

Contents

■ Projection 함수 정의

```
clc;
clear;
close all;

% 주어진 함수 정의
f = @(x) (2*x).*sin(2*pi * x) + 3;
f = @(x) sin(pi*x)
% error 값 생성
n_values = [4, 8, 16, 32, 64]; % n 값을 다양하게 설정
error_values = zeros(size(n_values));

for k = 1:length(n_values)
    n = n_values(k); % 행 및 열의 개수
    x = linspace(0, 1, n);

    % Projection 계산
    projected_function = l2_projection_1D(x, f);

    % error 계산
    error = 0;
    for i = 1:n-1
        exact_integral = integral(@(x) abs(f(x) - projected_function(i)), x(i), x(i + 1)); % 두 함수의 차의 절댓값을 적분하여 에러를 계산
        error = error + exact_integral; % 에러를 누적
    end
    error_values(k) = error; % 누적된 에러를 저장

    % 결과 플롯
    x_values = linspace(0, 1, 1000); % 더 많은 점으로 그래프를 그리기 위해
    figure;
    plot(x_values, f(x_values), 'b', 'LineWidth', 2); hold on;
    plot(x, projected_function, 'r--', 'LineWidth', 2);
    legend('Original Function', 'Projected Function', 'Location', 'best');
    xlabel('x');
    ylabel('f(x)');
    title(['Projection and Original of function, n = ', num2str(n)]);
    grid on;
    hold off;
end

% error plot
figure;
loglog(n_values, error_values, '-o', 'LineWidth', 2);
xlabel('n');
ylabel('Error');
title('Convergence Error');
grid on;

% error rate
rate = -(log(error_values(2:end)) - log(error_values(1:end-1))) / log(2);
disp('error rate');
disp(rate)
```

Projection 함수 정의

```
function [projected_function] = l2_projection_1D(x, f)
    n = length(x);
    h = x(2) - x(1);

    % loadVector를 계산(GQ).
    loadVector = zeros(n, 1);
    for i = 1:n-1
        intervalLength = x(i + 1) - x(i);
```

```

nGQ = 4; % 가우스 적분의 점의 개수
approx_values = GQIntegral1D(f, x(i), x(i+1), nGQ, 10);
loadVector(i) = loadVector(i) + approx_values/2;
loadVector(i + 1) = loadVector(i + 1) + approx_values/2;
end

% local_M 행렬 생성
local_M = [h/3, h/6; h/6, h/3];
% M_matrix
M = sparse(n, n);
for i = 1:n-1
    M(i:i+1, i:i+1) = M(i:i+1, i:i+1) + local_M;
end

x_solution = M \ loadVector;
projected_function = x_solution;
end

```

f =

다음 값을 갖는 function_handle:

```
@(x)sin(pi*x)
```

error rate

0.9426	1.0150	1.0131	1.0076
--------	--------	--------	--------





