**CODE ANALYSIS**

This MATLAB code is part of a **design and simulation workflow** for a MEMS (Microelectromechanical Systems) actuator or structure composed of **Bezier-shaped beams**. It integrates with **Abaqus**, a finite element simulation tool, to analyze the mechanical response of these structures.

Main code is ‘**sb7pt\_N40\_gap\_between\_upper\_lower.m**’.

Below is a **detailed explanation** of the code, section by section:

## ****Initial Setup and Cleaning****

clear all;

close all;

clc

delete \*.lck

delete \*.dat

* Clears workspace, closes all figures, and deletes Abaqus lock (\*.lck) and data (\*.dat) files to start fresh.

## ****Parameter Initialization****

format long

fitall\_sa = zeros(1000,7);

CHKEXIT = 9999999;

ALSDTOL = 0.05;

STABLIZ = 2e-2;

INCREMEINI = 1e-4;

INCREME = 0.01;

* fitall\_sa: Array to monitor optimization results.
* ALSDTOL: Tolerance for stabilization energy in simulation.
* STABLIZ, INCREMEINI, INCREME: Parameters for running Abaqus nonlinear analysis.

## ****Geometry and Material Constants****

L1 = 35; H1 = 50;

L2 = 70; H2 = 100;

NINNER = 40; NOUTER = 40;

YOUNG = 3.5e+3;

NUXY = 0.38;

DENS = 1.4e-6;

OPDIM = 3;

IPDIM = 0.8;

IPDIMOU = 0.8;

GAP = 5;

* Geometrical and material properties for upper and lower beams.
* Units: [mm], [MPa], [tonne/mm³].

## ****Load Optimized Control Points****

load -ascii iters7pt\_se6.txt

pop = iters7pt\_se6(end,:);

* Loads the **last optimized set of Bezier control points** from a file).

## ****Extract Control Points (Upper Beam)****

B2X = pop(1);

B2Y = pop(2);

…

B5X = pop(7);

B5Y = pop(8);

B6X = pop(7);

B6Y = pop(8);

…

B8Y = pop(18);

L1 = pop(19);

H1 = pop(20);

BxU = [0 B2X B3X B4X B5X B6X L1]';

ByU = [H1 B2Y B3Y B4Y B5Y B6Y H1]';

dummy = [1 1 1 1 1 1 1]';

[xL\_1b yL\_1b temp] = Bezierauto(BxU,ByU,dummy,NINNER);

* Constructs the **upper Bezier beam** shape using 7 control points.
* Bezierauto(...): A custom function to generate Bezier curves.
* The same value of B5X - B6X and B5Y - B6Y could be explained that it’s intentional to controls how the curve transitions toward the endpoint.

## ****Extract Control Points (Lower Beam)****

B5X = pop(end,7);

B5Y = pop(end,8);

B6X = pop(end,5);

B6Y = pop(end,6);

BxL = [0 B12X B11X ... L1]';

ByL = [0 B12Y B11Y ... H1]';

dummy = [1 1 1 1 1 1 1]';

[xL\_1a yL\_1a temp] = Bezierauto(BxL,ByL,dummy,NOUTER);

* Same as above, but for the **lower beam**. Control points differ from the upper beam to give it a different curve.

## ****Plotting Initial Shape****

figure(100);

hnd1 = plot(pointsx,pointsy,'-ob',pointsxOU,pointsyOU,'-or',BxU,ByU,'dm',BxL,ByL,'xk');

set(hnd1,'LineWidth',LW,'MarkerSize',MS);

set(gca)

xlabel('X [mm]');

ylabel('Y [mm]');

axis equal;

grid on;

* Plots both beams:
  + Blue line (-ob) for upper beam
  + Red line (-or) for lower beam
  + Control points as diamonds and crosses

## ****Simulation****

DELTATH = -10; % Vertical deformation

func\_sb\_gap(GAP, NINNER, NOUTER, ..., DELTATH, INCREME, ...)

* Calls a custom function func\_sb\_gap(...) that likely:
  + Generates an Abaqus input file
  + Simulates the structure with a vertical displacement (DELTATH)
  + Accounts for the **gap** between beams
  + Extracts forces or displacements