x = 1

y = [1 2 3 4 5 6]

z = [1 2 3; 4 5 6; 7 8 9]

a = [y y y]

b = [z z z]

c = [y; y; y]

d = [0:1/8:1; 0:1/16:.5; 0:1/32:.25]

e = [sin(2*pi*d)]

f = [y' y y']

g = [y y y']

h = [y y y']

i = [y y y']
»g=sin(2*pi*f)
%operation on list of sequences (ex. could have a list of sine sequences with different frequencies)
g =
Columns 1 through 4
0 0.70710678118655 1.00000000000000 0.70710678118655
0.38268343236509 0.70710678118655 0.92387953251129
0.19590032021613 0.38268343236509 0.5557023301960
Columns 5 through 8
0 -0.70710678118655 -1.00000000000000 -0.70710678118655
1.00000000000000 0.92387953251129 0.70710678118655 0.38268343236509
0.70710678118655 0.83146961230255 0.92387953251129 0.9807852804033
Column 9
0
1.00000000000000
»whos %variable list and sizes (helpful when doing vector multiplication)
Name      Size         Bytes  Class
a         3x3             72  double array
ans       1x10          4192  struct array
b         3x6            144  double array
c         6x3            144  double array
d         1x11           88  double array
e         1x11           88  double array
f         3x9           216  double array
g         3x9           216  double array
x         1x1            8  double array
y         1x6            48  double array
z         6x1            48  double array
Grand total is 276 elements using 5264 bytes
leaving 307693008 bytes of memory free.
»
»y-1 %addition
ans =
0 1 2 3 4 5
»y =
1
2
3
4
5
6
»z'*(y-1) %outer product 6x1 x 1x6 = 6x6 (makes a list of sequences, each sequence y multiplied by corresponding element of list in z)
ans =
0 1 2 3 4 5
0 2 4 6 8 10
0 3 6 9 12 15
0 4 8 12 16 20
0 5 10 15 20 25
0 6 12 18 24 30
»(y-1)*z %inner product 1x6 x 6x1 = 1x1 (multiplies each corresponding element of y and z, then adds the resulting elements together)
ans =
70
0*1+1*2+2*3+3*4+4*5+5*6  inner product
ans =
70
»[1 2 3]'*g %vector x matrix (sequence of inner products)
ans =
Columns 1 through 4
0 2.55774461196511 3.56226385946836 4.22157654526793
Columns 5 through 8
4.12132034355964 3.63506112074366 3.18585215990695 3.00031592475332
Column 9
3.00000000000000
»g was the list of sine sequences;
%but you can also think of it as a sequence of lists
%with each list being the values for the three sine waves at different times in the sequence.
%you are doing an inner product between [1 2 3] and each column (list) of the sine sequence.
%at each time in the sequence, you are scaling each sine wave by a different amount then adding them together.
»g = [sin(0) sin(0) sin(0) ...
sin(2pi) sin(2pi) sin(2pi) ...
sin(3pi) sin(3pi) sin(3pi) ...
]
»[1 2 3]'*[e**(1*sin(0)+2*sin(2pi)+3*sin(3pi)), 1*sin(1)+2*sin(2)+3*sin(3), 1*
 Function: input: f(row),X(row),fs(scalar),dur(scalar)
 output: xx(row)
 function xx=xsumcos(X,fs,dur)
 .
 .
 .
 xx =?
save as sumcos.m

```matlab
>> z1=5*exp(j*0.5*pi);
>> z2=5*exp(j*-0.25*pi);
>> zp=sumcos([1, 1/3],[z1,z2],10,1)

zp =
Columns 1 through 4
3.53553390593274   1.25442657826475  -0.08738044898976   0.18315912149992
Columns 5 through 8
2.05422141231050   4.82962913144534   7.39395888240421   8.64101238876062
Columns 9 through 11
7.90188453672496   5.20887876016010   1.29409522551261
```

```matlab
>> plot(zp)
```

```matlab
>> stem(zp)
```

```matlab
=======
>> [zp,t]=sumcos([1, 1/3],[z1,z2],10,1)

zp =
Columns 1 through 4
3.53553390593274   1.25442657826475  -0.08738044898976   0.18315912149992
Columns 5 through 8
2.05422141231050   4.82962913144534   7.39395888240421   8.64101238876062
Columns 9 through 11
7.90188453672496   5.20887876016010   1.29409522551261
```

```matlab
function [xx, t] = sumcos(f, X, fs, dur)

% make time sequence from fs & dur
t = ?
% use outer product to make a list of complex exponential sequences using f and t
% use the inner product to multiply list of phasors X with list of complex exponential sequences
xx = ?
```

```matlab
 función [xx,t]=sumcos(f,X,fs,dur) %make time sequence from fs & dur t=? % use outer product to make a list of complex exponential sequences using f and t % use the inner product to multiply list of phasors X with list of complex exponential sequences xx=?
```

=======
output xx and t
```matlab
>> [zp,t]=sumcos([1, 1/3],[z1,z2],10,1)

zp =
Columns 1 through 4
3.53553390593274   1.25442657826475  -0.08738044898976   0.18315912149992
Columns 5 through 8
2.05422141231050   4.82962913144534   7.39395888240421   8.64101238876062
Columns 9 through 11
7.90188453672496   5.20887876016010   1.29409522551261
```

```matlab
>> plot(zp)
```

```matlab
>> stem(zp)
```
Columns 1 through 4
0   0.10000000000000   0.20000000000000   0.30000000000000
Columns 5 through 8
0.40000000000000   0.50000000000000   0.60000000000000   0.70000000000000
Columns 9 through 11
0.80000000000000   0.90000000000000   1.00000000000000

>plot(z,p)

>cd dspfi
rst % change directory (you either must be in directory to use the functions, or add path of directory)
>ls % list directory
Writing_Fast_MATLAB_Code.pdf plinterp.m
alphacon.m    plpick.m
andemoc.m     pretty_w.m
bifsn.m       pumpkin.m
baboon.mat    replacez.m
beatcon.m     macks.mat
beatconb.m    show_img.m
cip.m         showspec.m
cosgen.m      snmodi
dsfp.mat       specptr.m
dsfp1.mat      striplot.m
dsfp2.mat      sumpndot.m
dsfpfirst.m    tools.gif
dmtfchk.m      tools.mat
dmtnmain.m     trulize.m
dehart12.mat   ucplot.m
dchart.mat     vowel_d.m
denav.m        wavesnds.m
denotes.m      waverisc.m
dft.m          wngui.m
furelise.mat  wngui.mat
insample.m     woodrond.m
inout.m        wrec.m
lab6.mat       winotes.m
lab7.mat       zcat.m
lenna.mat      zozonts.m
lenna12.mat    zdir
lenna_bl.mat   zone.mat
manipsin.m     zon012.mat
mattbx.m       zone0_mak.m
musicult.m     zprint.m
mysound.m      zvect.m
nveloper.m     zwgt.m
pez_31

>help zprint  % if you don't know what the command does, use help
ZPRINT printout complex # in rect and polar form
-----
usage:   zprint(z)
Z = vector of complex numbers; each one will be printed
in a format showing real, imag, mag and phase

>zprint([z1 z2])
Z = [X + jY  Magnitude  Phase  Phis(deg)]
0 -3.536 -3.536 5 86.00 90.00
3.536 -3.536 5 -86.00 -90.00

>help zvect
ZVECT Plot vectors in complex z-plane from zFrom to zTo
usage:   HV = zvect(From, <zTo>, <LTYPE>, <SCALE>)
Z:  is a vector of complex numbers; each one will be
displayed as an arrow emanating from the origin.
LTYPE: string containing any valid line type (see PLOT)
SCALE: controls size of arrowhead (default = 1.0)
(order of LTYPE and SCALE args doesn't matter)
HV: output handle from graphics of vector plot
** With only one input vector: `zvect(Z)`
displays Z as arrows emanating from the origin.
** If either zFrom or zTo is a scalar all vectors will
start or end at that point.

See also ZCAT, PLOT, COMPASS, ROSE, FEATHER, QUIVER.

```
»zvect([z1, z2])
```

```
»zcat([z1,z2])
```

=help zcat

**ZCAT**  Plot vectors in z-plane end-to-end

**usage:** `hv = zcat(Z, <LTYPE>, <SCALE>)`

*Z* = vector of complex numbers; each complex # is displayed
as a vector, with the arrows placed end-to-end

*LTYPE*: string containing any valid line type (see PLOT)

*SCALE*: varies size of arrowhead (default = 1.0)

(order of LTYPE and SCALE args doesn't matter)

*hv*: output handle from graphics of vector plot

See also ZVECT, COMPASS, ROSE, FEATHER, QUIVER.

```
»zcat([z1,z2])
```