Homework Assignment 4

Ran Li

March 29, 2018

Part 1: Poisson Equation Solver

False-transient method is applied to solve following Poisson Equation:

$$\nabla^2 \phi = 2 + 3y \tag{1}$$

where boundary conditions are:

$$\phi(0, y) = \frac{1}{2}y^3, \phi(1, y) = 1 + 3y + \frac{1}{2}y^3$$

$$\phi(x, 0) = x^2, \phi(x, 1) = x^2 + 3x + \frac{1}{2}$$

For false-transient method, instead of steady nature of the given equation, we plugin an time derivative and assume it's a transient problem.

$$\frac{\partial \phi}{\partial t} = \nabla^2 \phi - 2 - 3y \tag{2}$$

Nature of the resulting parabolic equation will ensure convergence of the false-transient equation. To derive numerical scheme of this problem, we go simply to FTCS approach:

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^{k}}{\Delta t} = \frac{\phi_{i+1,j}^{k} - 2\phi_{i,j}^{k} + \phi_{i-1,j}^{k}}{\Delta x^2} + \frac{\phi_{i,j+1}^{k} - 2\phi_{i,j}^{k} + \phi_{i,j-1}^{k}}{\Delta y^2} + w_{i,j} \tag{3}$$

where the tail term w_{ij} is nothing but RHS of original equation,

$$w_{i,j} = -2 - 3y \tag{4}$$

To better parametricalize the problem, we simply reorganize discretizing parameters: $\Delta x, \Delta y, \Delta t$ into $\Delta x, \rho_1, \beta$ with following formula:

$$\rho_1 = \frac{\Delta t}{\Delta x^2}, \beta = \frac{\Delta x}{\Delta y} \tag{5}$$

Reorganize equation (3) into following format:

$$\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \rho_1[\phi_{i+1,j}^{k} - 2\phi_{i,j}^{k} + \phi_{i-1,j}^{k} + \beta^2(\phi_{i,j+1}^{k} - 2\phi_{i,j}^{k} + \phi_{i,j-1}^{k}) + \Delta x^2 w_{i,j}] \quad (6)$$

Computational Scheme is visualized as following picture and BCs are placed at corresponding locations. Structral grid with designated size (51×51) is applied and will be used for solving the problem numerically. Exact values of BCs are computed and implemented as Dirichlet boundary condition.

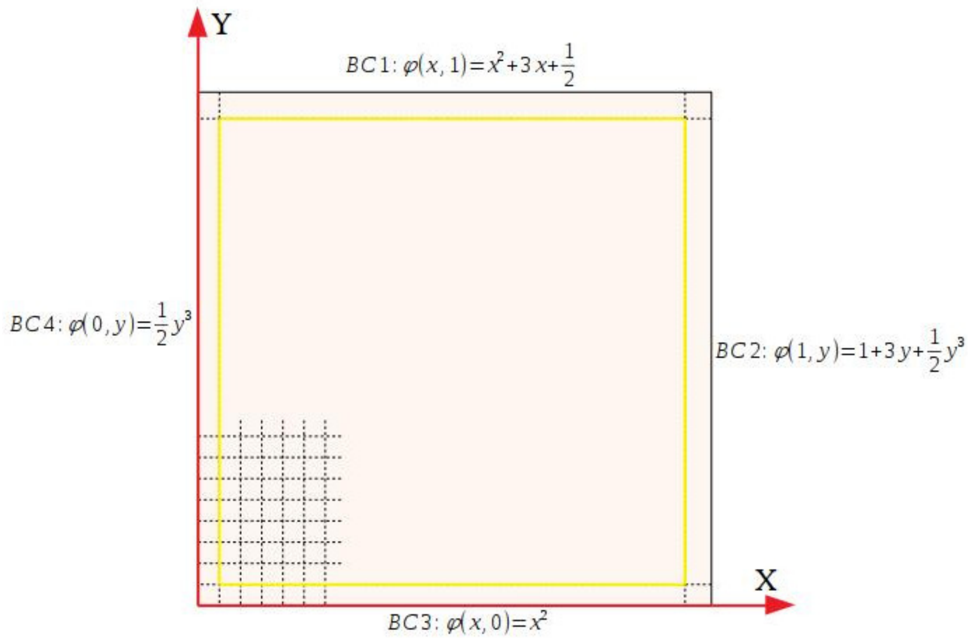

Figure 1: Computational Domain and Boundary Conditions

In final false-transient scheme (6), the tail term $\Delta x^2 w_{i,j}$ could be pre-calculated at each internal grid points(not needed on boundary since they are fixed by Dirichlet BCs). Then the time iterative computation is straight forward. By initializing all internal grid point i,j to have value of $\phi_{i,j} = 0$ and compute iteratively for next time step, we could easily solve for this problem.

Code for this problem is posted at the end of this report as appendix.

Sub-problem a>:

Conduct Neumann analysis to find out stability criterion of this specific problem by assuming solution $\phi_{i,j}^{k}$ have the following form:

$$\phi_{i,j}^{k} = \sigma^k e^{ax_i + by_j} \quad (7)$$

Substitute this back into our final false-transient scheme (6)

$$\sigma = 1 + \rho_1[(e^{a\Delta x} - 2 + e^{-a\Delta x}) + \beta^2(e^{b\Delta y} - 2 + e^{-b\Delta y}) + \Delta x^2 w_{i,j}] \tag{8}$$

To ensure that this algorithm is stable w.r.t. time iteration, magnitude σ shall satisfy following condtion:

$$|\sigma| \leq 1 \tag{9}$$

Substituting (8) into (9) and we will have:

$$|1 + \rho_1[(2\cos a\Delta x - 2) + \beta^2(2\sin b\Delta y - 2)] + \Delta t w_{i,j}| \leq 1 \tag{10}$$

Access each term on LHS: We could first assume that discretization quality are the same along x and y axises, which will result in $\Delta x = \Delta y = t, a = b = c$. Also since the tail term $\Delta t w_{i,j}$ contain an infinitestimally small term Δt while the other term is a “constant” , we could simply neglect it in our analysis of stability. Thus inequality (10) could be simplified:

$$|1 + 2\rho_1(1 + \beta^2)(\cos ct - 1)| \leq 1 \tag{11}$$

From property of cosine function we could further have:

$$1 - 4\rho_1(1 + \beta^2) \geq -1, \cos ct = -1$$

$$1 - 0 \leq 1, \cos ct = 1$$

Thus by rearranging terms in inequality above, we could have :

$$\rho_1 \leq \frac{1}{2(1 + \beta^2)} \tag{12}$$

So the given stability criterion is proven.

Sub-problem b>:

MATLAB code is generated to solve this 2D Poisson equation. Using the given grid size (51×51) to implement the scheme. Boundary conditions are calculated and included as Dirichlet BCs, while the initial ϕ field is set to 0 everywhere except for boundaries. Maximum error between 2 time steps is calculated at each time step and the iteration will not stop until it's value drop below 10^{-8}.

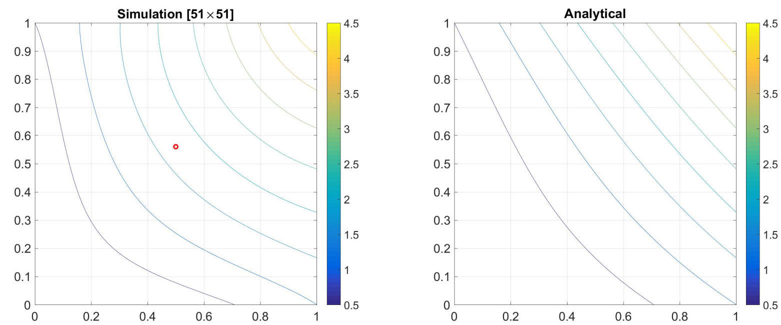

Figure 2: Simulation result V.S. Analytical Solution

In the figure, red dot indicates the location where error between simulation and analytical solution become maximum. Error distribution in the whole domain is given in figure below:

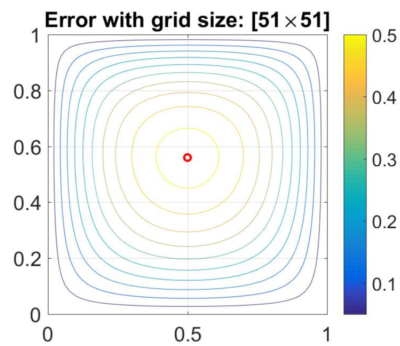

Figure 3: Error (between simulation and analytical result) distribution

Plot residue as a function of iteration number k down below. It's obvious that error is converging eventually to a very small value:

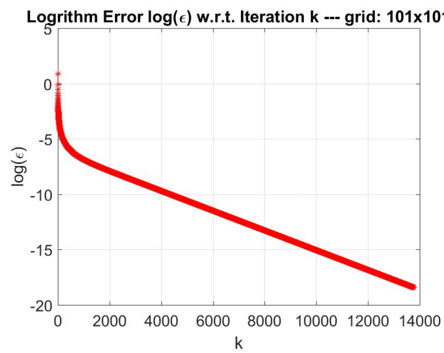

Figure 4: Residue with respect of iteration number

Sub-problem c>:

Improve grid quality to 101×101 and 201×201 respectively, compute for solution again using same protocle. Results from computations are shown down below:

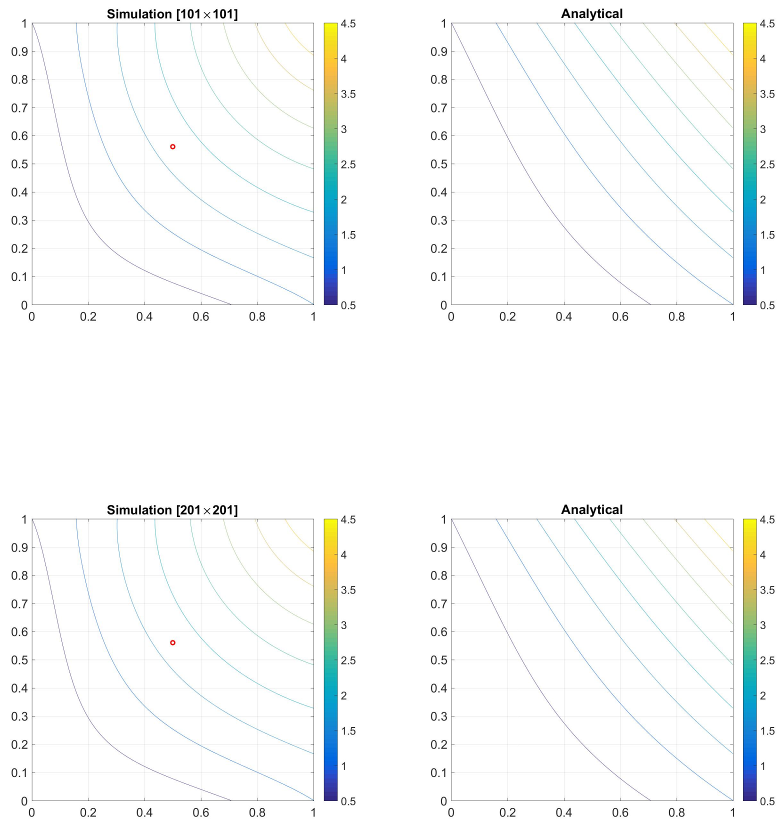

Figure 5: Simulation V.S. Analytical Solution with grid size of 101×101 and 201×201 respectively

Error distribution for the 2 cases are shown here:

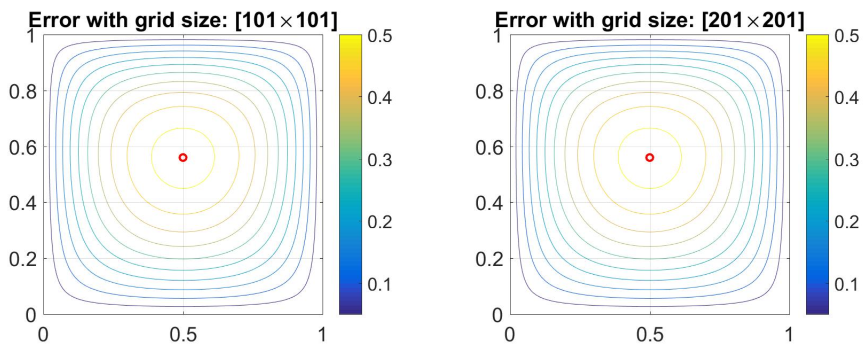

Figure 6: Error distribution for grid size of 101×101 and 201×201 respectively

Plotting maximum error as a function of number of points in either X or Y direction:

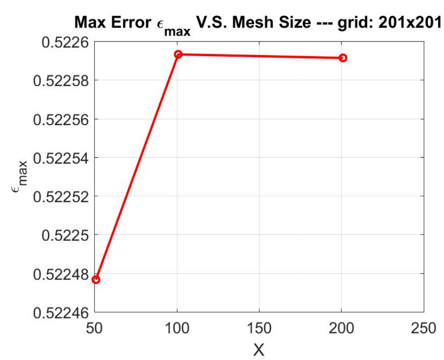

Figure 7: Maximum Error as function of grid quality

Part 2: Navier-Stokes Equation Solver

With the help of $\psi - \omega$ formulation, we could decompose the whole N-S equation into 2 Poisson equations. Then solve them alternatively using false-transient method to acquire solution.

Sub-problem a>:

Recall $\psi - \omega$ formulation of N-S equation:

$$\nabla^2 \psi = -\omega, \nabla^2 \omega = \frac{1}{\nu}\left(\frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}\right) \tag{13}$$

To non-dimensionlize the governing equation, reference parameter must be chosen first. In this case, we choose length of lid L, velocity of lid V as our referencial parameters(dimension). Thus each variable involved in the governing equation could be non-dimensionlized as follows:

$$x^* = \frac{x}{L}, y^* = \frac{y}{L}, u^* = \frac{u}{V}, v^* = \frac{v}{V}, \psi^* = \frac{\psi}{VL}, \omega^* = \frac{\omega}{V/L} \tag{14}$$

Thus subsitute them back to $\psi - \omega$ formulation:

$$\nabla^2\psi^* = -\omega^*, \nabla^2\omega^* = Re[\frac{\partial\psi^*}{\partial y^*}\frac{\partial\omega^*}{\partial x^*} - \frac{\partial\psi^*}{\partial x^*}\frac{\partial\omega^*}{\partial y^*}] \tag{15}$$

These will be our final non-dimensionized governing equations.

Sub-problem b>:

Code is given at the end of this report. Here I simply cover the governing equation, boundary conditions and description of algorithm.

To implement false-transient method, we add time derivative term to non-dimensionlized governing equations. Thus they are re-written as:

$$\frac{\partial\psi^*}{\partial t} = \nabla^2\psi^* + \omega^*, \frac{\partial\omega^*}{\partial t} = \frac{1}{Re}\nabla^2\omega^* - [\frac{\partial\psi^*}{\partial y^*}\frac{\partial\omega^*}{\partial x^*} - \frac{\partial\psi^*}{\partial x^*}\frac{\partial\omega^*}{\partial y^*}] \tag{16}$$

Apply simple FTCS difference scheme to false-transient equations above to acquire the formulation for actual computation:

$$\psi_{i,j}^{*k+1} = \psi_{i,j}^{*k} + \rho_1[\psi_{i+1,j}^{*k} - 2\psi_{i,j}^{*k} + \psi_{i-1,j}^{*k} + \beta^2(\psi_{i,j+1}^{*k} - 2\psi_{i,j}^{*k} + \psi_{i,j-1}^{*k}) + \Delta x^2\omega_{i,j}^*] \tag{17}$$

$$\omega_{i,j}^{*k+1} = \omega_{i,j}^{*k} + \rho_1[\frac{1}{Re}(\omega_{i+1,j}^{*k} - 2\omega_{i,j}^{*k} + \omega_{i-1,j}^{*k}) + \frac{1}{Re}\beta^2(\omega_{i,j+1}^{*k} - 2\omega_{i,j}^{*k} + \omega_{i,j-1}^{*k}) + \Delta x^2 N_{i,j}^*] \tag{18}$$

where:

$$N = -[\frac{\psi_{i,j+1}^{*k+1} - \psi_{i,j-1}^{*k+1}}{2\Delta y} \cdot \frac{\omega_{i+1,j}^{*k} - \omega_{i-1,j}^{*k}}{2\Delta x} + \frac{\psi_{i-1,j}^{*k+1} - \psi_{i+1,j}^{*k+1}}{2\Delta x} \cdot \frac{\omega_{i,j+1}^{*k} - \omega_{i,j-1}^{*k}}{2\Delta y}] \tag{19}$$

Boundary condition could be then designated following basic physical facts(no penetration and non-slip at boundaries) and non-dimensionlized nature:

- Non-slip Boundary Conditions:

$$\frac{\partial\psi^*}{\partial x^*}(0, y^*) = 0, \frac{\partial\psi^*}{\partial y^*}(x^*, 0) = 0, \frac{\partial\psi^*}{\partial x^*}(1, y^*) = 0, \frac{\partial\psi^*}{\partial y^*}(x^*, 1) = 1 \tag{20}$$

- No penetration Boundary Conditions:

$$\frac{\partial\psi^*}{\partial y^*}(0, y^*) = 0, \frac{\partial\psi^*}{\partial x^*}(x^*, 0) = 0, \frac{\partial\psi^*}{\partial y^*}(1, y^*) = 0, \frac{\partial\psi^*}{\partial x^*}(x^*, 1) = -\Delta y \tag{21}$$

- Constant ψ^* on boundary:

$$\psi^*(0,y^*) = \psi^*(1,y^*) = \psi^*(x^*,0) = \psi(x^*,1) = 0 \tag{22}$$

We follow following procedure to solve for this problem:

1. Initialize ψ^* and ω^* lattice by assign all 0 values everywhere. Plug in boundary conditions for ψ^* (22);
2. Solve the first false-transient equation (17);
3. Apply BCs to ω^* lattice using equations on page of 65 in our Note;
4. Differentiate ψ^* and ω^* spatially and calculate the tail term N^* in internal nodes;
5. Solve the second false-transient equation (18);
6. Keep iterate until maximum residue errors of ψ^* and ω^* both satisfies: $\epsilon_{\psi^*}^{k+1} = |\psi^{*k+1} - \psi^{*k}| \leq 10^{-8}$ and $\epsilon_{\omega^*}^{k+1} = |\omega^{*k+1} - \omega^{*k}| \leq 10^{-8}$

Sub-problem c>:

Solution of situation where $Re = 1$:

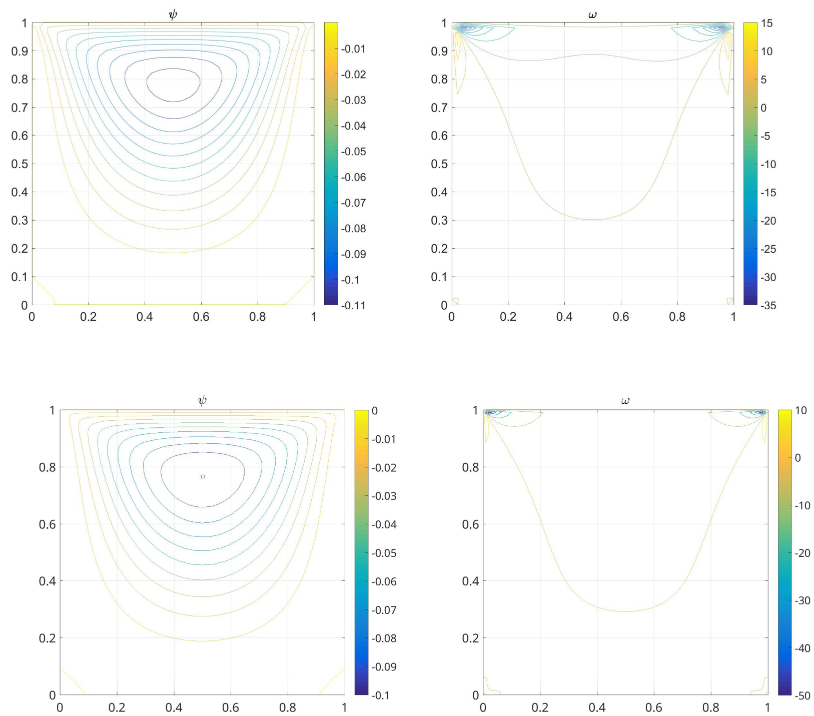

Figure 8: Solution from my code using 2 different grid sizes, top is 51×51, bottom is 101×101

Vorticity values at given vertical lines:

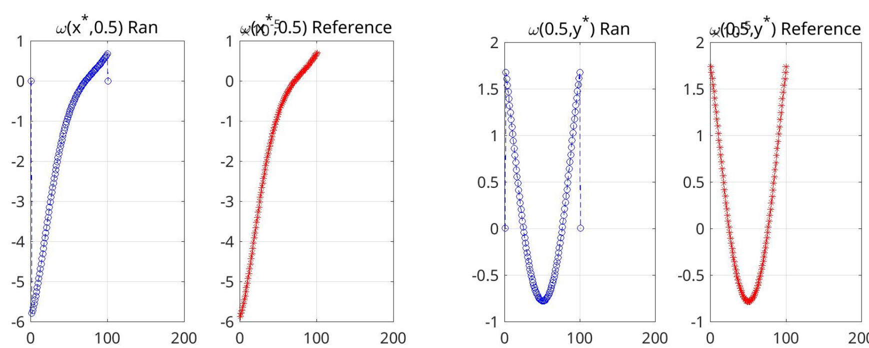

Figure 9: Vorticity Comparison

Comments:

Sub-problem d>:

Solution of situation where $Re = 100$:

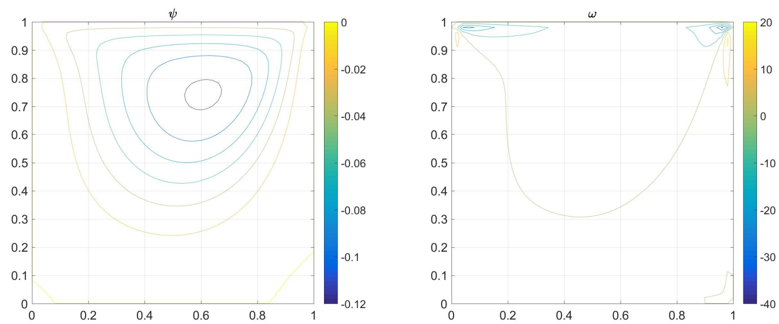

Figure 10: Grid size of 101×101

Vorticity values at given points from my computation results:

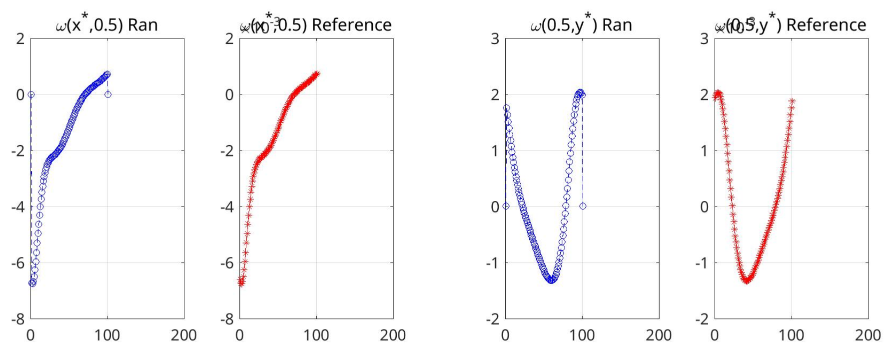

Figure 11: Vorticity Comparison

Sub-problem e>:

Solution of situation where $Re = 1000$:

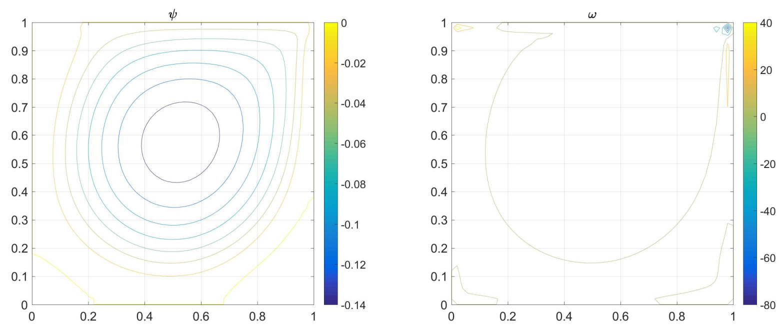

Figure 12: Grid size of 101 × 101

Vorticity values at given points from my computation results:

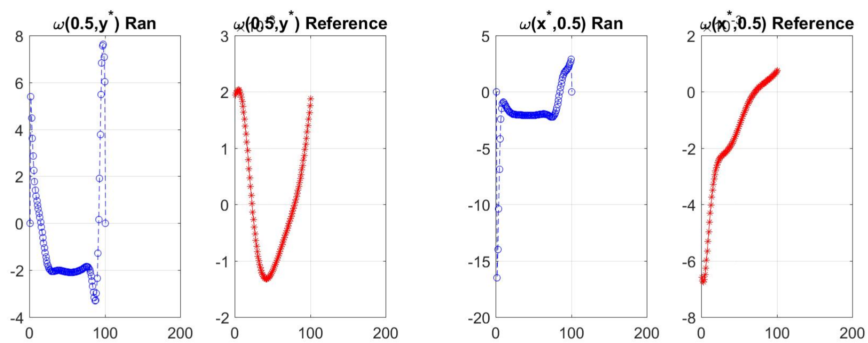

Figure 13: Vorticity Comparison