

MECH 570C Code Project 1

January 28, 2024

1 N-S

Function Definition

$\rho u_t + \rho u \cdot \text{div} u - \text{div}(\mu \nabla u) + \nabla p = f$ in Ω , $\text{div} u = 0$ in Ω , Dirichlet boundary condition $u = g_D$ on Γ_D , Neumann boundary condition $\frac{du}{dn} - np = g_N$ on Γ_N .

- **Function Name:** `navierStokes`
- **Inputs:**
 - `solver`, `fluid`, `pmc`: Structures containing various solver settings, fluid properties, and numerical method parameters.
 - `Sol`: Structure holding the solution vectors.
 - `cnn`, `crd`: Connectivity and coordinate matrices for the mesh.
 - `elemType`, `ndof`, `nen`, `nElem`, `BCCyl`: Parameters defining the type of elements used, degrees of freedom, number of nodes per element, total number of elements, and boundary condition information.

Quadrature Rules

- Sets up quadrature points (`gP`) and weights (`gW`) based on the element type (triangular or quadrilateral). These are used for numerical integration.
- Defines shape functions (`N`) and their derivatives (`Nx`, `Ny`) for the finite elements.

Boundary Conditions

- Applies Dirichlet and Neumann boundary conditions to the solution vector `Sol.u`.

Interpolation for Alpha Values

- Interpolates values for the generalized-alpha method, a numerical technique for time integration in transient problems.

Navier-Stokes Equations

- Prepares variables (`xxf`, `yyf`, `ux`, `uy`, etc.) for assembling the finite element matrices. These include coordinates, velocities, pressure, and additional variables.

Assembly of Galerkin and Petrov-Galerkin Terms

- Calls functions to form the left-hand side (LHS) and right-hand side (RHS) of the Navier-Stokes equations. This involves complex operations based on the finite element method.
- The Galerkin method is used for discretizing the problem, while Petrov-Galerkin is an enhanced approach for stability and accuracy.

Solving the Linear System

- Determines the free nodes not constrained by boundary conditions.
- Solves the linear system for the unknowns (`Increment`) using the assembled LHS and RHS.

Update and Output

- Updates the solution vectors (`Sol.u`, `Sol.uDot`, `Sol.p`) with the new increments.
- Calculates a norm (`NSnormIndicator`) to indicate the convergence or error of the current iteration.
- Outputs the updated solution structure `Sol` and the convergence/error indicator.

Summary of Data Flow

1. **Input Processing:** Takes in initial conditions, mesh information, and solver parameters.
2. **Setup:** Establishes quadrature rules and boundary conditions.
3. **Equation Assembly:** Forms the Navier-Stokes equations using finite element discretization.
4. **Solution Update:** Solves for increments and updates the solution.
5. **Output:** Returns the updated solution and a convergence/error metric.

2 Integrated Output

Function Definition

- **Function Name:** IntegratedOutput
- **Inputs:**
 - **Sol:** A structure containing the solution vectors (velocity, pressure, etc.).
 - **crd:** The coordinates of the mesh nodes.
 - **BCCyl:** Boundary condition data.
 - **fluid:** A structure containing fluid properties.
 - **cnn:** Connectivity matrix for the mesh elements.

Initialization

- Sets the number of element nodes (**n_{en}**) for a 2D element (4 for a quadrilateral element).
- Swaps the columns of **BCCyl** for further processing.
- Determines the number of elements (**nElem**) and degrees of freedom (**ndof**) involved in the boundary condition.

Boundary Layer Elements

- Identifies elements corresponding to the first layer of the boundary using `cnn` and `BCCyl`.
- Reorders the element points for reduced integration.

Quadrature Integration Setup

- Defines Gauss points (`gP`) and weights (`gW`) for numerical integration.
- Sets up shape functions (`N`) and their derivatives (`Nx`, `Ny`).

Localizing Data

- Extracts local coordinates (`xxf`, `yyf`), velocities (`ux`, `uy`), and pressure (`pres`) for each element.

Integration Process

- Loops over quadrature points to:
 - Calculate the Jacobian matrix `J` for coordinate transformation.
 - Compute the volume and normal vectors for each element.
 - Evaluate pressure and velocity gradients (`locgradUx`, `locgradUy`, `locgradVx`, `locgradVy`).
 - Compute the length/area (`A0`) of the line/surface integral.

Calculation of Forces

- The loop sums up the contributions from all quadrature points to calculate the integrated force over the element surface.
- The force components (X and Y directions) are to be computed within the loop (currently not implemented in the provided code).

Output

- Returns the total length of the integrated area (`Length`) and the force components (`Force`).