Table of Contents

Roshan Jaiswal-Ferri, Alessandro Tedeschi, Stefan Rosu

```
%Section - 02
%Aero 431 HW2: 4/28/25
```

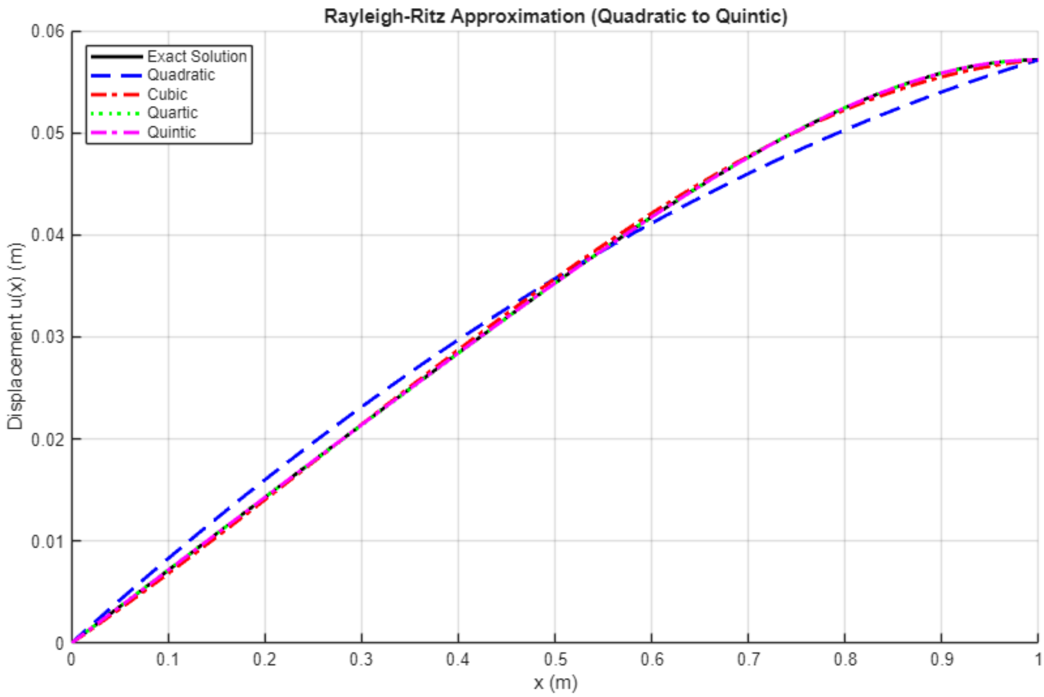
Workspace Prep

```
%warning off
format long       %Allows for more accurate decimals
close all;        %Clears all
clear all;        %Clears Workspace
clc;              %Clears Command
```

Rayleigh-Ritz

```
E = 70e9;      % Pa
A = 5e-6;      % m^2
L = 1;         % m
P = 100e3;     % N/m

syms x real

% Exact solution
u_exact_fn = @(xq) (P*L^2/(4*E*A)) * (xq/L - (1/5)*(xq/L).^5);

degrees = [2, 3, 4, 5];
colors = {'b--', 'r-.', 'g:', 'm-.'};
labels = {'Quadratic', 'Cubic', 'Quartic', 'Quintic'};

x_plot = linspace(0, L, 500);
figure;
hold on;
plot(x_plot, u_exact_fn(x_plot), 'k-', 'LineWidth', 2, 'DisplayName', 'Exact
Solution');

for idx = 1:length(degrees)
    n = degrees(idx);

    a = sym('a', [n,1], 'real');
```

```
    u_trial = 0;
    for i = 1:n
        u_trial = u_trial + a(i)*x^(i);
    end

    u_trial_dx = diff(u_trial, x);

    b = -P*(x/L)^3;
    Energy = (1/2)*E*A*int(u_trial_dx^2, x, 0, L) - int(b*u_trial, x, 0, L);

    eqns = gradient(Energy, a);
    eqns = simplify(eqns);
    [M, f] = equationsToMatrix(eqns, a);

    M = double(M);
    f = double(-f);

    a_sol = M\f;

    u_ritz_expr = subs(u_trial, a, a_sol);
    u_ritz_fn = matlabFunction(u_ritz_expr, 'Vars', x);

    plot(x_plot, u_ritz_fn(x_plot), colors{idx}, 'LineWidth', 2,
'DisplayName', labels{idx});
end

xlabel('x (m)');
ylabel('Displacement u(x) (m)');
title('Rayleigh-Ritz Approximation (Quadratic to Quintic)');
legend('Location', 'northwest');
grid on;
hold off;
```

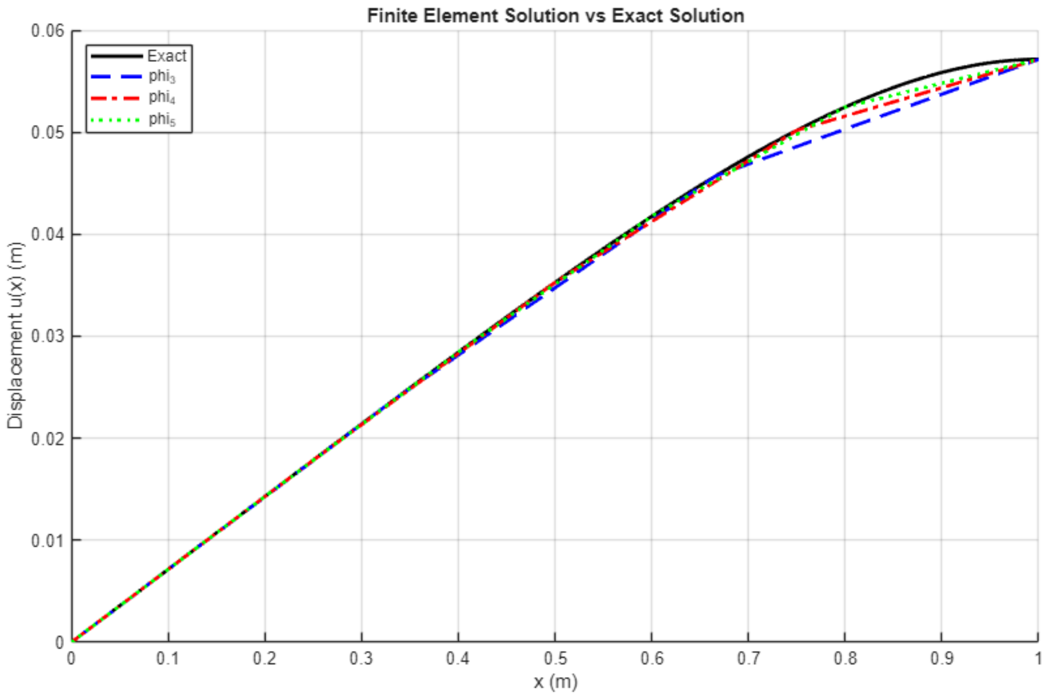


FEM Solution

```
E = 70e9;     % Pa
A = 5e-6;     % m^2
L = 1;        % m
P = 100e3;    % N/m

% Exact solution
u_exact_fn = @(xq) (P*L^2/(4*E*A)) * (xq/L - (1/5)*(xq/L).^5);

n_list = [3, 4, 5];
colors = {'b--', 'r-.', 'g:'};
x_plot = linspace(0, L, 500);

figure;
hold on;
plot(x_plot, u_exact_fn(x_plot), 'k-', 'LineWidth', 2, 'DisplayName',
'Exact');

for idx = 1:length(n_list)
    n_elements = n_list(idx);
    [x_nodes, U] = FEM_solver(n_elements, E, A, L, P);

    % Interpolated FEM solution
    u_fem_fn = @(xq) fem_interpolate(xq, x_nodes, U);

    plot(x_plot, arrayfun(u_fem_fn, x_plot), colors{idx}, 'LineWidth', 2, ...
         'DisplayName', sprintf('phi_%d', n_elements));
```

```
end

xlabel('x (m)');
ylabel('Displacement u(x) (m)');
title('Finite Element Solution vs Exact Solution');
legend('Location', 'northwest');
grid on;
hold off;

function [x_nodes, U_full] = FEM_solver(n_elements, E, A, L, P)
    syms x real

    x_nodes = linspace(0, L, n_elements+1);
    h = x_nodes(2) - x_nodes(1);

    K_full = zeros(n_elements+1);
    F_full = zeros(n_elements+1,1);

    for e = 1:n_elements
        x1 = x_nodes(e);
        x2 = x_nodes(e+1);

        Ke = (E*A/h) * [1 -1; -1 1];

        b = @(xq) P*(xq/L).^3;
        phi1 = @(xq) (x2 - xq)/h;
        phi2 = @(xq) (xq - x1)/h;

        Fe = zeros(2,1);
        Fe(1) = integral(@(xq) b(xq).*phi1(xq), x1, x2);
        Fe(2) = integral(@(xq) b(xq).*phi2(xq), x1, x2);

        idx = [e, e+1];
        K_full(idx, idx) = K_full(idx, idx) + Ke;
        F_full(idx) = F_full(idx) + Fe;
    end

    K_reduced = K_full(2:end, 2:end);
    F_reduced = F_full(2:end);

    U_reduced = K_reduced \ F_reduced;

    U_full = [0; U_reduced];
end

function u_val = fem_interpolate(xq, x_nodes, U)
    n_elements = length(x_nodes) - 1;
    h = x_nodes(2) - x_nodes(1);
    u_val = zeros(size(xq));

    for e = 1:n_elements
        x1 = x_nodes(e);
        x2 = x_nodes(e+1);
        idx = (xq >= x1) & (xq <= x2);
```

```
        if any(idx)
            xi = xq(idx);
            phi1 = (x2 - xi)/h;
            phi2 = (xi - x1)/h;
            u_val(idx) = U(e)*phi1 + U(e+1)*phi2;
        end
    end
end
```



Published with MATLAB® R2024b