🛠️ Objective:

The goal of this study is to perform a **linear static analysis** and subsequently conduct a **buckling (stability) analysis** of a curved shell structure using **Code_Aster**. The structure includes **stiffeners** and **ribs**, and the analysis is focused on extracting **critical buckling modes** and understanding structural behavior under a specified moment load.

---

1. Initialization

DEBUT()

We begin the script using `DEBUT()`, which is required in every Code_Aster command file to initialize the session.

---

2. Mesh Import

mesh = LIRE_MAILLAGE(FORMAT='MED', UNITE=20)

- **Why:** We import a `.med` mesh file generated earlier (e.g., from Gmsh).
- **Choice of format:** `'MED'` is the standard mesh format used in Code_Aster.
- **Unit 20:** Matches the unit used in the `.export` file.

---

3. Model Definition

model = AFFE_MODELE(...)

- **MODELISATION = 'DKT'**: We chose `'DKT'` (Discrete Kirchhoff Triangle), a shell element suitable for thin-walled structures and plate bending, especially where curvature and rotational DOFs are important.

- **PHENOMENE = 'MECANIQUE'**: We're dealing with mechanical deformations.
- **TOUT = 'OUI'**: Applies the model to all mesh elements.

---

4. Elemental Properties

elemprop = AFFE_CARA_ELEM(...)

- We defined varying **thicknesses (`EPAIS`)** for different parts:
  - **Curved surface**: 5 mm (very thin shell).
  - **Stiffeners**: 25 mm (adds longitudinal stiffness).
  - **Ribs**: 50 mm (transverse stiffening).
- **VECTEUR** specifies the **local orientation** of the shell elements for correctly applying thickness and stiffness directions.

Why multiple thicknesses?
To simulate realistic structural behavior. Curved shells alone are weak in buckling; stiffeners and ribs provide directional rigidity.

---

5. Material Properties

mater = DEFI_MATERIAU(...)

- We defined **isotropic elasticity** with:
  - **E = 70 GPa**: Common for Aluminum alloys.
  - **NU = 0.3**: Poisson's ratio, standard value for metals.

fieldmat = AFFE_MATERIAU(...)

- Applies the material to the entire model.

---

6. Boundary Conditions

bc = AFFE_CHAR_MECA(DDL_IMPO=...)

- All degrees of freedom (`DX`, `DY`, `DZ`, `DRX`, `DRY`, `DRZ`) are constrained for the `fixed` group.
- This creates a **fully clamped edge** or zone, essential to simulate realistic fixed support in buckling problems.

---

7. Loading

load = AFFE_CHAR_MECA(FORCE_NODALE=...)

- A **moment (`MX = 1.0`)** is applied to the group `Node`.
- This replicates a **unit moment** applied to trigger buckling.
- The direction (MX) aligns with axis **X**, making it easy to interpret buckling about the X-axis.

---

8. Static Analysis

reslin = MECA_STATIQUE(...)

- Solves the **linear static problem** at time `INST=1.0`.
- **OPTION = 'SIEF_ELGA'**: Stores **stress resultants (forces and moments)** at Gauss points, which are later used to compute geometric stiffness (KG matrix).

---

9. Post-Processing: Result Export

IMPR_RESU(FORMAT='MED', RESU=...)

- Exports linear analysis results for visualization in tools like Salome.
- **UNITE=80** is a standard choice for exporting results.

---

10. Extract Stress Field

field = CREA_CHAMP(...)

- **Purpose**: Extract the stress field from the static result (`SIEF_ELGA`) to compute the **geometric stiffness matrix** used in buckling analysis.
- **Why:** Euler buckling requires initial stress fields to assess how close the structure is to instability.

---

11. Stiffness Matrix Assembly

ASSEMBLAGE(...)

- Generates two matrices:
  - **KM**: Linear mechanical stiffness (`RIGI_MECA`).
  - **KG**: Geometric stiffness (`RIGI_GEOM`), derived from pre-stress.
- **Why needed:** Buckling is an eigenvalue problem that solves:
  $(KM - \lambda KG)\phi = 0$(KM - \lambda KG) \phi = 0

---

12. Buckling Mode Calculation

modes = CALC_MODES(...)

- **TYPE_RESU = 'MODE_FLAMB'**: Requests **buckling modes** (not vibration).
- **OPTION = 'PLUS_PETITE'**: Finds **lowest (most critical) eigenvalues** (first to buckle).
- **NMAX_CHAR_CRIT = 12**: Calculates first 12 buckling modes.
- **SOLVEUR_MODAL = 'SORENSEN'**: A robust eigenvalue solver suitable for large problems.
- **PREC_SHIFT, SEUIL_CHAR_CRIT**: Precision settings for numerical stability.

---

13. Export Buckling Modes

IMPR_RESU(FORMAT='MED', ...)

- Saves **buckling mode shapes** for viewing in Salome or ParaVis.
- You can visually inspect **how the structure deforms when it buckles**.

---

14. End of Analysis

FIN()

Marks the end of the script.

✅ Summary of Choices:

Feature	Why Chosen
`MODELISATION='DKT'`	Shell formulation suited for curved thin structures
Variable `EPAIS`	Realistic modeling of stiffeners and ribs
Unit moment `MX`	For normalized buckling analysis
`SIEF_ELGA` extraction	Required for calculating KG
`RIGI_MECA`, `RIGI_GEOM`	Fundamental matrices for buckling problem
`SORENSEN` solver	Efficient and accurate for modal/buckling problems
Export to `.med`	Easy integration with Salome for visualization
