ASE 382Q11 Computational Fluid Dynamics Final Project: *Finite-Volume Lid-Driven Cavity on Staggered Cartesian Grid*

Tristan Lee

October 18 2019

The incompressible Navier-Stokes equations were solved on a two dimensional staggered Cartesian grid for the lid-driven cavity problem using the finite volume method and a predictor-corrector algorithm. The order of convergence of the system was verified, and the results of simulations were favorably compared to published values.

Introduction

The lid-driven cavity (LDC) is a standard benchmark in computational fluid dynamics. In its simplest form, this system consists of a square cavity with side lengths L_{lid} that has been discretized in the x and y dimensions with spacing Δx and Δy respectively, with the top of the cavity having a fixed velocity U_{lid}. A diagram of the LDC system and the mentioned parameters is shown in Figure 1.

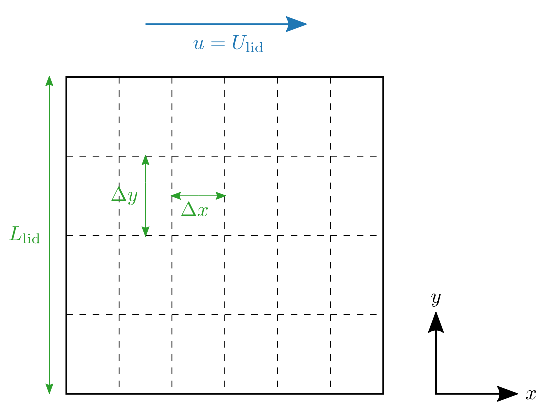

Figure 1: Lid-driven cavity system diagram and associated parameters

Introducing the additional assumptions of constant density and viscosity simplifies the problem further: the energy conservation equation becomes trivial, so only the momentum equations and incompressibility constraint remain. Imposing a constant viscosity also enables a Reynolds number (Re) to be specified globally. The steady-state solution of this system depends on the value of the system's Reynolds number.

The lid-driven cavity problem can be solved in a variety of ways. One common approach is to apply the finite volume formalism to the Navier-Stokes equations.

Background

In two dimensions, the nondimensionalized incompressible Navier-Stokes equations are given by:

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{0.1}$$

$$\frac{\partial v}{\partial t} + \frac{\partial v^2}{\partial y} + \frac{\partial uv}{\partial x} = -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{0.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{0.3}$$

where u and v are the components velocity along the x and y dimensions respectively, p is the pressure, and Re is the system's Reynolds number, given by:

$$\text{Re} = \frac{\rho' U_{\text{lid}} L_{\text{lid}}}{\mu'}, \tag{0.4}$$

where ρ' and μ' are the dimensional density and dynamic viscosity, and U_{lid} and L_{cav} are the velocity and size of the lid for the lid-driven cavity. Velocity quantities (u and v) have been normalized by the velocity of the lid (U_{lid}), spatial quantities (x and y) have been normalized by the side length of the cavity (L_{cav}), density and dynamic viscosity have been normalized by reference values (ρ_{ref} and μ_{ref}), and pressure has been normalized as:

$$p = \frac{p' - p_{\text{ref}}}{\rho' U_{\text{lid}}^2}. \tag{0.5}$$

It is convenient to solve this problem on a staggered grid, which consists of a primary grid and a dual grid, which is comprised of by north (N), south (S), east (E), and west (W) vertices surrounding a given point. A diagram of the staggered grid geometry is shown in Figure 2.

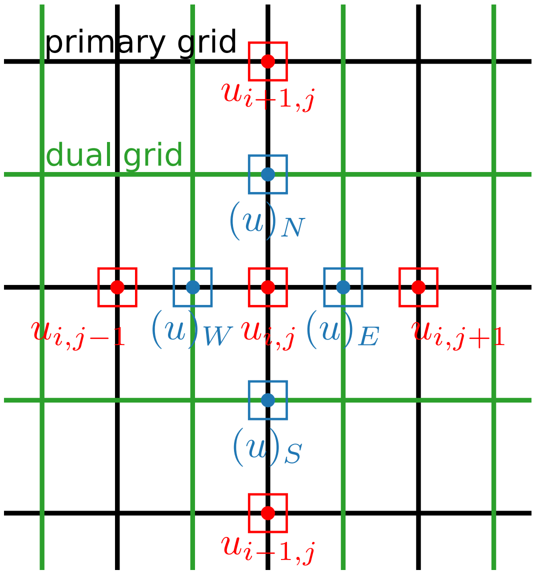

Figure 2: Staggered grid diagram

For the dual grid, the velocities and velocity derivatives are defined as:

$$(u)_N = \frac{1}{2}(u_{i,j} + u_{i,j+1}); \quad \left(\frac{\partial u}{\partial y}\right)_N = \frac{u_{i,j+1} - u_{i,j}}{\Delta y} \tag{0.6}$$

$$(u)_S = \frac{1}{2}(u_{i,j} + u_{i,j-1}); \quad \left(\frac{\partial u}{\partial y}\right)_S = \frac{u_{i,j} - u_{i,j-1}}{\Delta x} \tag{0.7}$$

$$(u)_E = \frac{1}{2}(u_{i,j} + u_{i+1,j}); \quad \left(\frac{\partial u}{\partial x}\right)_E = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \tag{0.8}$$

$$(u)_W = \frac{1}{2}(u_{i,j} + u_{i-1,j}); \quad \left(\frac{\partial u}{\partial x}\right)_W = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}. \tag{0.9}$$

All terms in Equation 0.1 can then be discretized to the first order in space with the finite-volume formalism by using the dual grid quantities defined above, with equivalent expressions for v expressed similarly:

$$\frac{\partial u^2}{\partial x} \approx \frac{(u^2)_E - (u^2)_W}{\Delta x} \tag{0.10}$$

$$\frac{\partial uv}{\partial y} \approx \frac{(uv)_N - (uv)_S}{\Delta y} \tag{0.11}$$

$$\frac{\partial p}{\partial x} \approx \frac{p_{i,j} - p_{i-1,j}}{\Delta x} \tag{0.12}$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\left(\frac{\partial u}{\partial y}\right)_E - \left(\frac{\partial u}{\partial y}\right)_W}{\Delta x} \tag{0.13}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{\left(\frac{\partial u}{\partial y}\right)_N - \left(\frac{\partial u}{\partial y}\right)_S}{\Delta y}. \tag{0.14}$$

A common approach for solving this kind of problem is a *predictor-corrector* method, in which an intermediate solution is *predicted* without taking the pressure gradient into account, then that intermediate solution is *corrected* by solving for pressure and subtracting the pressure gradient. For this project, the two-step Adams Bashforth was used as the predictor step, given by:

$$\tilde{u}^{k+1} = u^k + \frac{\Delta t}{2}\big(3f(u^k) - f(u^{k-1})\big), \tag{0.15}$$

where the governing equation is given by $\dot{u} = f(u)$, and the tilde over u^{k+1} indicates that its only the intermediate solution. For this system, f can be found by moving all spatial derivative terms in Equation 0.1 to the RHS.

The correction is generated by computing the pressure field such that the Poisson equation is satisfied, given the predicted velocity field:

$$\begin{aligned}&\frac{1}{\Delta x}\left(\left(\tilde{u}_{i+1,j} - \Delta t\frac{p_{i+1,j} - p_{i,j}}{\Delta x}\right) - \left(\tilde{u}_{i,j} - \Delta t\frac{p_{i,j} - p_{i-1,j}}{\Delta x}\right)\right)\\ &+\frac{1}{\Delta y}\left(\left(\tilde{v}_{i,j+1} - \Delta t\frac{p_{i,j+1} - p_{i,j}}{\Delta y}\right) - \left(\tilde{v}_{i,j} - \Delta t\frac{p_{i,j} - p_{i,j-1}}{\Delta y}\right)\right)\\ &= 0.\end{aligned} \tag{0.16}$$

Equation 0.16 can be solved for the pressure field by using the alternating direction implicit (ADI) method to solve the equation:

$$A_P p_P + \sum_{\ell} A_\ell p_\ell = Q_{i,j}, \tag{0.17}$$

where:

$$A_p = -\sum_{\ell} A_\ell = -2\left(\frac{\Delta x}{\Delta y} + \frac{\Delta y}{\Delta x}\right), \tag{0.18}$$

$\ell = \{N, S, E, W\}$ is the set of directions on the dual grid, and

$$Q_{i,j} = \frac{1}{\Delta t}\big((\tilde{u}_{i+1,j} - \tilde{u}_{i,j})\Delta y + (\tilde{v}_{i,j+1} - \tilde{v}_{i,j})\Delta x\big) \tag{0.19}$$

is the specified solution to the ADI problem.

After the corrector pressure is calculated, it is used to correct the prediction by subtracting the pressure gradient from $\tilde{u}$ and $\tilde{v}$:

$$u_{i+1,j} = \tilde{u}_{i+1,j} - \Delta t\frac{p_{i+1,j} - p_{i,j}}{\Delta x} \tag{0.20}$$

$$u_{i,j} = \tilde{u}_{i,j} - \Delta t\frac{p_{i,j} - p_{i-1,j}}{\Delta x} \tag{0.21}$$

$$v_{i,j+1} = \tilde{v}_{i,j+1} - \Delta t\frac{p_{i,j+1} - p_{i,j}}{\Delta y} \tag{0.22}$$

$$v_{i,j} = \tilde{v}_{i,j} - \Delta t\frac{p_{i,j} - p_{i,j-1}}{\Delta y}. \tag{0.23}$$

Methods

The solver was implemented as a Python package, called `LidDrivPy`, using the object-oriented paradigm. The implementation of array and matrix operations made use of the `NumPy` Python package. The ADI method employed the Thomas algorithm for solving tridiagonal matrices.

The pseudocode of the main simulation loop is shown in Algorithm 1.

Algorithm 1: Simulation pseudocode

input : N_{cells}: number of cells
Re: Reynolds number
u_{lid}: lid velocity
L_{lid}: cavity size
Δt: time-step
ω: ADI relaxation factor
N_t: maximum number of time-steps
ϵ_t: outer loop tolerance
N_p: maximum number of ADI iterations
ϵ_p: ADI loop tolerance
output: u, v, p that satisfy the steady-state solution of the lid-driven cavity problem with the specified parameters
for $k \leftarrow 0$ **to** N_t **do**
  1. Compute predictions $\tilde{u}^k, \tilde{v}^k$ using previous solutions $\tilde{u}^{k-1}, \tilde{v}^{k-1}$
  2. Use ADI to compute p^k that satisfies the Poisson equation given $\tilde{u}^k$ and $\tilde{v}^k$.
  3. Correct $\tilde{u}^k$ and $\tilde{v}^k$ to get u^k and v^k
  if $|u^k - u^{k-1}| < \epsilon_t$ **and** $|v^k - v^{k-1}| < \epsilon_t$ **then**
    break
  else
    continue;
  end
end

The simulation algorithm was used to generate a variety of solutions with different values for N_{cells} and Re. In practice, it was found that larger values of Re require smaller values of Δt and ϵ_p to converge successfully. In cases where the solution failed to converge, the time-step and/or ADI tolerance were decreased, which generally resulted in successful convergence.

`LidDrivPy` was developed using standard Python packaging conventions, and includes unit tests (in the `tests/` directory), examples (in the `examples/` directory), and HTML-formatted documentation (in the `docs/` directory). The documentation was generated using the *Sphinx* documentation generator.

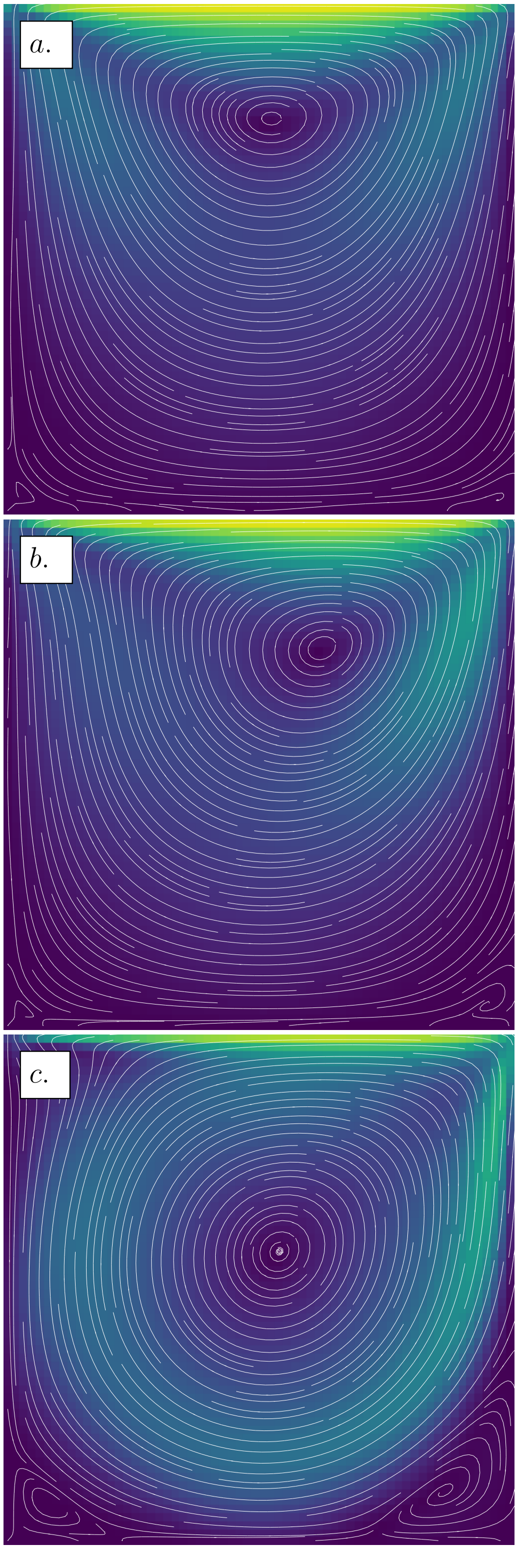

Figure 3: Stream plots overlain on velocity magnitude heatmaps, for $N_{\text{cells}} = 64$ and (a.) Re = 10 , (b.) Re = 100, and (c.) Re = 1000. All 3 heatmaps have same color scale.

Results

The LDC system was solved for multiple values of Re and N_{cells}. Figure 3 shows stream plots overlain on velocity magnitude heatmaps for values of Re equal to 10, 100, and 1000.

The plots in Figure 3 qualitatively match those found in other sources.

The convergence of the solver was tested by computing the steady-state solution of the LDC system for a given Reynolds number, for different values of N_{cells}. The values of N_{cells} were taken to be:

$$N_{\text{cells}} = 2^i + 1, \quad i \in [2, 3, 4, 5, 6, 7], \tag{0.24}$$

so that all arrays shared common grid locations. A plot of the center x slices of the steady-state u for $\text{Re} = 10$ for the specified values of N_{cells} is shown in Figure 4, and a similar plot for the center y slices of the steady-state v is shown in Figure 5.

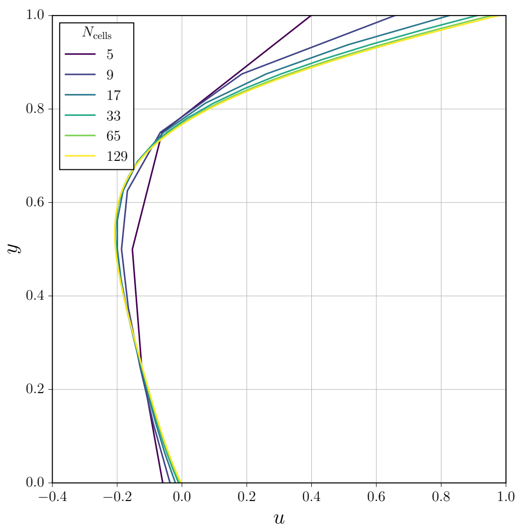

Figure 4: Convergence of center x slice of steady-state u for $\text{Re} = 10$, for different values of N_{cells} (y vs. u case).

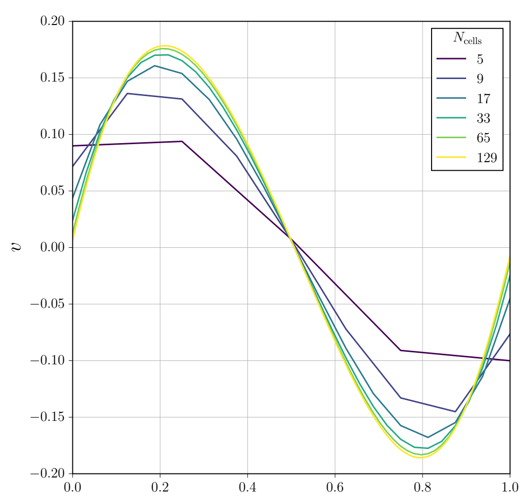

Figure 5: Convergence of center y slice of steady-state v for $\text{Re} = 10$, for different values of N_{cells} (x vs. v case).

The error as a function of N_{cells} was computed by taking the solution for $N_{\text{cells}} = 129$ as the "exact" solution, taking the same 5 evenly-spaced locations along the center x slice of the steady-state u for each value of N_{cells}, and taking the L^2 norm of the difference with the "exact" solution, i.e. for the y vs. u case:

$$\epsilon_N = ||u(N)_{N//2,::N//4} - u(129)_{64,::32}||_2, \tag{0.25}$$

and for the x vs. v case:

$$\epsilon_N = ||v(N)_{::N//4,N//2} - v(129)_{::32,64}||_2, \tag{0.26}$$

where $u(N)$ denotes the steady-state solution of u for $N_{\text{cells}} = N$, $//$ denotes integer division, and $::N//4$ indicates that values of the array were sampled every $N//4$ elements.

Figure 6 shows the error vs. N_{cells} for both the y vs. u and the x vs. v cases, as well as a best fit line in log-log space and the resulting slope in the legend.

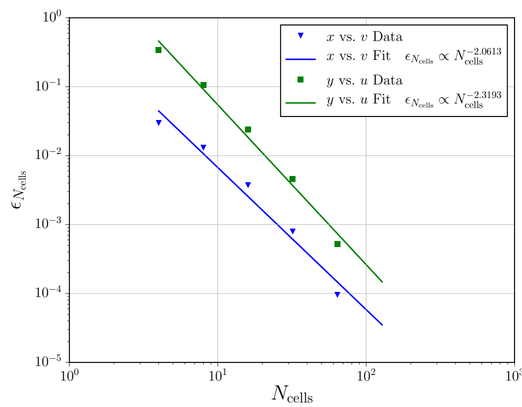

Figure 6: Error vs. N_{cells} for $\text{Re} = 10$

The order of convergence for both x vs. v and y vs. u is approximately 2, which is consistent with the two-step Adams Bashforth method that was employed in the prediction step being second-order. The convergence order of the x vs. v case is closer to the expected value than the y vs. u case, which may indicate an anisotropy in convergence order.