

Best practice coding

We will demonstrate the best practice coding by computing the Taylor green vortex fields. Try to write a Python code that computes the Taylor-Green vortex fields both theoretically and numerically following the instructions given below.

Taylor-Green vortex:

It is an unsteady flow of decaying vortex, and there exists an exact closed form solution of incompressible Navier-Stokes equations in Cartesian coordinates for this Taylor-Green vortex flow. The stream function (ψ) and vorticity (ω) of the Taylor-Green vortex flow is given by,

$$\psi = \sin x \sin y e^{-2\nu t}$$

where, x, y are the Cartesian coordinates in the range $0 \leq x, y \leq 2\pi$; ν is kinematic viscosity, and t indicates time. Based on this stream function, the velocity fields can be derived as,

$$u = \frac{\partial \psi}{\partial y} = \sin x \cos y e^{-2\nu t} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x} = -\cos x \sin y e^{-2\nu t}.$$

The vorticity (ω) of the flow-field is given by,

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = -\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2} = 2 \sin x \sin y e^{-2\nu t} = 2\psi$$

Since the Taylor-Green vortex has an exact closed form analytical solution, often it is used as benchmark reference case for validating numerical models.

Coding steps:

1. Generate a uniformly spaced 2D Cartesian grid $0 \leq x, y \leq 2\pi$, having N_x and N_y number of grid points along x - and y -directions respectively. Consider $N_x \neq N_y$.
2. Compute the velocity, stream function and vorticity fields on the (x, y) grid using the theoretical expressions for $\nu = 0.1 \text{ m}^2/\text{s}$ at $t = 1.0 \text{ s}$.
3. Next, compute the velocity fields (u^* , v^*) from the stream function by performing numerical differentiation such that,

$$u^* = \frac{\partial \psi}{\partial y} = \begin{cases} \frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta y}, & j = 0 \\ \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta y}, & 0 < j < N_y - 2 \\ \frac{\psi_{i,j} - \psi_{i,j-1}}{\Delta y}, & j = N_y - 1 \end{cases}$$

and,

$$v^* = -\frac{\partial \psi}{\partial x} = \begin{cases} \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x}, & i = 0 \\ \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x}, & 0 < i < N_x - 2 \\ \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta x}, & i = N_x - 1. \end{cases}$$

4. Then compute the vorticity field based on the velocity field (u^* , v^*) as below,

$$\omega^* = \frac{\partial v^*}{\partial x} - \frac{\partial u^*}{\partial y}.$$

Evaluate the differentiations numerically following the step 3,

$$\frac{\partial v^*}{\partial x} = \begin{cases} \frac{v_{i+1,j}^* - v_{i,j}^*}{\Delta x}, & i = 0 \\ \frac{v_{i+1,j}^* - v_{i-1,j}^*}{2\Delta x}, & 0 < i < N_x - 2 \\ \frac{v_{i,j}^* - v_{i-1,j}^*}{\Delta x}, & i = N_x - 1, \end{cases}$$

and

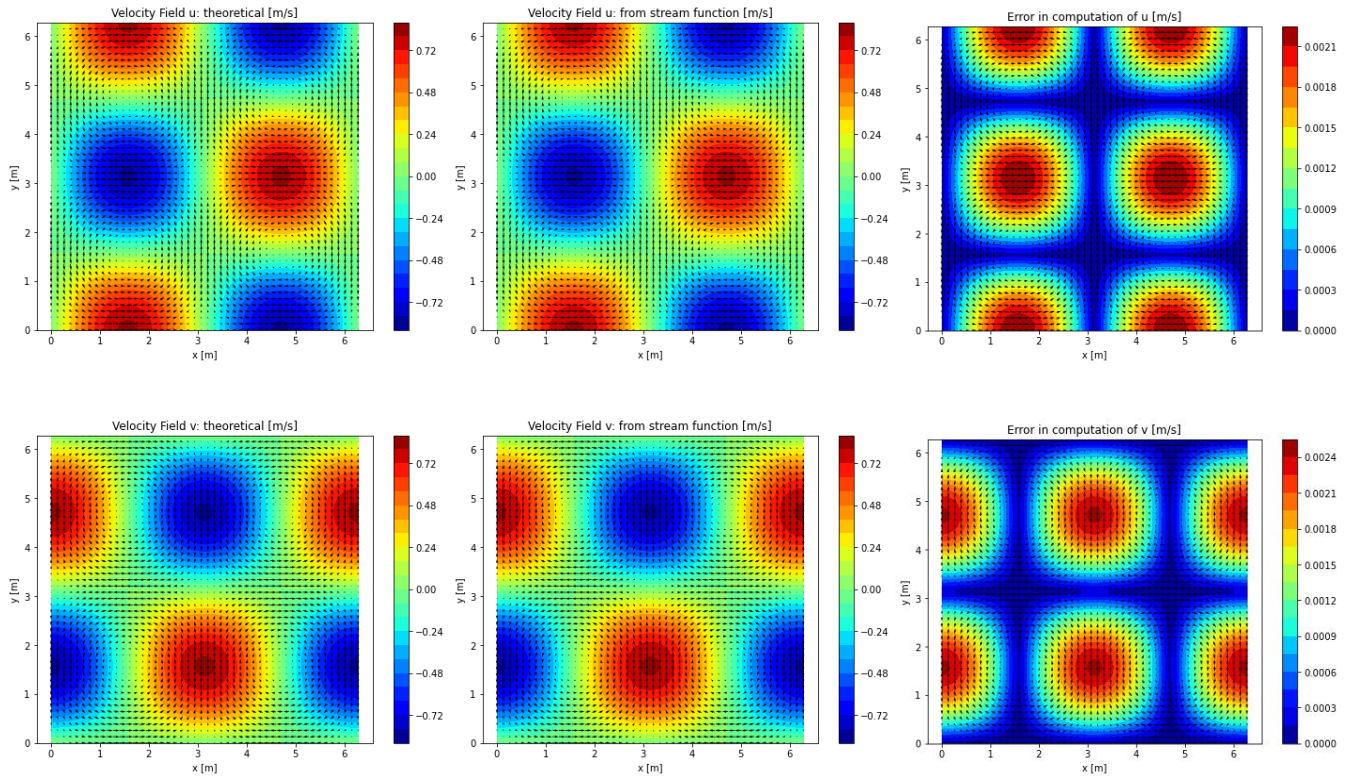
$$\frac{\partial u^*}{\partial y} = \begin{cases} \frac{u_{i,j+1}^* - u_{i,j}^*}{\Delta y}, & j = 0 \\ \frac{u_{i,j+1}^* - u_{i,j-1}^*}{2\Delta y}, & 0 < j < N_y - 2 \\ \frac{u_{i,j}^* - u_{i,j-1}^*}{\Delta y}, & j = N_y - 1. \end{cases}$$

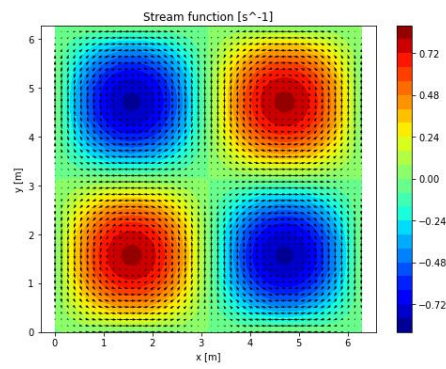
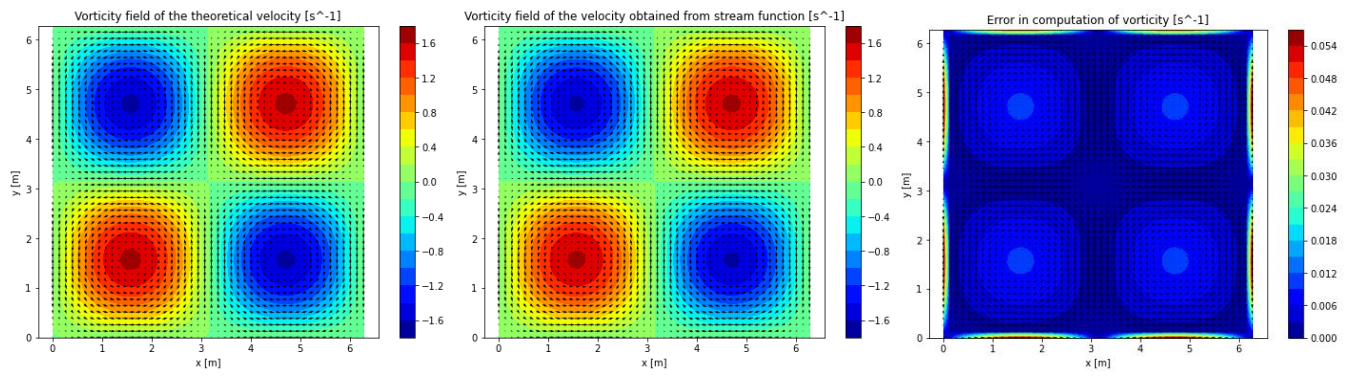
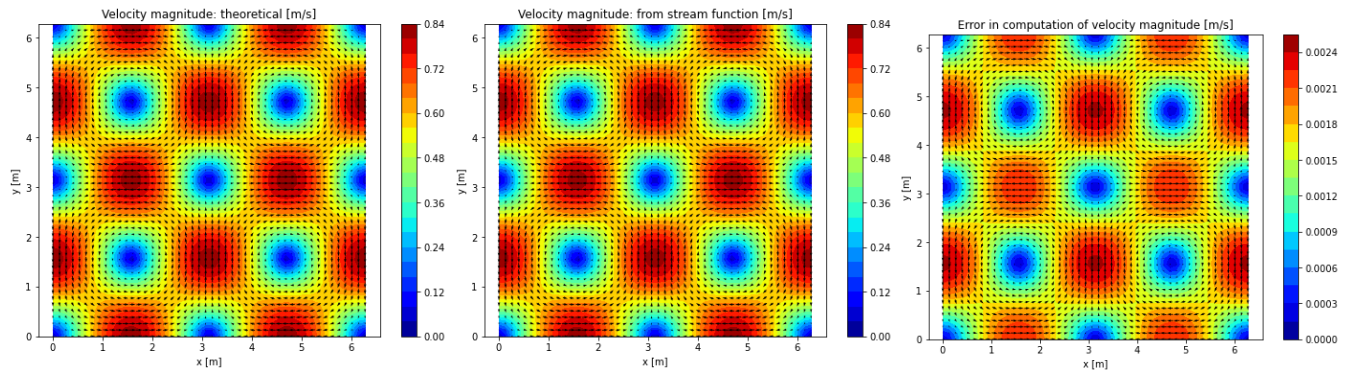
5. Now compute the error of the fields computed numerically with respect to their theoretical values,

$$\Phi_{error} = |\Phi - \Phi^*|,$$

where Φ indicates any field variable. Print the maximum error over the entire domain.

6. Finally, plot all the velocity and vorticity fields computed using both theoretical and numerical methods along with the error fields. Plot the contours along with velocity quiver.





BAD coding practice to avoid

No Comments or Documentation:

- Nothing is explained, making the code very difficult to follow.
- No description of input variables and their units
- Hard for readers to understand the meaning of the variables, functions, complex equations etc.

No Descriptive Variable Names:

- Poor Variable Naming: variables like `t`, `v`, and `v_th` are meaningless.
- Inconsistency in naming variables
- Similar symbols for different types of variables
- Abbreviated and unclear names make it hard to understand the logic.

Hardcoded Values:

- Parameters like grid size, viscosity, time, and colormap levels are hardcoded in multiple places.
- Constants like `pi` and `e` being hard coding

Poor Formatting:

- Code is not indented or aligned properly.
- Mixed case in print statements and lack of spacing for readability

Unformatted Plots:

- No axis labels, descriptive title, or consistent scaling.

No Functions:

- The code has no modularization, and repetitive logic is written out manually.
- **No parameter validation:** No validation for physical constraints (e.g., viscosity must be positive)
- Function is not generalized or validated for incorrect inputs.

Poor Flexibility and Scalability:

- Any change (e.g., resolution) requires editing multiple parts of the code.

Inefficient and Error-Prone:

- Explicit loops are computationally expensive for large grids.
- Code is harder to debug and maintain.

Repetition:

- Plotting blocks are repeated for every field.

No Error Handling:

- The code assumes everything will work without exceptions.