CEE 6513 Computational Methods in Mechanics

Homework 4

Student: Yu-Chen Pan GTID: 903918558

Problem 1.

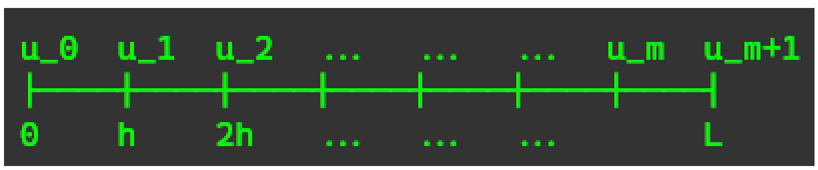
- Differential equation: $\frac{d^2u}{dx^2} = -1$

- Boundary Conditions: $u(0) = 0,\ \ u'(L) = 0$

- Exact solution:

$$\frac{d^2u}{dx^2} = -1,\ \frac{du}{dx} = -x + c_1,\ u = -\frac{x^2}{2} + c_1 x + c_2$$
$$u(0) = 0, \rightarrow\ c_2 = 0$$
$$u'(L) = 0, \rightarrow\ -L + c_1 = 0, \rightarrow\ c_1 = L$$
$$\rightarrow\ Exact\ solution: u = -\frac{x^2}{2} + Lx$$

- Second-order Finite Difference Method:



$\frac{1}{h^2}(u_{i-1} - 2u_i + u_{i+1}) = f(x_i)$, where $f(x_i) = -1, for\ i = 1, 2, \dots, m$

For Neumann boundary condition $u'(L) = 0$:

$$\begin{cases} \frac{1}{h^2}(u_m - 2u_{m+1} + u_{m+2}) = f(x_{m+1}) \\ \frac{1}{2h}(u_{m+2} - u_m) = 0 \end{cases}$$
$$\rightarrow \frac{1}{h}(u_m - u_{m+1}) = \frac{h}{2} f(x_{m+1})$$

Matrix form $Au = f$:

$$A = \frac{1}{h^2}\begin{bmatrix} h^2 & & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \dots & \dots & \dots & \\ & & & 1 & -2 & 1 \\ & & & & h & -h \end{bmatrix}$$

$$u = [u_0 \quad u_1 \quad u_2 \quad \dots \quad u_m \quad u_{m+1}]^T$$

$$f = \left[0 \quad f(x_1) \quad f(x_2) \quad \dots \quad f(x_m) \quad \frac{h}{2} f(x_{m+1})\right]^T$$

- Mesh size:

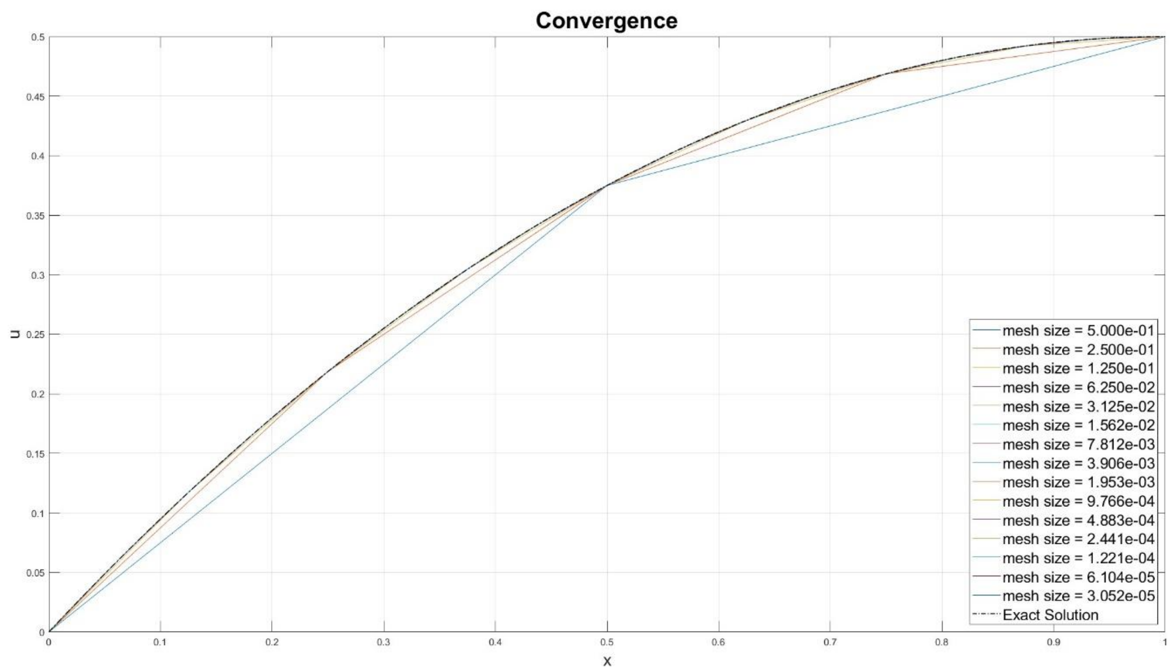
Let $m = 2^n - 1, \rightarrow m = 1, 3, 7, \dots$

$h = \frac{L}{m+1}$ (L = 1 for plotting) $, \rightarrow h = 0.5, 0.25, 0.125, \dots$

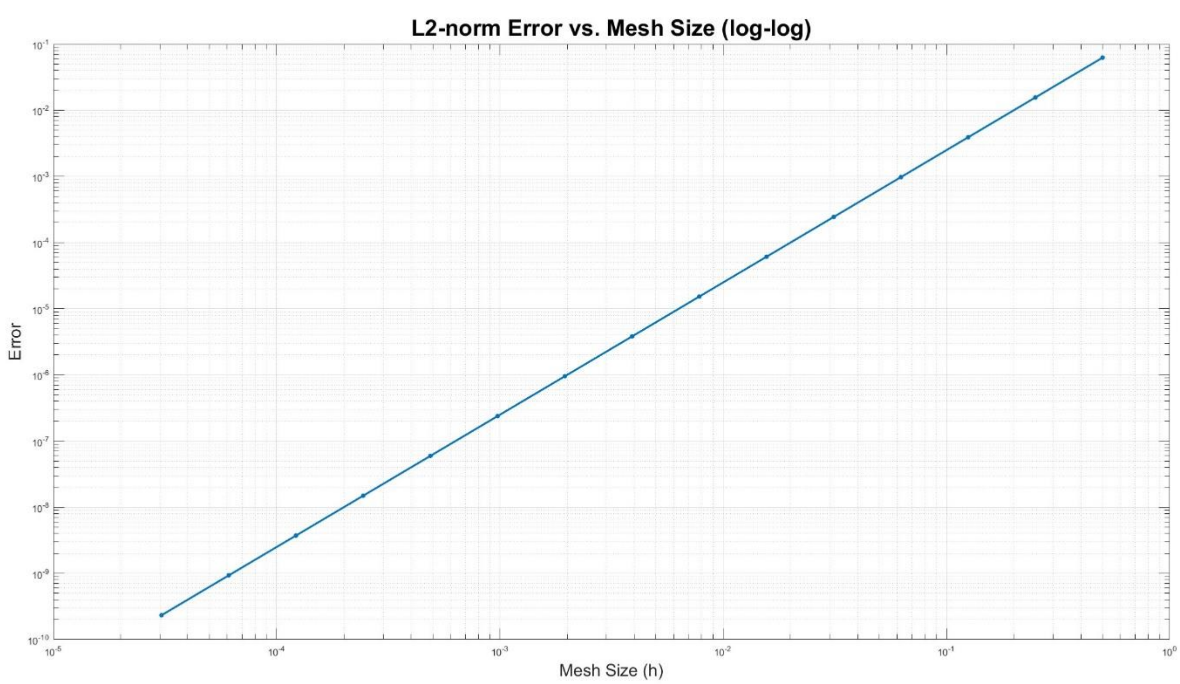
- Approximate solution:

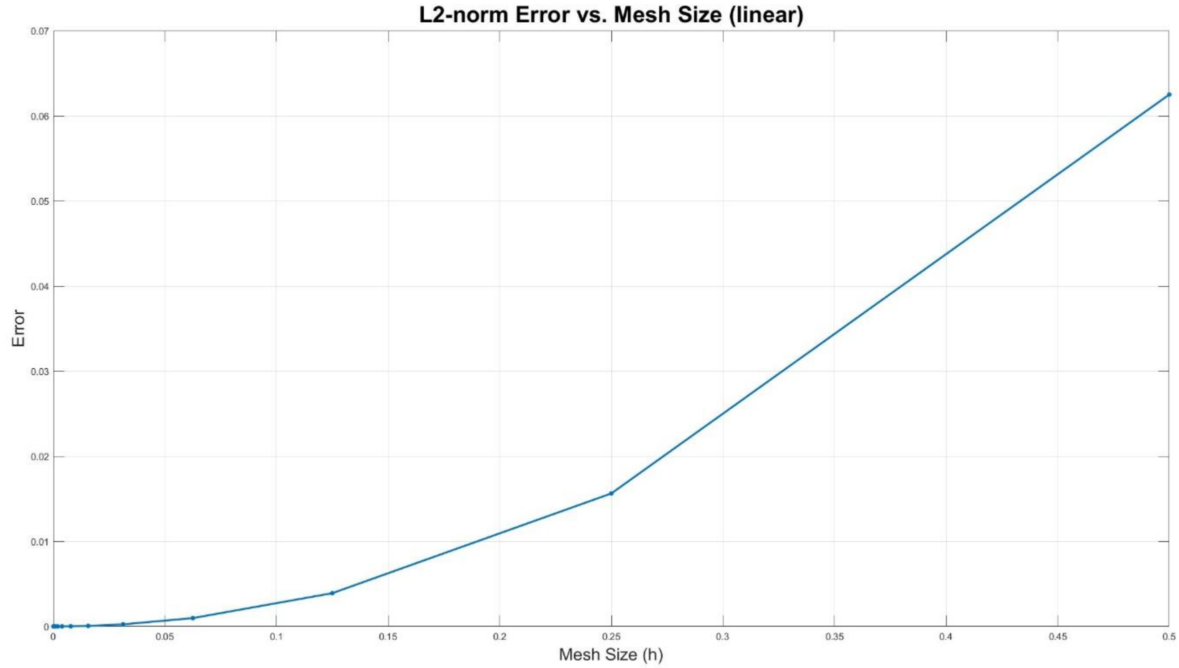
Mesh size	Coordinate	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u...
0.5	x	0	0.5	1															
	y (=u_i)	0	0.375	0.5															
0.25	x	0	0.25	0.5	0.75	1													
	y (=u_i)	0	0.21875	0.375	0.46875	0.5													
0.125	x	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1									
	y (=u_i)	0	0.11719	0.21875	0.30469	0.375	0.42969	0.46875	0.49219	0.5									
0.0625	x	0	0.0625	0.125	0.1875	0.25	0.3125	0.375	0.4375	0.5	0.5625	0.625	0.6875	0.75	0.8125	0.875	0.9375	1	
	y (=u_i)	0	0.06055	0.11719	0.16992	0.21875	0.26367	0.30469	0.3418	0.375	0.4043	0.42969	0.45117	0.46875	0.48242	0.49219	0.49805	0.5	
…	x	0	…							…									…
	y (=u_i)	0	…							…									…

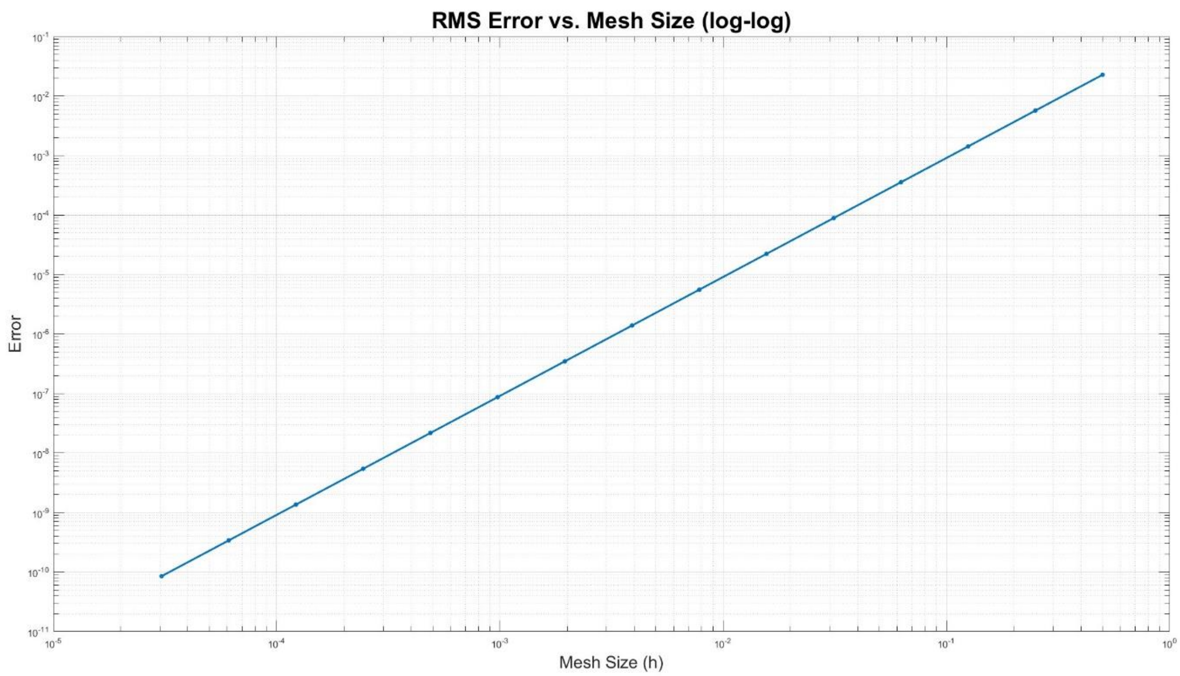
- Convergence:

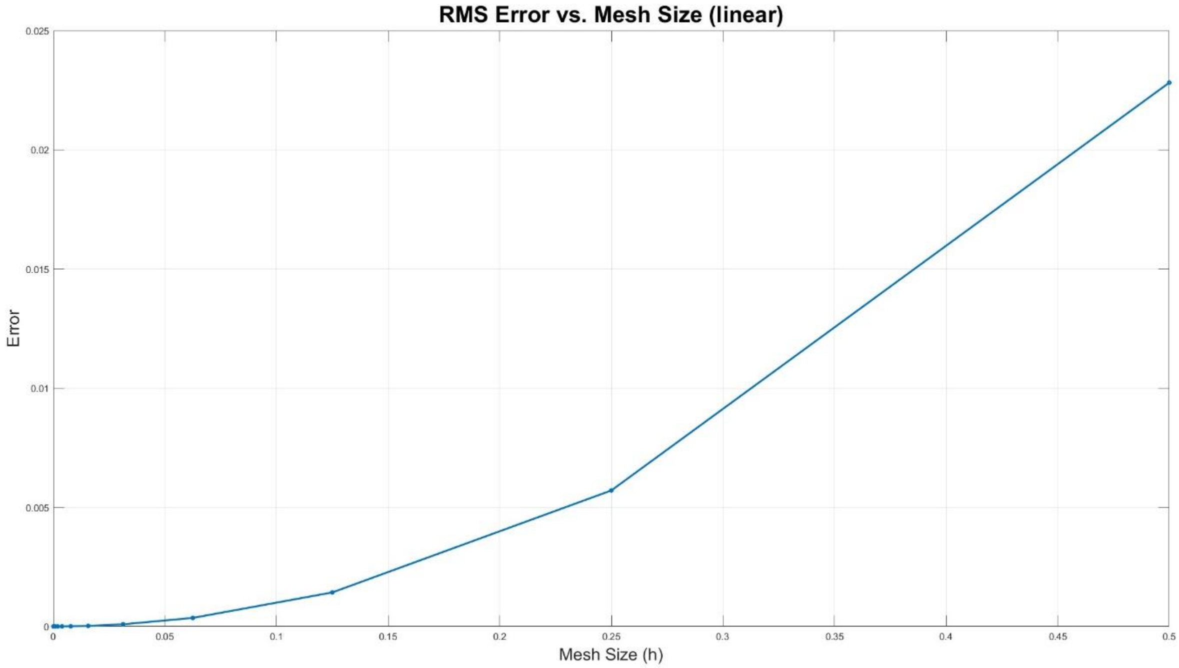


- Error:

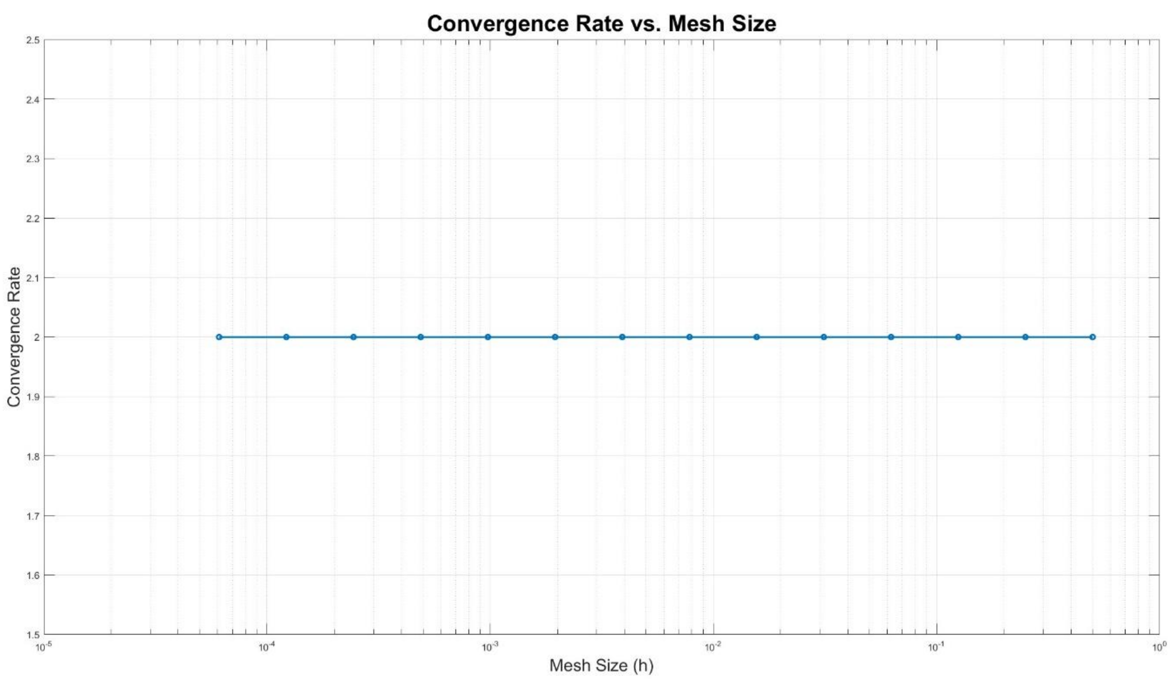



L2-norm Error vs. Mesh Size (linear)
Error
Mesh Size (h)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5


RMS Error vs. Mesh Size (log-log)
Error
Mesh Size (h)



- Convergence Rate:



```
%==============================================================================
%
%   Due 2023/10/25 23:59
%       Georgia Institute of Technology
%       CEE_6513
%       HW_4
%       Student: Yu-Chen Pan
%       GTID: 903918558
%
%==============================================================================
%
%   Problem_1:
%       -d^2(u)/dx^2 = 1
%       u(0) = 0
%       u'(L) = 0
%
%==============================================================================
%
%   Exact solution:
%       u(x) = (-x^2)/2 + L*x
%
%==============================================================================
%
%   Second-order Finite Difference Method:
%
%       u_0  u_1  u_2  ...  ...  ...  u_m  u_m+1
%       |----|----|----|----|----|----|----|
%       0    h    2h   ...  ...  ...       L
%
%       h^(-2)*(u_i-1 - 2*u_i + u_i+1) = f(x_i)
%
%       Au = B
%
%       A = [1    0
%            1   -2    1
%                 1   -2    1
%                      1   -2    1
%                      .    .    .
%                           .    .    .
%                                .    .    .
%                                     1   -2    1
%                                          1/h  -1/h]
%
%       u = [u_0    u_1    u_2    ...    u_m    u_m+1]^T
%
%       B = [0      f(x_1) f(x_2) ...    f(x_m) (h/2)*f(x_m+1)]^T
%
%==============================================================================

% Define problem parameters
L = 1;          % Length of the domain
n = 15;         % mesh refinements
f = -1;         % Source term

% Initialize arrays for mesh sizes, matrices, and solutions
m = zeros(n, 1);
h = zeros(n, 1);
```

```matlab
As = cell(1, n);
Bs = cell(1, n);
us = cell(1, n);

% Loop over mesh refinements to compute numerical solutions
for i = 1:n
    % Calculate number of internal nodes and mesh size
    m(i) = (2^(i) - 1);
    h(i) = L/(m(i) + 1);

    % Initialize matrices and vectors
    A = zeros(m(i) + 2);
    B = zeros(m(i) + 2, 1);

    % Set Dirichlet boundary condition at the left end
    A(1, 1) = 1;
    B(1) = 0;

    % Populate A matrix and B vector for internal nodes
    for j = 2:m(i)+1
        A(j, j - 1) = 1/(h(i)^2);
        A(j, j) = -2/(h(i)^2);
        A(j, j + 1) = 1/(h(i)^2);
        B(j) = f;
    end

    % Set Neumann boundary condition at the right end
    A(m(i) + 2, m(i) + 1) = 1/h(i);
    A(m(i) + 2, m(i) + 2) = -1/h(i);
    B(m(i) + 2) = (h(i)/2)*f;

    % Solve the system using sparse matrix
    A_sparse = sparse(A);
    u = A_sparse\B;

    % Store matrices and solution
    As{i} = A;
    Bs{i} = B;
    us{i} = u;
end

% Plot numerical solutions for different mesh sizes
figure;
for i = 1:n
    x = (0:h(i):L)';
    plot(x, us{i}, '-', 'DisplayName', sprintf('mesh size = %.3e', h(i)));
    hold on;
end

% Plot the exact solution
x_exact = linspace(0, L, 5000);
y_exact = (-x_exact.^2)/2 + x_exact;
plot(x_exact, y_exact, 'k:', 'LineWidth', 1, 'DisplayName', 'Exact Solution');

% Set plot properties for convergence plot
xlabel('x', 'FontSize', 18);
ylabel('u', 'FontSize', 18);
```

```
title('Convergence', 'FontSize', 24);
lgd = legend('show', 'Location','southeast');
fontsize(lgd,16,'points')
grid on;

% Compute L2-norm errors for each mesh size using a fine grid
num_fine_points = 5000;
x_fine = linspace(0, L, num_fine_points);
L2_n_errors = zeros(1, n);

for i = 1:n
    u_interpolated = interp1((0:h(i):L)', us{i}, x_fine, 'linear');
    y_exact_fine = (-x_fine.^2)/2 + x_fine;
    L2_n_errors(i) = norm(u_interpolated - y_exact_fine, 2) / norm(y_exact_fine, 2);
end

% Display L2-norm errors
disp('Errors for each mesh size:');
disp(L2_n_errors);

% Compute RMS errors for each mesh size using a fine grid
rms_errors = zeros(1, n);

for i = 1:n
    u_interpolated = interp1((0:h(i):L)', us{i}, x_fine, 'linear');
    y_exact_fine = (-x_fine.^2)/2 + x_fine;

    % RMS Error computation
    differences = u_interpolated - y_exact_fine;
    rms_errors(i) = sqrt(mean(differences.^2));
end

% Display RMS errors
disp('RMS Errors for each mesh size:');
disp(rms_errors);

% Compute and display convergence rates using RMS error
convergence_rates = zeros(1, n-1);
for i = 1:n-1
    convergence_rates(i) = log(rms_errors(i)/rms_errors(i+1)) / log(2);
end
disp('Convergence rates:');
disp(convergence_rates);

% Plot L2-norm error vs. mesh size (log-log)
figure;
loglog(h, L2_n_errors, '-o', 'LineWidth', 2, 'MarkerSize', 3);
xlabel('Mesh Size (h)', 'FontSize', 18);
ylabel('Error', 'FontSize', 18);
title('L2-norm Error vs. Mesh Size (log-log)', 'FontSize', 24);
grid on;

% Plot L2-norm error vs. mesh size (linear)
figure;
plot(h, L2_n_errors, '-o', 'LineWidth', 2, 'MarkerSize', 3);
xlabel('Mesh Size (h)', 'FontSize', 18);
```

```
ylabel('Error', 'FontSize', 18);
title('L2-norm Error vs. Mesh Size (linear)', 'FontSize', 24);
grid on;

% Plot RMS error vs. mesh size (log - log)
figure;
loglog(h, rms_errors, '-o', 'LineWidth', 2, 'MarkerSize', 3);
xlabel('Mesh Size (h)', 'FontSize', 18);
ylabel('Error', 'FontSize', 18);
title('RMS Error vs. Mesh Size (log-log)', 'FontSize', 24);
grid on;

% Plot RMS error vs. mesh size (linear)
figure;
plot(h, rms_errors, '-o', 'LineWidth', 2, 'MarkerSize', 3);
xlabel('Mesh Size (h)', 'FontSize', 18);
ylabel('Error', 'FontSize', 18);
title('RMS Error vs. Mesh Size (linear)', 'FontSize', 24);
grid on;

% Plot convergence rate vs. mesh size
figure;
semilogx(h(1:end-1), convergence_rates, '-o', 'LineWidth', 2, 'MarkerSize', 3);
xlabel('Mesh Size (h)', 'FontSize', 18);
ylabel('Convergence Rate', 'FontSize', 18);
title('Convergence Rate vs. Mesh Size', 'FontSize', 24);
ylim([1.9 2.1]);
grid on;
```