# Steady-State Stokes Equations For Impressible Flow

Caiyou Yuan

January 31, 2019

# 1 Algorithm

## 1.1 Continuous Equations

2D steady-state Stokes equations for impressible flow are

$$\begin{cases} -\Delta\vec{u} + \nabla p &= \vec{F} \\ \nabla\cdot\vec{u} &= 0 \end{cases} \tag{1}$$

where $\vec{u} = [u, v]^T$,$\vec{F} = [f, g]^T$. Sometimes, it's preferable to rewrite the above equations as

$$\begin{cases} -\Delta u & + & \partial_x p & = & f \\ -\Delta v & + & \partial_y p & = & g \\ \partial_x u & + & \partial_y v & = & 0 \end{cases} \tag{2}$$

## 1.2 Finite Difference Equations

Discretize (2), we get

$$\begin{cases} (-\Delta)_h u_h & + & (\partial_x)_h p_h & = & f_h \\ (-\Delta)_h v_h & + & (\partial_y)_h p_h & = & g_h \\ (\partial_x)_h u_h & + & (\partial_y)_h v_h & = & 0 \end{cases} \tag{3}$$

where $(\cdot)_h$ denotes the discretization on the uniform grid whose grid size is $h$. We can rewrite (3) in matrix form, so we'll see it's actually a saddle points problem.

$$\begin{bmatrix} A_h & B_h \\ B_h^T & 0 \end{bmatrix} \begin{bmatrix} U_h \\ p_h \end{bmatrix} = \begin{bmatrix} F_h \\ 0 \end{bmatrix} \tag{4}$$

where

$$A_h = \begin{bmatrix} (-\Delta)_h & 0 \\ 0 & (-\Delta)_h \end{bmatrix} \quad B_h = \begin{bmatrix} (\partial_x)_h \\ (\partial_y)_h \end{bmatrix} \quad U_h = \begin{bmatrix} u_h \\ v_h \end{bmatrix} \tag{5}$$

Here we discretize (2) on an uniform staggered grid. In 2D case, the grid define cells, and each cell has 4 face. The discretion of x-component velocity $u$ is defined at the center of x-faces of cells, i.e., faces perpendicular to the x-coordinate. The discretion of y-component velocity $v$ is defined at the y-faces. Discretion of pressure $p$ is located at the cell centers.

On sucn a staggered grid, we discretize the x/y-component momentum equation in x/y-face centers, and impressible condition $\nabla\cdot\vec{u} = 0$ at cell centers. Laplacian operator is approximated by five points scheme, while partial derivative is approximated by center difference.

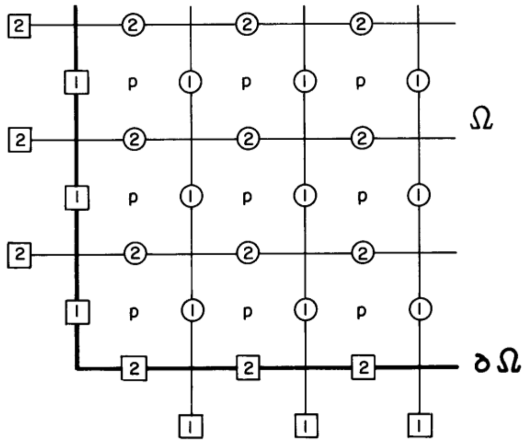


Figure 1: Staggered Grid in 2D ( Location 1 store $u$, Location 2 store $v$)

### 1.3 Distributive Gauss-Seidal Method

Conventional Gauss-Seidal Method is not appliable for solving finite difference equation (4), for its coefficient matrix has zero diagonal elements. So, here we use the so-called DGS method. DGS method firstly conducts Gauss-Seidal procedure for components $u_h$ and $v_h$. Then, for each cell,

1. Compute current residual

$$r_{ij} = F^p_{ij} - [(\partial_x)_h u_{ij} + (\partial_y)_h v_{ij}]$$

   Although here $F^p_{ij} = 0$ actually, we don't omit it for the multigrid method [1].

2. Eliminate current residual

$$\begin{aligned} u_{i+1/2,j} &= u_{i+1/2,j} + \delta; \\ u_{i-1/2,j} &= u_{i-1/2,j} - \delta; \\ v_{i+1/2,j} &= v_{i+1/2,j} + \delta; \\ v_{i-1/2,j} &= v_{i-1/2,j} - \delta; \end{aligned} \tag{6}$$

   where $\delta = r_{ij} h/4$

3. Update pressure $p$

$$\begin{aligned} p_{i,j} &= p_{i,j} + 4\delta/h; \\ p_{i+1,j} &= p_{i+1,j} - \delta/h; \\ p_{i-1,j} &= p_{i-1,j} - \delta/h; \\ p_{i,j+1} &= p_{i,j+1} - \delta/h; \\ p_{i,j-1} &= p_{i,j-1} - \delta/h; \end{aligned} \tag{7}$$

   The pressure changes are in the way such that momentum-equations residuals at all points remain unchange. If the cell is located near the boundary, the above procedure has some tiny modifications. All situations are graphically presented as Figure 2.

[1]In multigrid method, we have to solve the residual equations on the coarser grid, and now the right hand side on the $p$ component is not zero probably.

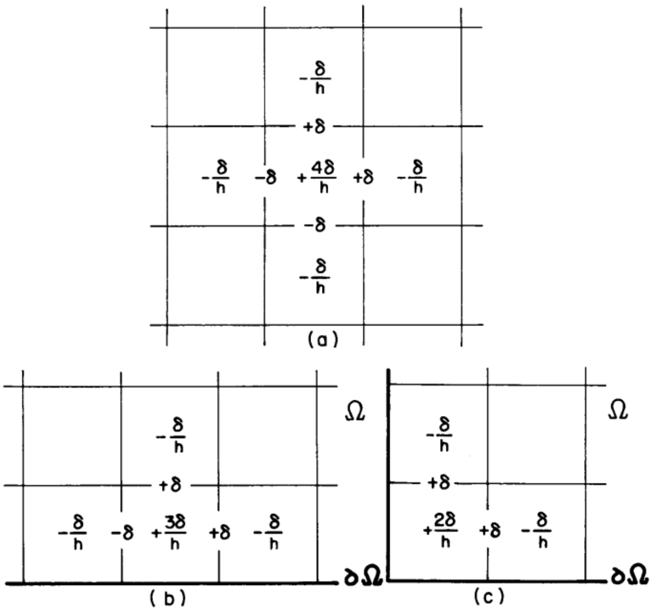


Figure 2: DGS Update Cases

### 1.4 Multigrid Technique

Just as conventional Gauss-Seidal method, DGS has the same limits:

1. Converge speed is slow;
2. Converge speed declines when grid gets finer.

Multigrid Method overcome such limits, get a much better result. See the results.

## 2 Numerical Test

Set problem region $\Omega = [0,1]^2$

$$\begin{array}{rl} f(x,y) = & -4\pi^2(2\cos(2\pi x)-1)\sin(2\pi y)+x^2 \\ g(x,y) = & \quad 4\pi^2(2\cos(2\pi y)-1)\sin(2\pi x) \end{array} \tag{8}$$

So the true solution for (2) is

$$\begin{array}{rl} u(x,y) = & \quad (1-\cos(2\pi x))\sin(2\pi y) \\ v(x,y) = & -(1-\cos(2\pi y))\sin(2\pi x) \\ p(x,y) = & \quad x^3/3-1/12 \end{array} \tag{9}$$

The boundary condition is given accordingly. Numerical test results are listed as followed,

| | DGS V-Cycle Results | | |
|---|---|---|---|
| Num | n=128 | n=256 | n=512 |
| 1 | 8.31191e+00 | 1.04701e+01 | 1.37668e+01 |
| 2 | 5.14701e-01 | 5.93665e-01 | 7.21926e-01 |
| 3 | 3.79273e-02 | 4.12965e-02 | 4.67891e-02 |
| 4 | 3.02427e-03 | 3.18206e-03 | 3.42251e-03 |
| 5 | 2.53225e-04 | 2.61750e-04 | 2.72443e-04 |
| 6 | 2.18033e-05 | 2.23540e-05 | 2.28567e-05 |
| 7 | 1.90423e-06 | 1.94612e-06 | 1.97266e-06 |
| 8 | 1.67320e-07 | 1.70904e-07 | 1.72546e-07 |
| 9 | 1.47222e-08 | 1.50547e-08 | 1.51765e-08 |
| 10 | 1.29381e-09 | 1.32663e-09 | 1.34501e-09 |
| L2 Error | 4.07741e-04 | 1.01930e-04 | 2.54820e-05 |
| Time/s | 1.02915e-01 | 3.85292e-01 | 1.69065e+00 |

The L2 errors confirm second order convergence. And compare results from case $n = 128, 256, 512$, we'll see multigird method has a rapid convergence rate which is irrelavent with the problem size.