Matlab Practical III: Graphics and Matrices

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Handle Graphics

more on
help plot
fplot(’sin’, [0 2*pi])
fplot(’sin’, [0 2*pi], ’b*’)    % plot styles
fplot(’sin’, [0 2*pi], ’b-’)
fplot(’sin’, [0 2*pi], ’b--’)
fplot(’sin’, [0 2*pi], ’b:’)
fplot(’sin’, [0 2*pi], ’bˆ’)
% adjust settings from the figure window
% click on the sine graph
get(gco)
delete(gco)

fplot(’sin’, [0 2*pi])
hold on
fplot(’cos’, [0 2*pi])
% click on the sine graph
set(gco, ’linewidth’, 4)
% click on the cosine graph
set(gco, ’linewidth’, 4, ’color’, ’r’)
txt = gtext(’sin(x)’)
set(txt, ’fontsize’, 18)
txt = gtext(’cos(x)’)
set(txt, ’fontsize’, 16)
title(’trig functions’, ’fontsize’, 20, ...
      ’fontname’, ’times’)

% print from FILE menu in the figure window
print -deps2  myfig.eps   % print b/w postscript
print -depsc2 mycfig.eps  % print color postscript
close

hist(randn(1000,1))       % chunky: 10 bins
hist(randn(100000,1),50)  % better with 50 bins
```

```
% Three Dimensional Data

z = peaks;            % peaks: built-in data set
size(z)
surf(z)
surf(peaks(100))
view([90 0])          % az/el: side view
view([0 90])          % az/el: top view
view([-37.5 30])      % az/el: standard view
rotate3d              % manual rotation
colorbar
shading interp
shading flat
colormap(cool)
colormap(hot)
colormap(jet)         % default colormap
close

surf(peaks)
surfc(peaks)
mesh(peaks)
meshc(peaks)
pcolor(peaks)

contour(peaks)
contourf(peaks)
help contour          % contour options
contour(peaks, 20)    % show more contours
contourf(peaks, 20)
[C,h] = contour(peaks);   % label contours
clabel(C,h)
[C,h] = contour(peaks, [1 2 3]);  % 3 contours
clabel(C,h)

for j=10:10:150, mesh(peaks(j)), disp(j), end
for j=10:10:150, mesh(peaks(j)), drawnow, end

help movie            % movie maker
help guide            % gui builder
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matrices, sparse matrices, matrix graphics

A = randn(512); b = randn(512,1);
length(A)
size(A)
size(randn(5,10),1)
size(randn(5,10),2)
tic
toc
tic, x1 = inv(A)*b; toc    % solve linear system Ax = b for x
tic, x2 = A\b; toc         % backslash is better, faster
norm(x1-x2)

A = randn(800);
b = randn(800,1);
tic; x = A\b; toc          % dense solve
tic; x = triu(A)\b; toc    % triangular solve
spy(triu(A))
```

```
A = -diag(ones(49,1),1);
A(1:5,1:5)
A = A - diag(ones(49,1),-1) + 2*eye(50);
A(1:5,1:5)
spy(A)
Ai = inv(A);
spy(Ai)
spy(abs(Ai)>1)
spy(abs(Ai)>5)
spy(abs(Ai)>10)
surf(Ai)
view(0,-90)
contour(Ai)
view(0,-90)
clear

A = 2*eye(600) - diag(ones(599,1),1) - diag(ones(599,1),-1);
B = sparse(A);
whos
help sparse
C = full(B);
tic; A*A; toc
tic; B*B; toc
clear

help spdiags
help speye
A = spdiags(ones(500,1)*[-1 2 -1], [-1 0 1], 500, 500);
A(1:5, 1:5)
full(A(1:5,1:5))
b = ones(500,1);
tic; x = A\b; toc               % solve a linear system: sparse
tic; x = full(A)\b; toc         % solve a linear system: full

type sparse_ex                  % display file sparse_ex.m
sparse_ex                       % run the program in sparse_ex.m
A = randn(7)
eig(A)                          % eigenvalues
[V,D] = eig(A+A')               % eigenvalues with eigenvectors
[E,indx] = sort(abs(diag(D)))   % sort eigenvalues
V = V(:,indx)                   % reorder eigenvectors


help sparfun
help eigs
help svds
```

the file sparse_ex.m

```
for j= 1:10
   A = spdiags(randn(2^j,5), [-2:2], 2^j, 2^j);
   b = randn(2^j,1);
   tic
   x_sparse = A\b;
   s_time(j) = toc;
   A = full(A);
   tic
   x_full = A\b;
   f_time(j) = toc;
   fprintf('dimension: %6d   difference: %10.7e\n', ...
      2^j, norm(x_sparse-x_full));
end
clf, subplot(1,2,1)
loglog(2.^[1:10], s_time, 'b-'), hold on
loglog(2.^[1:10], f_time, 'r-')
xlabel('dimension'), ylabel('time, sec')
subplot(1,2,2)
plot(2.^[1:10], s_time, 'b-'), hold on
plot(2.^[1:10], f_time, 'r-')
xlabel('dimension'), ylabel('time, sec')
```