Thermoflow2D Solver Pro

Manual for 2D Incompressible Flow with Heat Transfer Coupling

CFD Solutions Inc.

June 8, 2025

Revolutionary CFD Simulation for the Future

Discover the power of advanced 2D incompressible flow and heat transfer coupling.

School:	China-UK Low Carbon College
Class ID:	24M002
Student ID:	124280910038
Name:	Jiaxin Wang

https://github.com/wjx0209/Thermoflow2D-Solver-Pro.git

Empowering Innovation in Fluid Dynamics

Contents

1 Introducing Thermoflow2D Solver Pro

Welcome to **Thermoflow2D Solver Pro**, the cutting-edge solution for simulating 2D incompressible flow with heat transfer coupling. Designed for engineers and researchers, our solver combines state-of-the-art numerical methods with user-friendly interfaces to deliver unparalleled performance and accuracy.

1.1 Why Choose Thermoflow2D Solver Pro?

- **Powerful**: Solves complex Navier-Stokes equations with coupled heat transfer.
- **Flexible**: Supports laminar/turbulent flows and natural/forced convection.
- **Efficient**: Leverages parallel computing (OpenMP/MPI) for fast simulations.
- **Accessible**: Intuitive setup and visualization tools for all expertise levels.

1.2 Applications

- Aerospace: Optimize thermal management in aircraft components.
- Automotive: Enhance heat exchanger designs.
- Energy: Simulate convection in renewable energy systems.
- Research: Explore fluid dynamics and heat transfer phenomena.

2 Governing Equations

The Thermoflow2D Solver Pro is based on the vorticity-stream function formulation of the incompressible Navier-Stokes equations coupled with the energy equation. The system of partial differential equations consists of:

- The continuity equation, which ensures mass conservation;
- The vorticity-stream function formulation of the momentum equations;
- The energy conservation equation with convection and diffusion.

In their dimensional scalar form, these equations are:

2.1 Continuity Equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

2.2 Vorticity Definition

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{2}$$

2.3 Stream Function Equation (Poisson Form)

$$\nabla^2 \psi = -\omega \tag{3}$$

2.4 Vorticity Transport Equation

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \tag{4}$$

2.5 Energy Equation

$$\rho c_p \left(\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} \right) = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \tag{5}$$

Here, u and v are the velocity components in x and y directions, ω is the vorticity, ψ is the stream function, T is temperature, ρ is density, μ is dynamic viscosity, $\nu = \mu/\rho$ is the kinematic viscosity, k is thermal conductivity, and c_p is the specific heat capacity at constant pressure.

3 Advanced Numerical Methods

Thermoflow2D Solver Pro employs a classic yet effective combination of finite difference schemes and iterative solvers to resolve 2D incompressible fluid and thermal convection problems based on the streamfunction-vorticity formulation.

3.1 Spatial Discretization

- **Central Difference Schemes** are employed for both convective and diffusive terms, achieving second-order spatial accuracy on a structured grid:

  $$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} \approx \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x},$$
  $$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i,j} \approx \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2}.$$

  These are used in solving vorticity transport and temperature transport equations.
- **Uniform Cartesian Grid:** The domain is discretized using evenly spaced nodes in both x and y directions, allowing easy implementation of finite difference operators.
- **Streamfunction-Vorticity Formulation:** The Navier-Stokes equations are reformulated to avoid pressure calculation by introducing streamfunction ψ and vorticity ω:

  $$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x},$$
  $$\nabla^2 \psi = -\omega.$$

3.2 Temporal Discretization

- **Explicit Euler Method** is used for time advancement of vorticity and temperature fields:

  $$\phi^{n+1} = \phi^n + \Delta t \cdot RHS^n,$$

  where RHS^n contains convection and diffusion terms at time step n.
- **CFL-Constrained Time Step:** The time step Δt is carefully chosen to satisfy both convective and diffusive stability limits:

  $$\Delta t_{\text{adv}} \leq \frac{0.5 \min(\Delta x, \Delta y)}{U_{\text{max}}},$$
  $$\Delta t_{\text{diff}} \leq \frac{0.25 \min(\Delta x^2, \Delta y^2)}{\max(\nu, \alpha)}.$$

  The final time step used is the minimum of the two.
- **Fixed Time Stepping:** All simulations run with constant Δt to ensure simplicity and consistent error control.

3.3 Solution Algorithm

- **Successive Over-Relaxation (SOR)** is used to solve the Poisson equation for the streamfunction:

$$\psi_{i,j}^{(k+1)} = \frac{1}{2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)} \left[\frac{\psi_{i+1,j}^{(k)} + \psi_{i-1,j}^{(k)}}{\Delta x^2} + \frac{\psi_{i,j+1}^{(k)} + \psi_{i,j-1}^{(k)}}{\Delta y^2} + \omega_{i,j} \right].$$

  Iteration continues until the residual falls below a threshold (e.g., 10^{-5}).

- **Vorticity Transport Equation** is solved explicitly to update ω:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu \nabla^2 \omega.$$

- **Temperature Transport Equation** (convection-diffusion) is similarly solved:

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha \nabla^2 T.$$

- **Velocity Reconstruction:** At each time step, u and v are recalculated from ψ via finite differences to ensure divergence-free flow.
- **Boundary Conditions:** Both Dirichlet and Neumann conditions are applied explicitly based on physical boundary settings (e.g., no-slip walls, adiabatic or isothermal surfaces).

4 Solver Structure

Thermoflow2D Solver Pro is designed with an interactive GUI-driven architecture that balances flexibility, modularity, and real-time visualization for CFD-heat transfer simulations.

4.1 Modular Design

Module	Description
GUI Interface	Built with *Tkinter* and *ttkbootstrap*, it allows users to input domain parameters, set boundary conditions, and control simulation flow.
Configuration Manager	Provides JSON-based file I/O for saving and loading user-defined simulation parameters.
Solver Engine	Encapsulates explicit finite difference algorithms, vorticity-streamfunction formulation, SOR iteration, and time-stepping schemes.
Boundary Handler	Applies Dirichlet or Neumann conditions for both velocity and temperature fields based on user-defined toggles.
Animation Renderer	Uses *matplotlib.animation* and *FigureCanvasTkAgg* to dynamically plot flow fields and scalar quantities.

4.2 Core Functional Components

- **Class:** *CFDSimulationGUI*
  Central class controlling the user interface, simulation logic, and plot rendering.
- **Flow and Thermal Input Modules**
  Parameters like domain size, Reynolds/Prandtl numbers, boundary conditions, and grid resolution are bound to *tk.DoubleVar* and handled by flow/thermal widget panels.
- **Numerical Solver Core**
  Implements central-difference spatial discretization, explicit Euler time stepping, and SOR-based Poisson solver for the streamfunction.

- **Field Memory Structures**
  Arrays such as *u_full, v_full, omega, psi, T* are stored over time via *histories* for animation and 1D slicing visualization.
- **Error and Stability Control**
  Computes stable Δt using CFL and Fourier criteria; includes residual-based SOR loop with convergence threshold.
- **Real-Time Visualization**
  Multi-panel *matplotlib* figures display velocity vectors, streamlines, temperature contours, and centerline plots.

4.3 User Interaction Workflow

- User enters simulation parameters via GUI.
- Pressing *Run Simulation* triggers the solver, applies boundary conditions, and begins iteration.
- Fields are updated per timestep; intermediate results stored.
- After simulation, animated plots are rendered in the GUI.
- Users can export/load configurations as JSON.

5 Instructions

5.1 Installation

To install *Thermoflow2D Solver Pro*, follow these steps:

- Clone the repository:
  ```
  $ git clone https://github.com/wjx0209/Thermoflow2D-Solver-Pro.git
  ```
- Navigate to the project directory:
  ```
  $ cd Thermoflow2D-Solver-Pro
  ```
- Run the installer script with optional prefix:
  ```
  $ ./install.sh -prefix=/opt/thermoflow2d
  ```
- Add the binary path to your environment variables (e.g., in ~̃.bashrc or ~̃.zshrc):
  ```
  export PATH=/opt/thermoflow2d/bin:$PATH
  ```
- Verify installation:
  ```
  $ thermoflow2d -version
  ```

5.2 Running Your First Simulation

Follow the steps below to set up and launch the simulation GUI from the terminal:

```
# Step 1: Save the GUI Python script
$ nano Thermoflow2D_GUI.py
# (Paste the full code, then Ctrl+O, Enter, Ctrl+X to save and exit)

# Step 2: (Optional but recommended) Create and activate a virtual environment
$ python3 -m venv venv
$ source venv/bin/activate

# Step 3: Install required dependencies
$ pip install numpy matplotlib ttkbootstrap

# Step 4: Run the GUI application
$ python3 Thermoflow2D_GUI.py
```

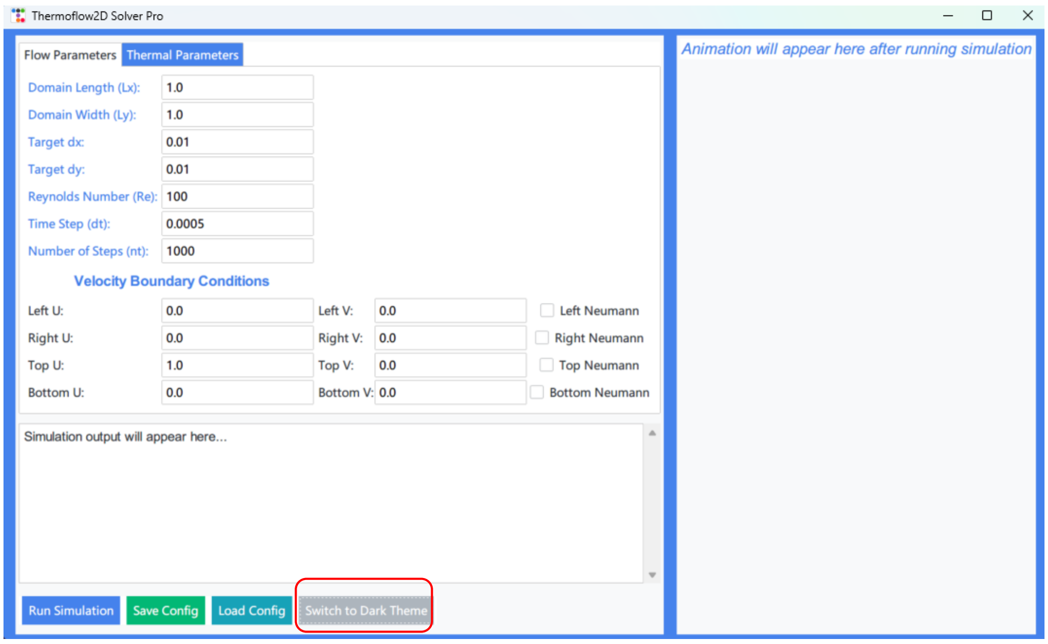
Note: If you are using a remote server, ensure that X11 forwarding is enabled (e.g., using `ssh -X`) to display the GUI properly.

Once the application starts, input the simulation parameters via the graphical interface, then click `Run Simulation` to begin. Results will be visualized in animated plots within the same window.

5.3 Interface Description

Figure 1 shows the main interface of **Thermoflow2D Solver Pro**, which is divided into three key areas:

- **Input Panel (Left)**: Users can set flow parameters such as domain size, mesh spacing, Reynolds number, and time-stepping configurations. It also includes boundary condition settings for velocity and temperature.
- **Console Output (Bottom Left)**: Displays simulation progress, warnings, and messages for user feedback.
- **Visualization Panel (Right)**: The simulation results, including velocity field, temperature contours, and streamlines, are dynamically rendered in this section.



Domain Length (Lx):	1.0
Domain Width (Ly):	1.0
Target dx:	0.01
Target dy:	0.01
Reynolds Number (Re):	100
Time Step (dt):	0.0005
Number of Steps (nt):	1000

Velocity Boundary Conditions

Left U:	0.0	Left V:	0.0	☐ Left Neumann
Right U:	0.0	Right V:	0.0	☐ Right Neumann
Top U:	1.0	Top V:	0.0	☐ Top Neumann
Bottom U:	0.0	Bottom V:	0.0	☐ Bottom Neumann

Simulation output will appear here...

Animation will appear here after running simulation

Figure 1: Thermoflow2D Solver Pro Interface

Theme Toggle: The interface supports both **light** and **dark** themes for visual comfort in different working conditions. Click the `Switch to Dark Theme` button (circled in red) to instantly adapt the color scheme.

5.4 Input Configuration

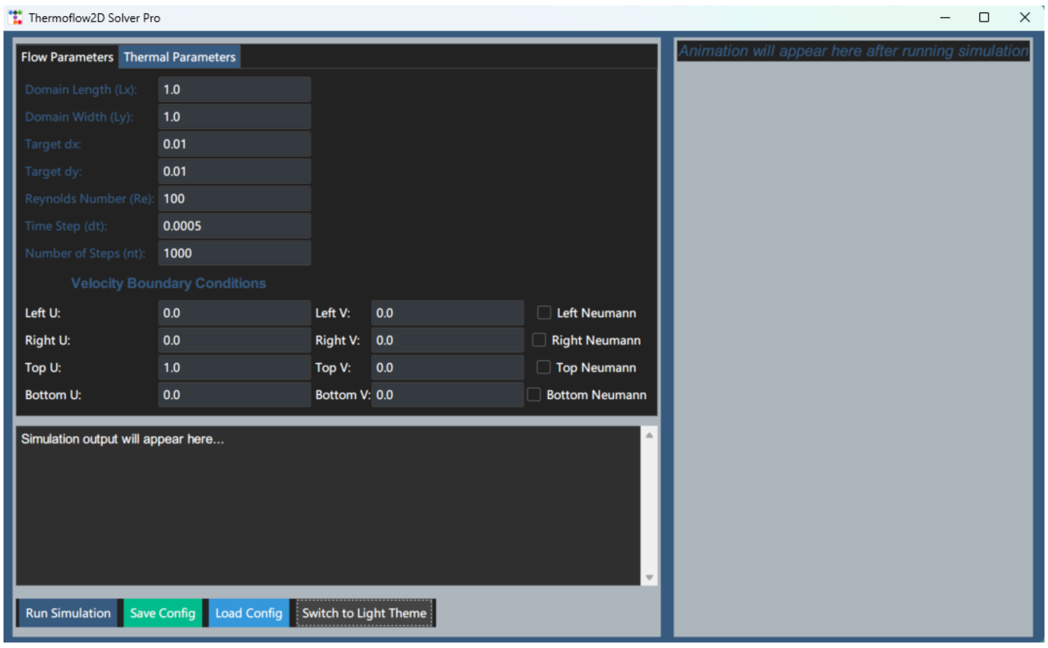
▷ JSON-based for easy setup.

▷ Example configuration available in the user manual.

▷ The input parameters can be changed at will, or they can be saved in advance and loaded and then manually modified in the GUI interface.

6 Results and Insights

Cases with different boundary conditions are calculated here to demonstrate the solution results of this solver. Includes 2D field plots (such as temperature field, velocity field) and 1D line plots (variable distribution along the centerline or user-specified section).



Domain Length (Lx): 1.0
Domain Width (Ly): 1.0
Target dx: 0.01
Target dy: 0.01
Reynolds Number (Re): 100
Time Step (dt): 0.0005
Number of Steps (nt): 1000

Velocity Boundary Conditions

Left U:	0.0	Left V:	0.0	Left Neumann
Right U:	0.0	Right V:	0.0	Right Neumann
Top U:	1.0	Top V:	0.0	Top Neumann
Bottom U:	0.0	Bottom V:	0.0	Bottom Neumann

```
Simulation output will appear here...
```

Figure 2: Dark Theme Thermoflow2D Solver Pro

6.1 Validation Cases

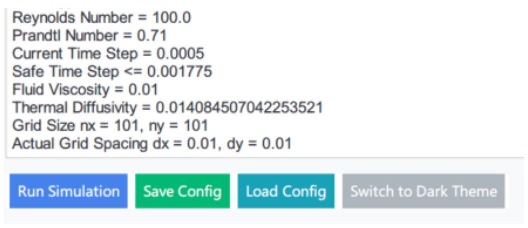
6.1.1 2D Lid-Driven

This is a square cavity flow verification case. The parameter settings and boundary condition settings are shown in the following table.

Table 1: Simulation conditions and boundary settings

Parameter	Setting	Description
Domain Size	$L = 1.0$	Square domain 1.0×1.0
Grid Resolution	$nx = ny = 101$	Grid size 101×101
Lid Velocity	$U = 1.0$	Uniform velocity on top lid
Reynolds Number	$Re = 100$	Moderate Reynolds number
Prandtl Number	$Pr = 0.71$	Typical for air
Top Wall	$T = 300\,\mathrm{K}$	Hot wall (Dirichlet)
Bottom Wall	$T = 30\,\mathrm{K}$	Cold wall (Dirichlet)
Left/Right Wall	$\partial T/\partial x = 0$	Adiabatic (Neumann)
Time Step	$dt = 0.0005$	Satisfies explicit scheme stability
Total Steps	$nt = 1000$	Total simulation steps

Because this solver uses the displayed finite difference method and SOR method, numerical instability may occur. Therefore, dt should not be too large. Therefore, simulation information will be displayed in the middle of the GUI interface to prompt you whether the currently selected dt meets the stability conditions, as shown in the figure below.



```
Reynolds Number = 100.0
Prandtl Number = 0.71
Current Time Step = 0.0005
Safe Time Step <= 0.001775
Fluid Viscosity = 0.01
Thermal Diffusivity = 0.014084507042253521
Grid Size nx = 101, ny = 101
Actual Grid Spacing dx = 0.01, dy = 0.01
```

Figure 3: GUI interface simulation process box information

The results of the two-dimensional temperature field and velocity field and One-dimensional line plot

Table 2: Simulation setup and boundary condition summary

Parameter	Value	Description
Domain Size	$L_x = 5.0,\ L_y = 1.0$	Channel flow dimensions
Grid Size	$nx = 501,\ ny = 101$	Number of grid points
Reynolds Number	100	Governs fluid flow regime
Prandtl Number	0.71	For air at standard conditions
Kinematic Viscosity	$\nu = \frac{UL}{Re}$	Computed from $U = 1.0$ and Re
Time Step	$dt = 0.0005$	Fixed time step
Total Steps	$nt = 1000$	Total number of simulation steps
Left Wall Velocity	$u = 1.0,\ v = 0.0$	Inlet (Dirichlet)
Right Wall Velocity	Zero-gradient	Outlet (Neumann)
Top/Bottom Velocity	$u = v = 0.0$	No-slip wall (Dirichlet)
Temperature Top Wall	300 K	Dirichlet or Neumann (configurable)
Temperature Bottom Wall	30 K / Adiabatic	Dirichlet or Neumann (configurable)
Temperature Left Wall	300 K / Adiabatic	Dirichlet or Neumann (configurable)
Temperature Right Wall	270 K / Adiabatic	Dirichlet or Neumann (configurable)

of temperature and velocity on the horizontal and vertical center lines are as follows:

6.1.2 2D Channel

The relevant parameters and boundary conditions of this case in the solver are shown in the following table.

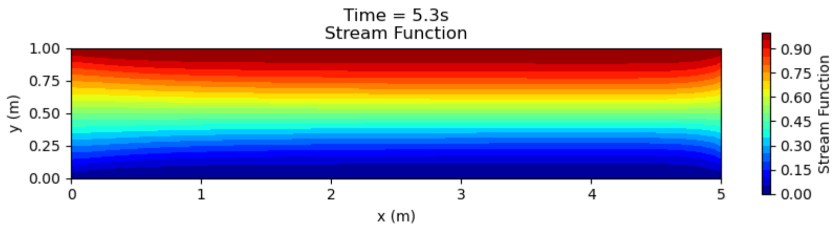


Figure 4: Channel diagram

The results of the two-dimensional temperature field and velocity field and One-dimensional line plot of temperature and velocity on the horizontal and vertical center lines are as follows:

6.2 Results Analysis

Overall, it can realize flexible switching of speed boundary conditions and temperature boundary conditions, and support users to flexibly modify size, Reynolds number, speed and temperature parameters.

However, there are still some problems with the solver. For example, when the channel switches to the lid, there are some obvious differences in the speed boundary between the two. Currently, I can only make the two switch at the speed boundary, but there are still some bugs in the boundary of the stream function.

Another point is the instability of the numerical method. The solver is currently best used for solving the test under low Reynolds number, low initial flow velocity, and low time step. Otherwise, the numerical method will be unstable. In the future, the solver will be upgraded to a more stable numerical method.

6.3 Reflection on the use of AI tools

Thermoflow2D Solver Pro was mainly developed using **ChatGPT 4o**, which mainly involved **code adjustment**, **GUI interface design**, **visualization result beautification**, and **Latex document editing**.

7 Future Improvements

- ▷ Currently, only the effect of temperature on the velocity field is considered, that is, the coupling between the temperature field and the velocity field is unidirectional. The temperature field is affected by the velocity field (through the convection term), and the velocity field is not affected by the temperature (no buoyancy term). In the future, we can consider developing a deep coupling solver.
- ▷ Consider using *Javascript* as the front end, using *http* as the interface, creating a web page, replacing the current GUI program, and avoiding the problem of lag.
- ▷ The inherent problems of instability of numerical methods and boundary conditions mentioned above can be improved subsequently.

8 Contact Us

Thermoflow2D Solver Pro is more than a tool—it's a gateway to innovation.

Visit `https://github.com/wjx0209/Thermoflow2D-Solver-Pro.git` to explore tutorials, case studies, and community forums.

Start simulating today!