Detailed Documentation of the Solver Methodology

1. Governing Equations

This solver addresses the **transient, incompressible 2D Navier–Stokes equations**:

Momentum Equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}$$

Continuity Equation (Incompressibility)

$$\nabla \cdot \mathbf{u} = 0$$

Where:

- $\mathbf{u} = (u, v)$is the velocity field
- $p$is pressure (density assumed $\rho = 1$)
- $t$is time
- $\nu = 1/Re$is the kinematic viscosity
- $(\mathbf{u} \cdot \nabla)\mathbf{u}$represents convection
- $\nu \nabla^2 \mathbf{u}$represents diffusion

---

2. Numerical Method Summary

The solver uses the **Finite Element Method (FEM)** with a **fractional-step-like approach**.

Spatial Discretization

- Quadrilateral **Q1/Q1 elements**
- Bilinear shape functions for both velocity and pressure

Time Integration

- **Explicit Forward Euler**

Stabilization Techniques

1. **GLS (Galerkin/Least-Squares)** to stabilize equal-order elements and convection at $Re = 1000$
2. **Pressure Penalty Approach**
    - Continuity equation is enforced via a penalty-like pressure equation instead of a Poisson solve

---

3. Solution Algorithm

The main routine runs time steps $n = 0,1, \dots, N_t$, and each step consists of:

Step 1 — Pressure Calculation

(Function: *solve_pressure*)

Purpose: Obtain pressure field p^nthat penalizes the divergence of the current velocity field.

We solve the linear system:

$$K_p \mathbf{p}^n = \mathbf{R}_p$$

Pressure Stiffness Matrix

$$[K_p]_{ij} = \int_\Omega (\nabla N_i \cdot \nabla N_j)\, d\Omega$$

Right-Hand Side

$$[R_p]_i = -\frac{1}{\epsilon}[M_L]_{ii}(\nabla \cdot \mathbf{u}^n)_i$$

Where:

- $M_L$is the lumped mass matrix
- $\epsilon = 0.001$is a stabilization parameter

Constraint

- One node is fixed $p_0 = 0$to remove singularity

Step 2 — Momentum Update

(Function: *solve_momentum*)

Velocity is updated explicitly:

$$[M_L]_{ii} u_i^{n+1} = [M_L]_{ii} u_i^n + [R_u]_i$$

Residual Impulse

$$[R_u]_i = -\Delta t \sum_e \int_{\Omega_e} (\mathbf{R}_{\text{conv}} + \mathbf{R}_{\text{press}} + \mathbf{R}_{\text{diff}})\, d\Omega$$

With contributions:

- Convection: $\mathbf{R}_{\text{conv}} = w_{gls}(\mathbf{u} \cdot \nabla)u$
- Pressure: $\mathbf{R}_{\text{press}} = w_{gls} \frac{\partial p}{\partial x}$
- Diffusion: $\mathbf{R}_{\text{diff}} = \nu(\nabla N_i \cdot \nabla u)$

GLS applies to convection and pressure; diffusion uses standard Galerkin.

---

4. GLS Stabilization Formulation

Modified Test Function

$$w_{gls} = N_i + \tau(\mathbf{u} \cdot \nabla N_i)$$

This introduces streamline-aligned artificial diffusion to stabilize convection.

Stabilization Parameter τ

$$\tau = [(\frac{2}{\Delta t})^2 + (\frac{2Pe}{h})^2 + (\frac{4\nu}{h^2})^2]^{-1/2}$$

Where:

- $h = \sqrt{\det J}$ is element characteristic length
- Pe is Peclet number

$$Pe = \frac{|\mathbf{u}|\, h}{2\nu}$$

$|\mathbf{u}|$is the local velocity magnitude.

Final Notes

- Equal-order $Q1/Q1$elements require stabilization — handled by GLS
- Pressure resolved via penalty formulation
- Explicit time stepping simplifies matrix solves at cost of stability requirements