```
% Multigrid
L  = 3;
h  = 1/2^L;
hx = 1;
ii = 50000;
w  = 0.5;
% Step 1 Weighted Jacobi & Gasuss-Seidel Smoother

% Weighter Jacobi

u     = zeros(hx/h+1,hx/h+1);
u_w_j = Smoother_Weighted_Jaco(h,hx,u,ii,w);
u_g_s = Smoother_Gauss_Seidel1(h,hx,u,ii);

ii1   = 10;
ii2   = 11;
ii3   = 12;
ii4   = 13;

u_g_s_1 = Smoother_Gauss_Seidel1(h,hx,u,ii1);
u_g_s_2 = Smoother_Gauss_Seidel1(h,hx,u,ii2);
u_g_s_3 = Smoother_Gauss_Seidel1(h,hx,u,ii3);
u_g_s_4 = Smoother_Gauss_Seidel1(h,hx,u,ii4);

k1_k  = max(max((u_g_s_4-u_g_s_3)));
k_k1  = max(max((u_g_s_3-u_g_s_2)));
k1_k2 = max(max((u_g_s_2-u_g_s_1)));
disp('The convergence rate is:');
q     = log(abs(k1_k/k_k1))/log(abs(k_k1/k1_k2))


% Plot the error for h = 1/16
figure(1);
L  = 4;
h  = 1/2^L;
u     = zeros(hx/h+1,hx/h+1);
err_lst =  Smoother_Gauss_Seidel_alpha(h,hx,u,ii);
subplot(2,2,1);
plot([3:3:42],err_lst)
title('h=1/16 error plot')


% Get Step number for h=1/4
L  = 2;
h  = 1/2^L;
u  = zeros(hx/h+1,hx/h+1);
disp('The step required to get h^2 error h = 1/4 is:');
nstep1 =  Smoother_Gauss_Seidel_belta(h,hx,u,ii)


% Get Step number for h = 1/128
L  = 7;
```

```
h  = 1/2^L;
u  = zeros(hx/h+1,hx/h+1);
disp('The step required to get h^2 error h = 1/128 is:');
nstep2 =  Smoother_Gauss_Seidel_belta(h,hx,u,ii)

L  = 1;
h  = 1/2^L;
u  = zeros(hx/h+1,hx/h+1);
nstep3 =  Smoother_Gauss_Seidel_belta(h,hx,u,ii);

L  = 6;
h  = 1/2^L;
u  = zeros(hx/h+1,hx/h+1);
nstep4 =  Smoother_Gauss_Seidel_belta(h,hx,u,ii);

subplot(2,2,2);
plot([1 2 6 7],[nstep3 nstep1 nstep4 nstep2],"o");
xlabel('L')
ylabel('*# steps')
title('# of steps r.p.t L ')

% Choose a random initial guess and plot the error in the first three
 steps
L  = 4;
h  = 1/2^L;
u  = zeros(hx/h+1,hx/h+1);
[err_lst, rando] =  Smoother_Gauss_Seidel_gamma(h,hx,u,ii);
subplot(2,2,3);
plot([1 2 3],err_lst);
title('random start')

% Step 2 Two-Grid Method

figure(2);
subplot(2,1,1);
L  = 2;
h  = 1/2^L;
hx = 1;
u  = zeros(hx/h+1,hx/h+1);
u_g_s_1 = Smoother_Gauss_Seidel1(h,hx,u,10000);
J = 2;
mu = 10;
f  = h^2*ones((hx/h+1));
err_lst1 = [];
for i =1:10
    u = two_grid(u,f,J,mu);
    err_lst1 = [err_lst1 abs(max(max(u(2:end-1,2:end-1)-u_g_s_1)))];
end
plot(err_lst1);
title('h=1/4');

subplot(2,1,2);
L  = 7;
h  = 1/2^L;
```

```
hx = 1;
u  = zeros(hx/h+1,hx/h+1);
u_g_s_2 = Smoother_Gauss_Seidel1(h,hx,u,10000);
J = 2;
mu = 10;
f  = h^2*ones((hx/h+1));
err_lst2 = [];
for i =1:10
    u = two_grid(u,f,J,mu);
    err_lst2 = [err_lst2 abs(max(max(u(2:end-1,2:end-1)-u_g_s_2)))];
end

plot(err_lst2);
title('h=1/128');

% Step 3 Vcycle
figure(3)
L  = 7;
h  = 1/2^L;
mu = 10;
hx = 1;
u2  = zeros(hx/h+1,hx/h+1);
error =1;
tol = 0.1*h^2;
J = 4;
f  = h^2*ones((hx/h+1));
u_g_s_2 = Smoother_Gauss_Seidel1(h,hx,u2,10000);
kk =1;
error_lst = [];
while error > tol
    u2 = Vcycle1(u2,J);
    kk = kk+1;
    error = abs(max(max(u2(2:end-1,2:end-1)-u_g_s_2)));
    error_lst = [error_lst error];
end
plot(error_lst);
title("V cycle")



% Multigrid on Hierarchical Grids
% Step 1
[node,elem] = circlemesh(0,0,1,0.25);
J = 4;
HB_cell       = cell(J,1);
bdNode_cell   = cell(J,1);
freeNode_cell = cell(J,1);

for i = 1:J
    [node,elem,~,HB] = uniformrefine(node,elem);
    HB_cell{i}       = double(HB);
    [bdNode,bdEdge,isBdNode,isBdElem]       = findboundary(elem);
    freeNode                                = find(~isBdNode);
```

```
    bdNode_cell{i}                          = bdNode;
    freeNode_cell{i}                        = freeNode;
end
%[node,elem,~,HB] = uniformrefine(node,elem);
%[elem,HB,~] = uniformcoarsenred(elem);
figure(4);
% Step 2
% showmesh(node,elem);
N                                       =  length(node);
mid1                                    =
 (node(elem(:,2),:)+node(elem(:,3),:))/2;
mid2                                    =
 (node(elem(:,3),:)+node(elem(:,1),:))/2;
mid3                                    =
 (node(elem(:,1),:)+node(elem(:,2),:))/2;
area0                                   = assembling_area(node,elem);
bt1                                     =
 area0.*(f1(mid2)+f1(mid3))/6;
bt2                                     =
 area0.*(f1(mid3)+f1(mid1))/6;
bt3                                     =
 area0.*(f1(mid1)+f1(mid2))/6;
b                                       = accumarray(elem(:),
[bt1;bt2;bt3],[N 1]);

[bdNode,bdEdge,isBdNode,isBdElem]       = findboundary(elem);
freeNode                                = find(~isBdNode);
u                                       = zeros(N,1);
u                                       = rand(N,1);
u(bdNode)                               = g_D(node(bdNode,:));
A                                       = assembling(node,elem);

A_cell                                  = cell(J,1);
A_cell{J}                               = A;
A_new                                   = A;
A_tril                                  = cell(J,1);
A_triu                                  = cell(J,1);
A_tril{J}                               = tril(A_new);
A_triu{J}                               = triu(A_new);
A_pro                                   = cell(J,1);
A_res                                   = cell(J,1);

for i = J-1:-1:2
    pro                                 = prolongation_HB1(HB_cell{i
+1});
    res                                 = pro';
    A_pro{i+1}                          = pro;
    A_res{i+1}                          = res;
    A_new                               = res*A_new*pro;
    A_cell{i}                           = A_new;
    A_tril{i}                           = tril(A_new);
    A_triu{i}                           = triu(A_new);
end
```

```
i =1;
pro                                     = prolongation_HB1(HB_cell{i
+1});
res                                     = pro';
A_pro{2}                                = pro;
A_res{2}                                = res;
A_cell{1}                               = res*A_new*pro;

r_list = [];

for i = 1:30
    r                                       = b - A*u;
    r_list                                  = [r_list max(abs(r))];
    e                                       =
 Vcycle(r,J,A_cell,A_tril,A_triu,A_pro,A_res,bdNode_cell,freeNode_cell);
    u                                       = u+e;
end
semilogy(r_list)
title("semilogy plot for rand initial condition J = 4")









figure(5);

time_list_three = [];
time_list_four = [];
time_list_five = [];
time_list_six = [];

for J = 3:6

    [node,elem] = circlemesh(0,0,1,0.25);
    HB_cell       = cell(J,1);
    bdNode_cell   = cell(J,1);
    freeNode_cell = cell(J,1);

    for i = 1:J
        [node,elem,~,HB] = uniformrefine(node,elem);
```

```
        HB_cell{i}       = double(HB);
        [bdNode,bdEdge,isBdNode,isBdElem]       = findboundary(elem);
        freeNode                                = find(~isBdNode);
        bdNode_cell{i}                          = bdNode;
        freeNode_cell{i}                        = freeNode;
    end
    %[node,elem,~,HB] = uniformrefine(node,elem);
    %[elem,HB,~] = uniformcoarsenred(elem);
    % Step 2
    % showmesh(node,elem);
    N                                       =  length(node);
    mid1                                    =
 (node(elem(:,2),:)+node(elem(:,3),:))/2;
    mid2                                    =
 (node(elem(:,3),:)+node(elem(:,1),:))/2;
    mid3                                    =
 (node(elem(:,1),:)+node(elem(:,2),:))/2;
    area0                                   =
 assembling_area(node,elem);
    bt1                                     =
 area0.*(f1(mid2)+f1(mid3))/6;
    bt2                                     =
 area0.*(f1(mid3)+f1(mid1))/6;
    bt3                                     =
 area0.*(f1(mid1)+f1(mid2))/6;
    b                                       = accumarray(elem(:),
[bt1;bt2;bt3],[N 1]);

    [bdNode,bdEdge,isBdNode,isBdElem]       = findboundary(elem);
    freeNode                                = find(~isBdNode);
    u                                       = zeros(N,1);
    u                                       = rand(N,1);
    u(bdNode)                               = g_D(node(bdNode,:));
    A                                       = assembling(node,elem);

    A_cell                                  = cell(J,1);
    A_cell{J}                               = A;
    A_new                                   = A;
    A_tril                                  = cell(J,1);
    A_triu                                  = cell(J,1);
    A_tril{J}                               = tril(A_new);
    A_triu{J}                               = triu(A_new);
    A_pro                                   = cell(J,1);
    A_res                                   = cell(J,1);

    for i = J-1:-1:2
        pro                                 =
 prolongation_HB1(HB_cell{i+1});
        res                                 = pro';
        A_pro{i+1}                          = pro;
        A_res{i+1}                          = res;
        A_new                               = res*A_new*pro;
        A_cell{i}                           = A_new;
        A_tril{i}                           = tril(A_new);
```

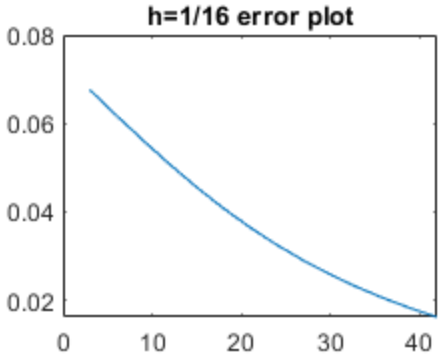
```
        A_triu{i}                                   = triu(A_new);
    end

    i =1;
    pro                                             =
prolongation_HB1(HB_cell{i+1});
    res                                             = pro';
    A_pro{2}                                        = pro;
    A_res{2}                                        = res;
    A_cell{1}                                       = res*A_new*pro;


    for i = 1:30
        tic;
        r                                               = b - A*u;
        r_list                                          = [r_list
max(abs(r))];
        e                                               =
Vcycle(r,J,A_cell,A_tril,A_triu,A_pro,A_res,bdNode_cell,freeNode_cell);
        u                                               = u+e;
        toc;
        t1                                              = toc;
        if J == 3
            if i == 1
                time_list_three = [t1];
            else
                time_list_three = [time_list_three
t1+time_list_three(end)];
            end
        elseif J == 4
            if i == 1
                time_list_four = [t1];
            else
                time_list_four = [time_list_four
t1+time_list_four(end)];
            end
        elseif J == 5
            if i == 1
                time_list_five = [t1];
            else
                time_list_five = [time_list_five
t1+time_list_five(end)];
            end
        else
            if i == 1
                time_list_six = [t1];
            else
                time_list_six = [time_list_six t1+time_list_six(end)];
            end
        end
    end

end
```

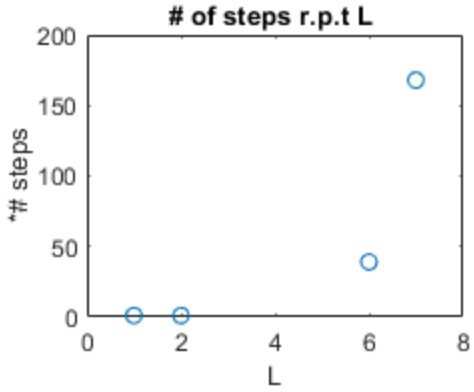
```
plot(time_list_three)
hold on;
plot(time_list_four)
plot(time_list_five)
plot(time_list_six)
legend("three","four","five","six");
title("time vers J = 3,4,5,6")
```

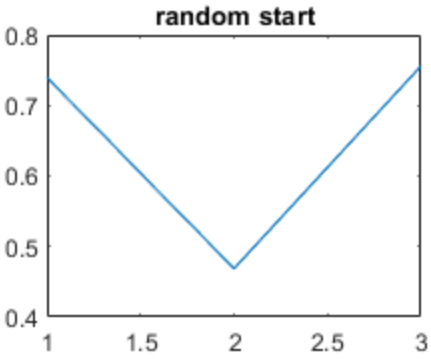
```
The convergence rate is:

q =

    1.0029

The step required to get h^2 error h = 1/4 is:

nstep1 =

     1

The step required to get h^2 error h = 1/128 is:

nstep2 =

   168

Elapsed time is 0.004084 seconds.
Elapsed time is 0.002651 seconds.
Elapsed time is 0.001850 seconds.
Elapsed time is 0.001723 seconds.
Elapsed time is 0.001842 seconds.
Elapsed time is 0.001757 seconds.
Elapsed time is 0.001651 seconds.
Elapsed time is 0.001638 seconds.
Elapsed time is 0.001623 seconds.
Elapsed time is 0.001628 seconds.
Elapsed time is 0.001624 seconds.
Elapsed time is 0.001629 seconds.
Elapsed time is 0.001636 seconds.
Elapsed time is 0.001785 seconds.
Elapsed time is 0.001803 seconds.
Elapsed time is 0.001685 seconds.
Elapsed time is 0.001632 seconds.
Elapsed time is 0.001642 seconds.
Elapsed time is 0.001646 seconds.
Elapsed time is 0.001633 seconds.
Elapsed time is 0.001703 seconds.
Elapsed time is 0.001635 seconds.
Elapsed time is 0.001643 seconds.
Elapsed time is 0.001828 seconds.
Elapsed time is 0.001799 seconds.
Elapsed time is 0.001668 seconds.
Elapsed time is 0.001634 seconds.
Elapsed time is 0.001656 seconds.
```

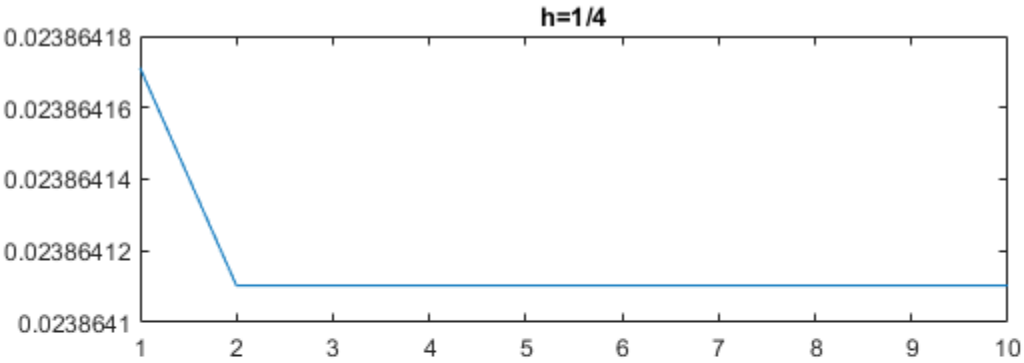
```
Elapsed time is 0.001638 seconds.
Elapsed time is 0.001626 seconds.
Elapsed time is 0.006761 seconds.
Elapsed time is 0.006879 seconds.
Elapsed time is 0.006791 seconds.
Elapsed time is 0.006839 seconds.
Elapsed time is 0.006527 seconds.
Elapsed time is 0.006977 seconds.
Elapsed time is 0.006699 seconds.
Elapsed time is 0.006786 seconds.
Elapsed time is 0.006801 seconds.
Elapsed time is 0.006517 seconds.
Elapsed time is 0.006911 seconds.
Elapsed time is 0.006635 seconds.
Elapsed time is 0.006825 seconds.
Elapsed time is 0.006912 seconds.
Elapsed time is 0.006590 seconds.
Elapsed time is 0.006946 seconds.
Elapsed time is 0.006575 seconds.
Elapsed time is 0.006968 seconds.
Elapsed time is 0.006522 seconds.
Elapsed time is 0.006595 seconds.
Elapsed time is 0.006833 seconds.
Elapsed time is 0.006630 seconds.
Elapsed time is 0.007057 seconds.
Elapsed time is 0.006563 seconds.
Elapsed time is 0.006646 seconds.
Elapsed time is 0.006691 seconds.
Elapsed time is 0.006653 seconds.
Elapsed time is 0.006839 seconds.
Elapsed time is 0.006481 seconds.
Elapsed time is 0.006611 seconds.
Elapsed time is 0.052863 seconds.
Elapsed time is 0.043260 seconds.
Elapsed time is 0.041578 seconds.
Elapsed time is 0.040264 seconds.
Elapsed time is 0.040323 seconds.
Elapsed time is 0.044693 seconds.
Elapsed time is 0.040712 seconds.
Elapsed time is 0.040461 seconds.
Elapsed time is 0.043288 seconds.
Elapsed time is 0.039941 seconds.
Elapsed time is 0.040138 seconds.
Elapsed time is 0.043102 seconds.
Elapsed time is 0.040325 seconds.
Elapsed time is 0.040729 seconds.
Elapsed time is 0.043196 seconds.
Elapsed time is 0.041186 seconds.
Elapsed time is 0.040893 seconds.
Elapsed time is 0.042690 seconds.
Elapsed time is 0.040898 seconds.
Elapsed time is 0.040340 seconds.
Elapsed time is 0.042746 seconds.
Elapsed time is 0.041834 seconds.
```

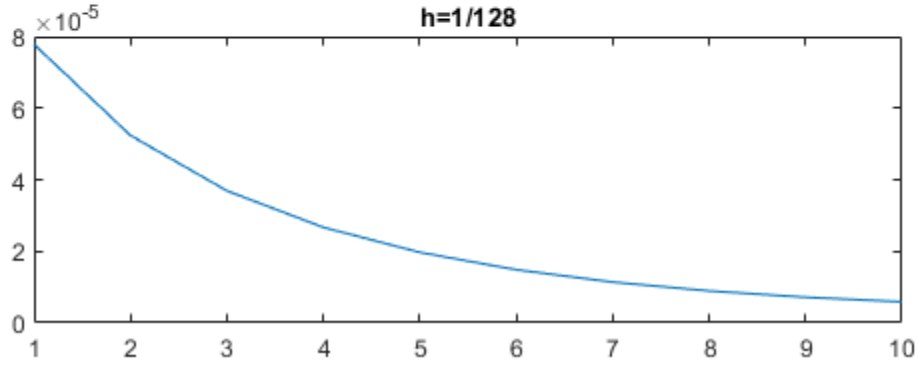
```
Elapsed time is 0.041877 seconds.
Elapsed time is 0.045833 seconds.
Elapsed time is 0.041191 seconds.
Elapsed time is 0.041118 seconds.
Elapsed time is 0.049191 seconds.
Elapsed time is 0.042136 seconds.
Elapsed time is 0.041092 seconds.
Elapsed time is 0.042765 seconds.
Elapsed time is 0.216661 seconds.
Elapsed time is 0.216816 seconds.
Elapsed time is 0.220180 seconds.
Elapsed time is 0.216748 seconds.
Elapsed time is 0.225539 seconds.
Elapsed time is 0.220850 seconds.
Elapsed time is 0.217506 seconds.
Elapsed time is 0.216671 seconds.
Elapsed time is 0.217203 seconds.
Elapsed time is 0.217105 seconds.
Elapsed time is 0.216146 seconds.
Elapsed time is 0.216665 seconds.
Elapsed time is 0.216624 seconds.
Elapsed time is 0.220249 seconds.
Elapsed time is 0.216292 seconds.
Elapsed time is 0.216392 seconds.
Elapsed time is 0.216327 seconds.
Elapsed time is 0.217041 seconds.
Elapsed time is 0.216114 seconds.
Elapsed time is 0.216506 seconds.
Elapsed time is 0.216168 seconds.
Elapsed time is 0.215696 seconds.
Elapsed time is 0.229557 seconds.
Elapsed time is 0.215593 seconds.
Elapsed time is 0.226841 seconds.
Elapsed time is 0.221719 seconds.
Elapsed time is 0.215650 seconds.
Elapsed time is 0.224660 seconds.
Elapsed time is 0.215888 seconds.
Elapsed time is 0.214939 seconds.
```

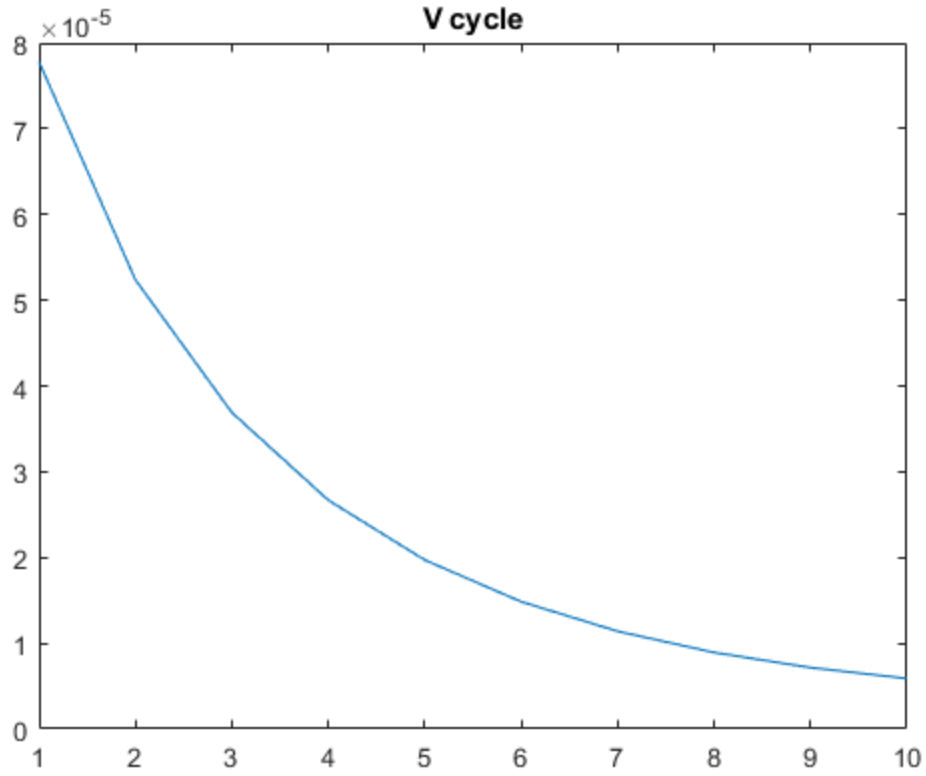

h=1/16 error plot
0.08
0.06
0.04
0.02
0
10
20
30
40

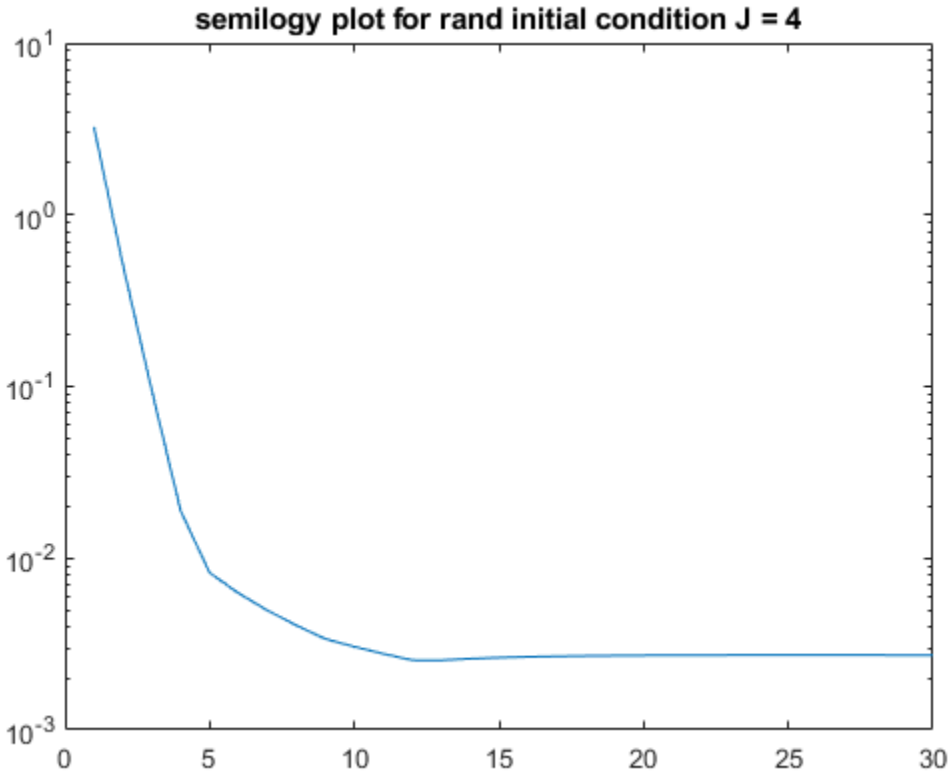

# of steps r.p.t L
200
150
100
50
0
*# steps
0
2
4
6
8
L

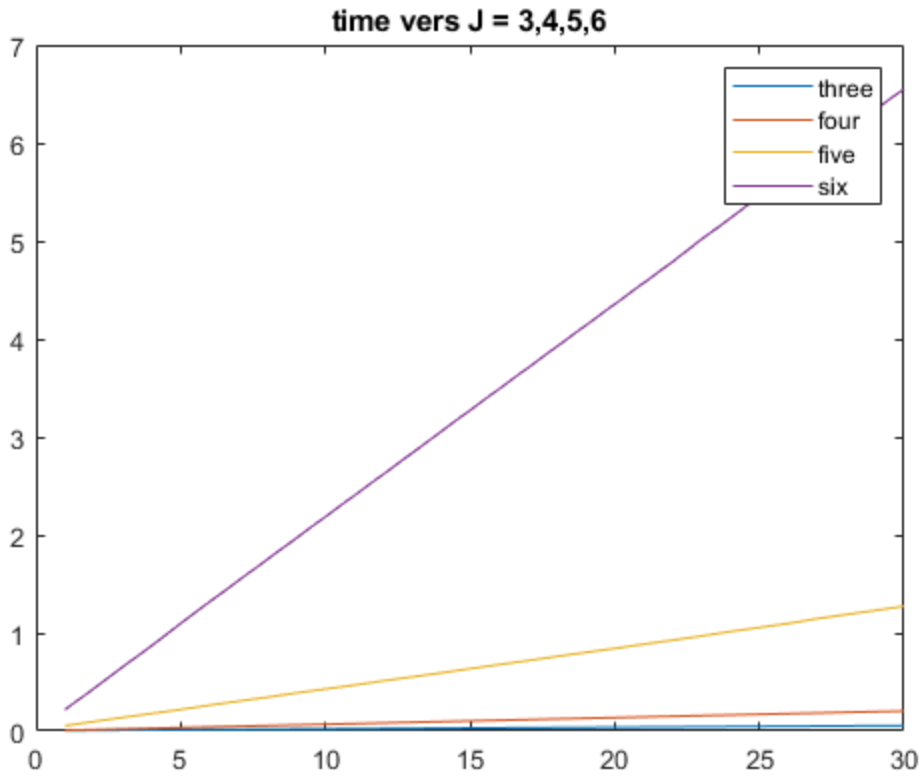

random start
0.8
0.7
0.6
0.5
0.4
1
1.5
2
2.5
3


h=1/4
0.02386418
0.02386416
0.02386414
0.02386412
0.0238641
1
2
3
4
5
6
7
8
9
10


h=1/128
×10-5
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10


V cycle
×10^-5
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
10


semilogy plot for rand initial condition J = 4
10^1
10^0
10^-1
10^-2
10^-3
0
5
10
15
20
25
30



Published with MATLAB® R2018b