```
%make a scale-free network in MATLAB
clear
nodes=100; %number of nodes
bias=2; %degree of rich-get-richer bias
links=500; %number of bi-directional connections(edges)
net=zeros(nodes,nodes); %make the net with zero connections

%every node starts with at least one random connection
for n=1:nodes %walkthru with each node
    x=ceil(rand*nodes); %pick a random other node
    if x==n %no self-connections
        x=ceil(rand*nodes); %come up with a new x
    end
    net(n,x)=1; %make it a bi-directional connection
    net(x,n)=1; %make it a bi-directional connection
end
net=net.*((eye(nodes,nodes)-1)*-1);%remove any self-
connections

%randomly seed a small handful of additional connections
%so that not every node has equal "wealth"
for n=1:ceil(nodes/10) %pick about 10% of them
    x=ceil(rand*nodes);
    y=ceil(rand*nodes);
    if x==y %no self-connections
        y=ceil(rand*nodes); %come up with a new x
    end
    net(x,y)=1; %make it a bi-directional connection
    net(y,x)=1; %make it a bi-directional connection
end
net=net.*((eye(nodes,nodes)-1)*-1);%remove any self-connections

%nodes with more connections tend to get more new connections
while nnz(net)/2<links % keep going until links is reached
    %probability that a node gets a new connection
    %gets higher when it already has more connections
    PEdge=sum(net)/nodes;%calculate relative number of conn's
    PEdge=PEdge.^bias/sum(PEdge.^bias);
     %convert relative number of connections to biased prob's
    choice=rand;
    criterion=0;
    for nn=1:nodes %pick one row based on probs
        criterion=criterion+PEdge(nn);
          if choice<criterion
             choice=nn;
             break %break out of loop
          end
```

```
    end
    for nnn=1:10000 %do this many times to make sure you
                    %find an unfilled connection
        i=choice; %place a link in the chosen row
        j=ceil(rand*nodes); %connect it to a random node
        if net(i,j)==0  %if the chosen link is blank
           net(i,j)=1; %then fill it in
           net(j,i)=1; %bi-drectionally
           break       %and then break out of the for loop
        end
    end %otherwise, keep looking for an unfilled connection
end

%make sure no self-connections
net=net.*((eye(nodes,nodes)-1)*-1);
%make sure graph undirected, i.e., all connections bi-
directional
net=(net+net')>0;

%histogram of connections
subplot(1,3,1)
hist(sum(net'),10)
[x,n]=hist(sum(net'),10);
%loglog histogram of connections
subplot(1,3,2)
loglog(n,x,'*k')
axis([1 100 1 100])
%graph of connections
N = length(net);
theta = linspace(0,2*pi,N)';
xy = [cos(theta) sin(theta)];
subplot(1,3,3)
gplot(net,xy)
```

Results should look a bit like this: