of numbers in a vector x of length n . Three common methods are

- Arithmetic mean:

$$y = \frac{1}{n} (x_1 + x_2 + \cdots + x_n)$$

- Geometric mean (positive numbers):

$$y = (x_1 x_2 \cdots x_n)^{1/n}$$

- Harmonic mean (positive numbers):

$$y = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}}$$

Write a function that returns one of these averages according to the user’s specification:

```
function y = average(x,type)
%function y = average(x,type)
%
% Input
%   x = vector of numbers
%   type = char variable indicating type,
%         starting with 'a' or 'g' or 'h' (lower or upper case)
%
% Output
%   y = average value
%
```

`type` will be a character array. Check the first letter of this variable, convert to lower case if necessary, and accept 'a' or 'g' or 'h'. (Suggestion: use a `switch` statement.) Check `x` for values that would be inappropriate for the requested formula. Check out your function as follows:

```
x = 6:3:166;
a = average(x,'Arithmetic')
h = average(x,'h')
g = average(x,'geom')
x = 0:4:296;
a = average(x,'Arithmetic')
h = average(x,'h')
g = average(x,'geom')
```

2. Problem 36, p. 258: bandpass filter. Write a function `RV = lowpass(R,C,omega)` as described in the text. Do *not* create your plot inside the function, do this in the calling script instead, as follows:

```
R = ...
C = ...
omega = logspace(-2,6,200); %% frequencies spaced uniformly on a log scale
RV = lowpass(R,C,omega);    %% compute output voltage/input voltage
loglog(omega,RV);
xlabel('\omega, rad/sec');
ylabel('Voltage ratio V_o/V_i');
grid
```

Label your axes and add a title, “Frequency response for a lowpass filter.” What do you observe about this plot? Why is it called a filter? What gets filtered out and what is allowed to get through? The output voltage is less than 10% of the input voltage over what frequency range?

3. Problem 37, p. 259. Quite similar to the previous problem, but for a bandpass filter: `RV = bandpass(R,C,L,omega)`. Make a similar plot and answer the same questions.
4. (a) Write a `for` statement (which need not be a function) to calculate the first `n` terms of a Taylor series expansion of the exponential function

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

for given x and n , e.g.,

```
x = 2;
n = 4;
y = 0;
for ....
    y = ...
end
fprintf('exp(%f) is approximately %6.3f using %d terms\n',x,n,y);
```

- (b) Replace the `for` loop with a vector operation that computes `y` in a single line of code.