Lista 4 Codigos

1.

```
clear all
close all
clc

% Dados
A = 1;
L_0 = 1;
E = 1;
beta_0 = pi / 3;

% Parametros numericos
tolerance = 1e-3;
residual = 10;
max_iterations = 40;

% Carga
q = 2.321;
ncarga = 40;

q_i = linspace(0, q, ncarga + 1);

u = zeros(ncarga, 1);

for n = 1:ncarga

    w = 0;

    u_aux = u(n);


    while abs(residual) > tolerance && w < max_iterations

        L = sqrt(u_aux^2 - 2 * L_0 * sin(beta_0) * u_aux + L_0^2);

        beta = asin((L_0 * sin(beta_0) - u_aux) / L);

        N = (A * E / L_0) * (L_0 - L);

        f = 2 * N * sin(beta);

        k = 2 * N * ((cos(beta)^3) / (L_0 * cos(beta_0))) + 2 * (A * E / L_0) * sin(beta)^2;
```

```
        residual = q_i(n + 1) - f;

        delta_u = (1 / k) * residual;

        u_aux = u_aux + delta_u;

        w++;
    end

    u(n + 1) = u_aux;
    residual = 10;

end

% Solução analitica
q_anal = zeros(1, ncarga + 1);
u_anal = linspace(0, u(end), ncarga + 1);

for i = 1:length(u_anal)

    q_anal(i) = 2 * (A * E / L_0) * (L_0*sin(beta_0) - u_anal(i)) * ((1/sqrt((u_anal(i)/L_0)

end

% Plot the results
figure(1)
plot(u, q_i, 'o-', 'DisplayName', 'Solução Numérica')
hold on
plot(u_anal, q_anal, 'r-', 'DisplayName', 'Solução Analítica')
hold off
xlabel('u (m)')
ylabel('q (N/m)')
title('Carga vs Deslocamento')
legend('show')
grid on

% Save the results in csv format
headers = {'Displacement (m)', 'Load (N/m)'};
fid = fopen('main_1.csv', 'w');
fprintf(fid, '%s,%s\n', headers{1}, headers{2});
fclose(fid);
dlmwrite('main_1.csv', [u, q_i.'], '-append');
% Save the figure
saveas(gcf, 'main_1.png')
```

2.

```
clear all
close all
clc

% Dados
A = 1;
L_0 = 1;
E = 1;
beta_0 = pi / 3;

% Parametros numericos
tolerance = 1e-3;
residual = 10;
max_iterations = 40;

% Carga
q = 2.321;
ncarga = 80;

q_i = linspace(0, q, ncarga + 1);

% Varia o k_s de com incrementos de 0, 2kl no intervalo de 0, 1kl a 3, 0kl

k_l = 2 * (A * E / L_0) * sin(beta_0)^2;

k_s = [0.1:0.2:3.0]*k_l;

u = zeros(ncarga, length(k_s));

for s = 1:length(k_s)

    for n = 1:ncarga

        w = 0;

        u_aux = u(n, s);

        while abs(residual) > tolerance && w < max_iterations

            L = sqrt(u_aux^2 - 2 * L_0 * sin(beta_0) * u_aux + L_0^2);

            beta = asin((L_0 * sin(beta_0) - u_aux) / L);

            N = (A * E / L_0) * (L_0 - L);
```

```
            f = 2 * N * sin(beta);

            k = 2 * N * ((cos(beta)^3) / (L_0 * cos(beta_0))) + 2 * (A * E / L_0) * sin(beta

            f_mola = k_s(s) * u_aux;

            q_aux = q_i(n + 1) - f_mola;

            residual = q_aux - f;

            delta_u = (1 / k) * residual;

            u_aux = u_aux + delta_u;

            w++;
        end

        u(n + 1, s) = u_aux;
        residual = 10;

    end
    display(['k_s = ', num2str(k_s(s)), ' k_l = ', num2str(k_l), ' u = ', num2str(u(n + 1, s

end

% Plota os resultados de q em funcao de u
figure(1)
for s = 1:length(k_s)
    plot(u(:, s), q_i, 'o-', 'DisplayName', ['k_s = ', num2str(k_s(s))])
    hold on
end
xlabel('u')
ylabel('q')
title('Carga em funcao do deslocamento')
legend('show')
grid on


% Save the results in csv format
headers = cell(1, length(k_s) + 1);
headers{1} = 'Load (N/m)';
for s = 1:length(k_s)
    headers{s + 1} = sprintf('Displacement for k_s=%.2f*k_l (m)', k_s(s)/k_l);
end
```

```
% Create CSV file
fid = fopen('main.csv', 'w');

% Write header
header_str = strjoin(headers, ',');
fprintf(fid, '%s\n', header_str);
fclose(fid);

% Prepare data matrix - first column is load, remaining columns are displacements
result_data = zeros(ncarga+1, length(k_s) + 1);
result_data(:, 1) = q_i.';
for s = 1:length(k_s)
    result_data(:, s + 1) = u(:, s);
end

% Append data to CSV
dlmwrite('main.csv', result_data, '-append');
% Save the figure
saveas(gcf, 'main.png')
```