2D graphics with matrix transforms

```
% ReactorLab.net 2016

% also see http://mathforum.org/mathimages/index.php/Transformation_Matrix
```

define original 2D object

```
% enter coordinates of vertices of one object
% easiest to type by entering x,y pairs as rows in array
g = [0.5 0.5
    1.5 0.5
    1.5 1.5
    0.5 1.5
    0.5 0.5];

% then transpose so x's in row 1, y's in row 2
g = g';

% then add row to make augmented matrix
[rows, cols] = size(g);
g = [g; ones(1,cols)];

% make backup copy to start fresh for each example below
gOrig = g;

% plot original object
plot(g(1,:),g(2,:),'k')
title('object (black) rotated (blue, red), translated (green) and zoomed (dashed)')
am = 4;
axis([-am am -am am])
```

translate 2D object

```
g = gOrig; % get backup copy

tx = 1.0; % units of translation in x direction
ty = 2.0; % units of translation in y direction

% generate translation matrix
t = eye(3,3);
t(1,3) = tx;
t(2,3) = ty;

% translate
g = t * g;

hold on
plot(g(1,:),g(2,:),'g')
axis([-am am -am am])
```

rotate 2D object about figure origin

```
g = gOrig; % get backup copy
```

```
% specify CCW rotation, use degrees here, so use sind, cosd
th = 60;

% specify rotation matrix
r = [cosd(th) -sind(th) 0
    sind(th) cosd(th)   0
       0       0        1];

% rotate object with matrix multiplication
g = r * g;

% each new x = cosd(th) * x - sind(th) * y
% each new y = sind(th) * x + cosd(th) * y

% plot rotated object and o at center of rotation
hold on
plot(g(1,:),g(2,:),'b',0,0,'bo')
axis([-am am -am am])
```

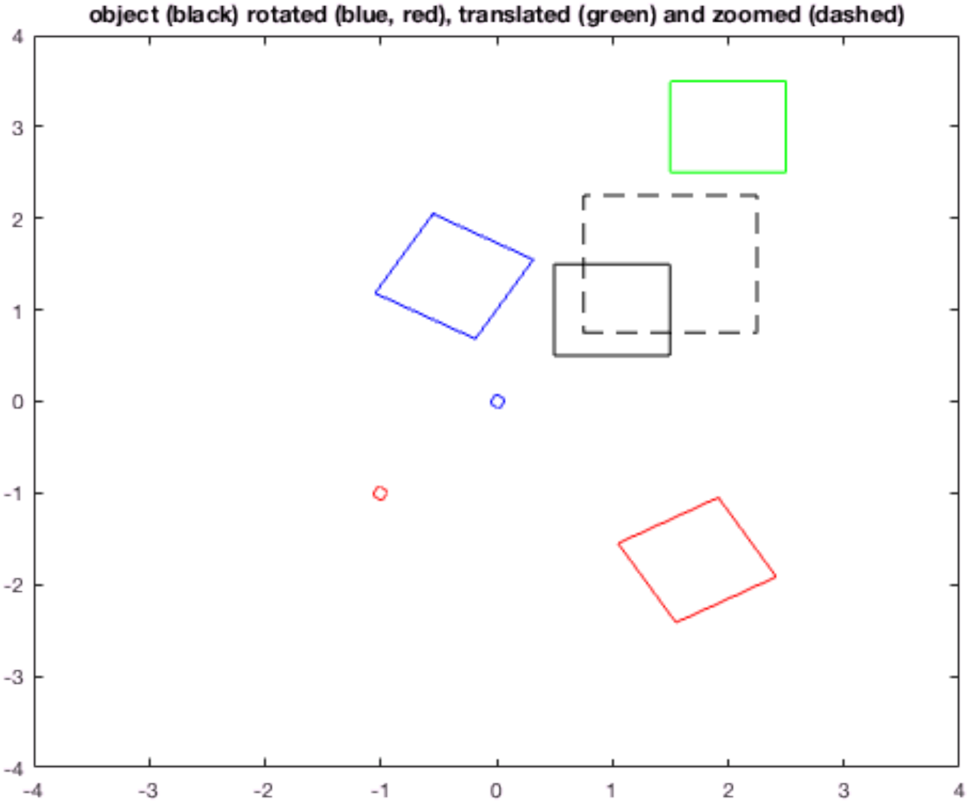
rotate 2D object around specified point

```
g = gOrig; % get backup copy

% specify center of rotation in figure coordinate system
cr = [-1; -1];

% specify CCW rotation, use degrees here, so use sind, cosd
th = -60;

% move object to new coordinate system
% where center of rotation is at figure origin
t = eye(3,3);
t(1,3) = -cr(1,1);
t(2,3) = -cr(2,1);
g = t * g;

% rotate object
r = [cosd(th) -sind(th) 0
    sind(th) cosd(th)   0
       0       0        1];
g = r * g;

% translate so center of rotation back to original location
t = eye(3,3);
t(1,3) = cr(1,1);
t(2,3) = cr(2,1);
g = t * g;

% plot rotated object and o at center of rotation
hold on
plot(g(1,:),g(2,:),'r',cr(1,1),cr(2,1),'ro')
axis([-am am -am am])
```

zoom 2D scene on origin

```
g = gOrig; % get backup copy
```

```
s = 1.5; % zoom factor, <1 zoom out, >1 zoom in
g = s * g; % s is a scalar so * and .* do same thing

hold on
plot(g(1,:),g(2,:),'k--')
axis([-am am -am am])
```

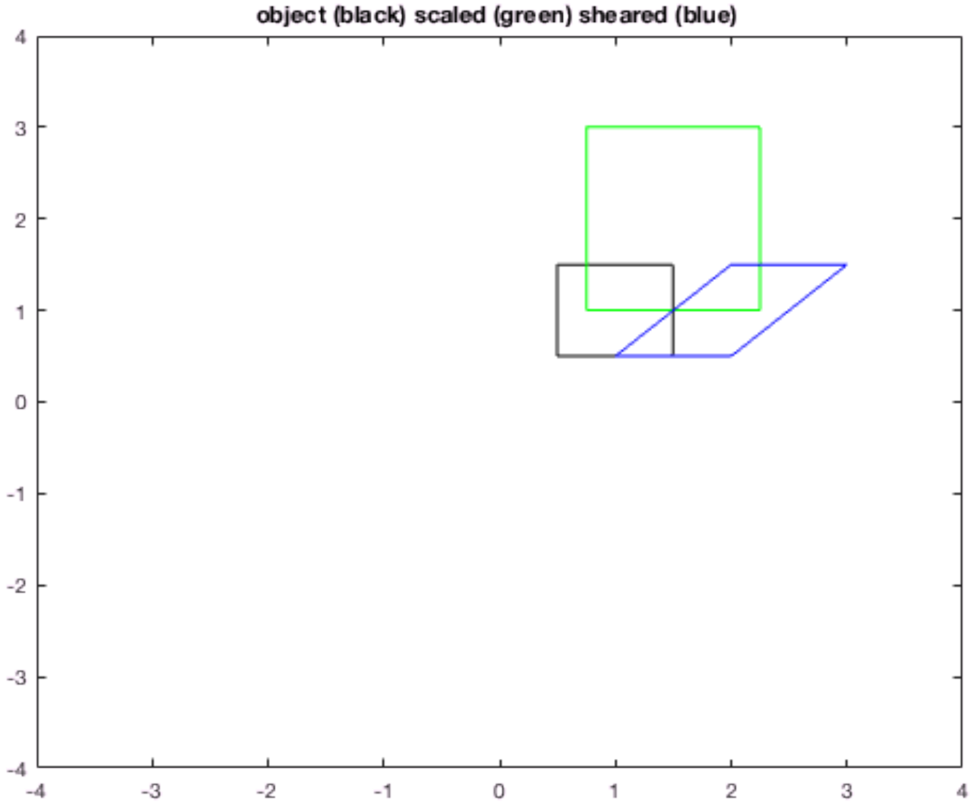


scale object

```
g = gOrig; % get backup copy

sx = 1.5; % scale factor for x direction
sy = 2; % scale factor for y direction

% scale
s = eye(3,3);
s(1,1) = sx;
s(2,2) = sy;
g = s * g;

figure(2)
plot(gOrig(1,:),gOrig(2,:),'k')
title('object (black) scaled (green) sheared (blue) ')
am = 4;
axis([-am am -am am])
hold on
plot(g(1,:),g(2,:),'g')
```

shear object

```
g = gOrig; % get backup copy

shx = 1; % shear factor for x direction
shy = 0; % shear factor for y direction

% shear
sh = eye(3,3);
sh(1,2) = shx;
sh(2,1) = shy;
g = sh * g;

hold on
plot(g(1,:),g(2,:),'b')
axis([-am am -am am])
```


object (black) scaled (green) sheared (blue)