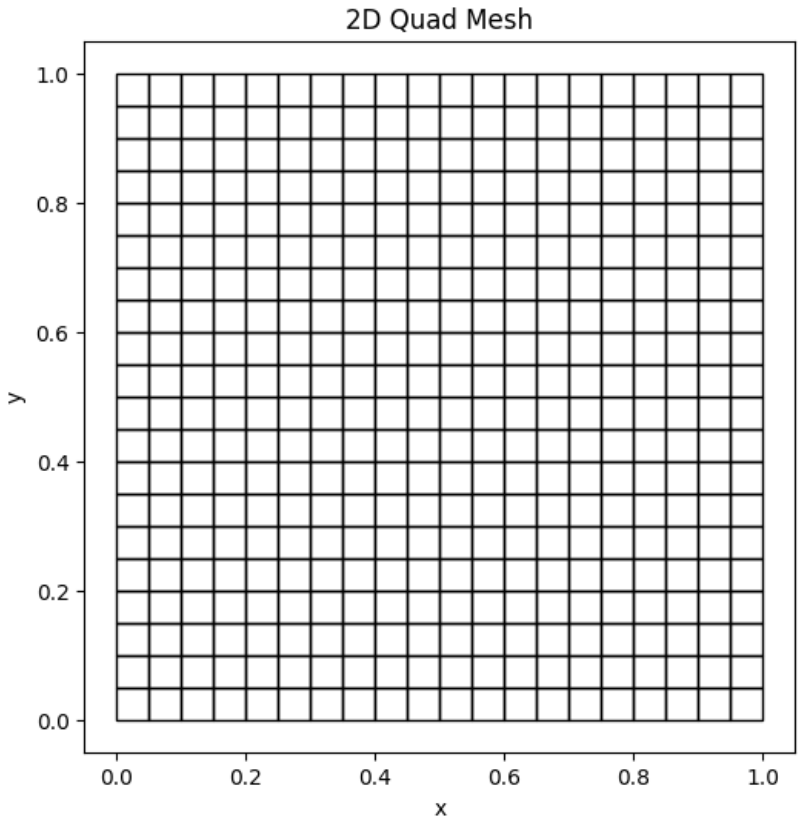
This project implements a complete 2D Finite Element Method (FEM) solver from the ground up to simulate incompressible fluid flow in a **lid-driven cavity** at a Reynolds number (Re) of 1000.

The project is broken into three main stages:

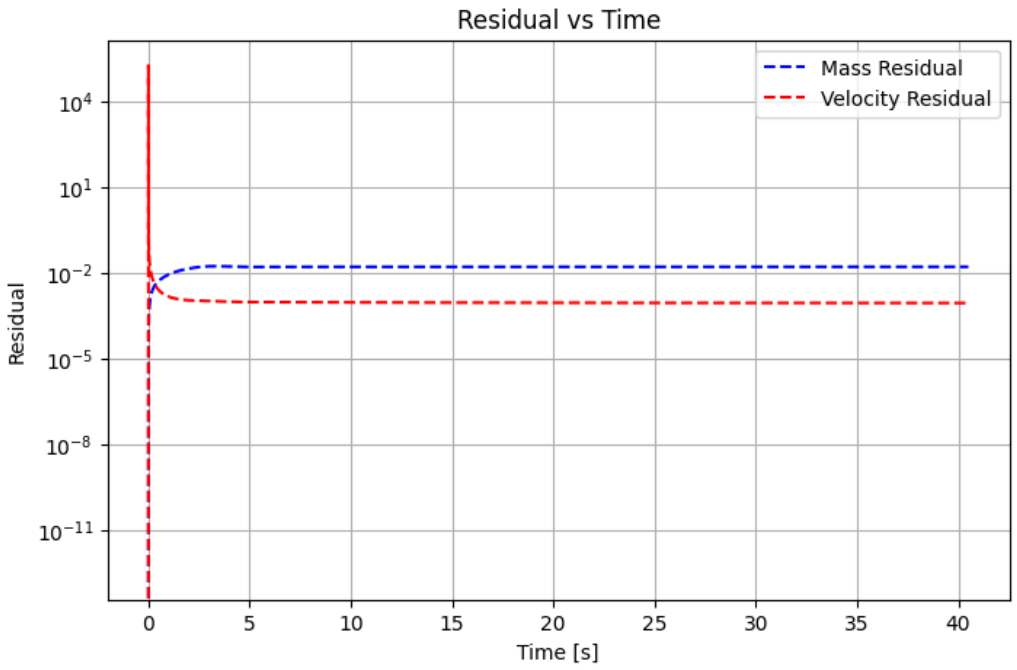
1. Mesh Generation and Validation

- **Geometry Definition:** The script begins by programmatically writing a .geo file. This file defines a 1x1 square domain, which is the standard geometry for the lid-driven cavity problem.
- **Structured Meshing (Gmsh):** It uses Gmsh's Transfinite and Recombine commands to generate a **structured mesh** composed entirely of quadrilateral (quad) elements.
- **External Call:** The Python subprocess module is used to call the Gmsh executable externally, processing the .geo file and outputting a .msh file.
- **Mesh Import (meshio):** The meshio library reads the generated mesh data (node coordinates and element connectivity) into the Python environment.
- **Mesh Quality Checks:** Before starting the simulation, the script performs critical validation on the mesh:
    - **Node Reordering:** A function (reorder_quad_nodes) ensures that the nodes for each element are sorted in a consistent counter-clockwise order. This is essential for correct Jacobian calculation.
    - **Jacobian Check:** It iterates through all elements and calculates the determinant of the Jacobian matrix to ensure it is positive (detJ > 0), verifying that no elements are inverted or malformed.
    - **Aspect Ratio Check:** The aspect ratio of each element is calculated to assess the overall quality and uniformity of the mesh.



2. 2D Navier-Stokes FEM Solver

- **Core Method:** The project implements a transient (time-dependent) solver for the incompressible Navier-Stokes equations using bilinear quadrilateral (Q1) elements for both velocity and pressure fields (a Q1/Q1 formulation).
- **Stabilization (GLS):** Because equal-order elements (Q1/Q1) do not natively satisfy the LBB (inf-sup) condition and to handle the convection-dominated flow at Re=1000, the solver employs a **Galerkin/Least-Squares (GLS) stabilization** technique. This is visible in the solve_momentum function, where the standard test function (N) is modified by a stabilization term (tau) to create the GLS weighting function (w_u, w_v).



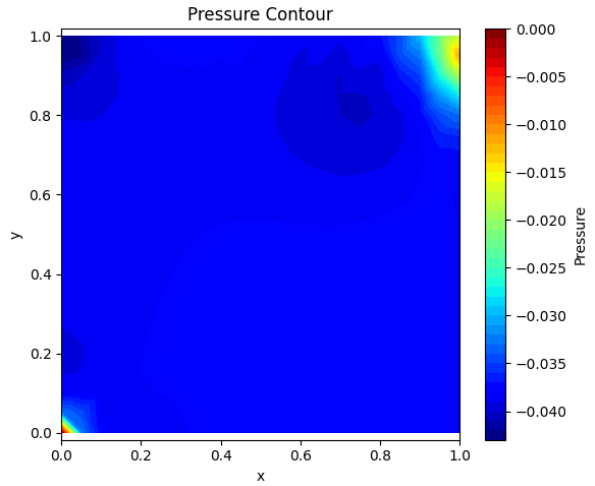
- **Solution Algorithm:** A fractional-step (or projection-like) method is used:
    1. **Pressure Solve:** At the beginning of each time step, a pressure Poisson-like equation is solved (solve_pressure) to get the pressure field p that enforces mass conservation.
    2. **Momentum Solve:** The new pressure p is used in the solve_momentum function to explicitly update the velocity fields (u, v) for the next time step, incorporating the GLS stabilization.
    3. **Boundary Conditions:** The apply_velocity_bc function enforces no-slip conditions on the bottom and side walls (u=0, v=0) and the "lid" condition on the top wall (u=1, v=0).
- **Implementation:** The code uses scipy.sparse for efficient assembly of global matrices (e.g., pressure stiffness K_p, gradient operators G_x, G_y) and numba's @jit decorator to accelerate computationally intensive functions like solve_momentum and shape_functions.

3. Post-Processing and Visualization

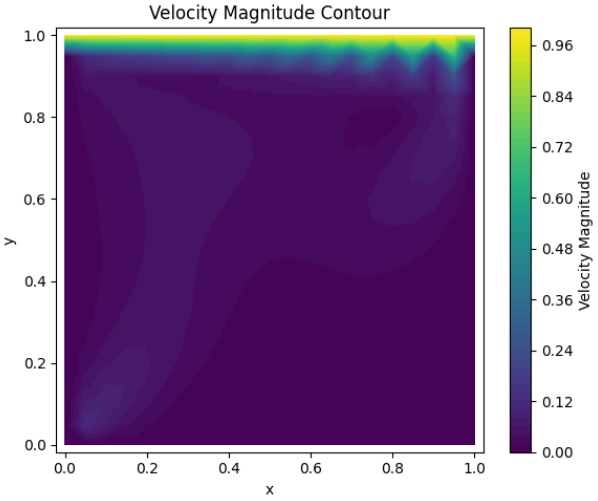
After the time-stepping simulation completes, the script analyzes and visualizes the results:

- **Residual Plotting:** A plot of the mass and velocity residuals over time is generated to monitor the solver's convergence to a steady-state solution.

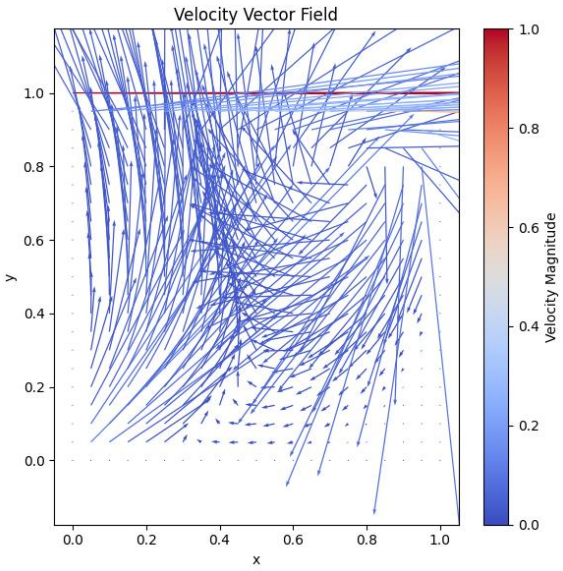
- **Field Visualization:** matplotlib is used to create several plots of the final flow field:
    - **Pressure Contour Plot**



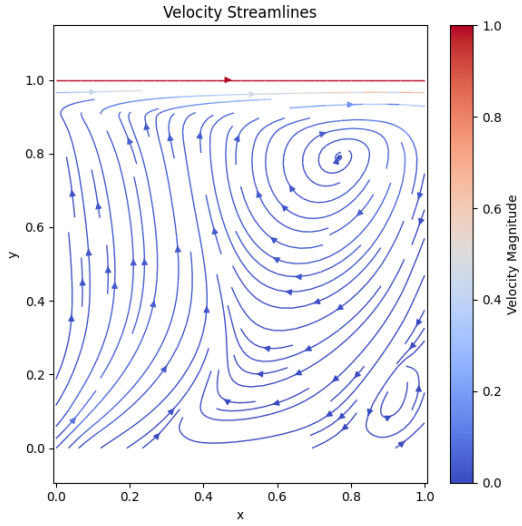
    - **Velocity Magnitude Contour Plot**



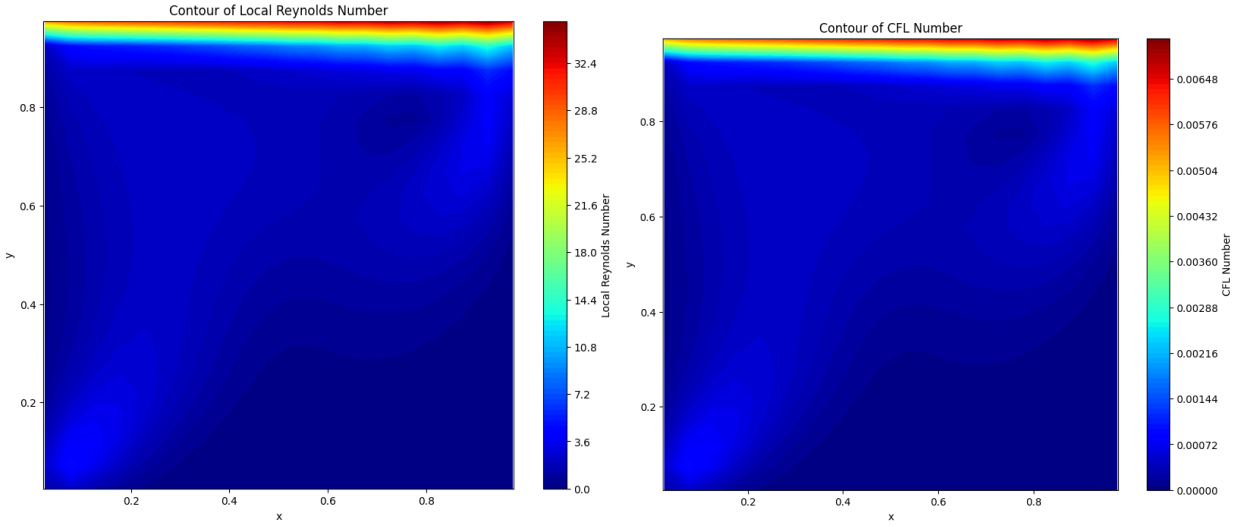
    - **Velocity Vector (Quiver) Plot**



- **Velocity Streamlines**



- **Quantitative Analysis:** The script calculates and plots contours of the **local Reynolds number** and **CFL number** for each element, providing insight into the simulation's numerical stability and properties across the domain. It also includes a function to compute the **primary vortex length**, a standard metric for validating lid-driven cavity results.


Contour of Local Reynolds Number
y
x
0.2
0.4
0.6
0.8
Local Reynolds Number
0.0
3.6
7.2
10.8
14.4
18.0
21.6
25.2
28.8
32.4
Contour of CFL Number
CFL Number
0.00000
0.00072
0.00144
0.00216
0.00288
0.00360
0.00432
0.00504
0.00576
0.00648