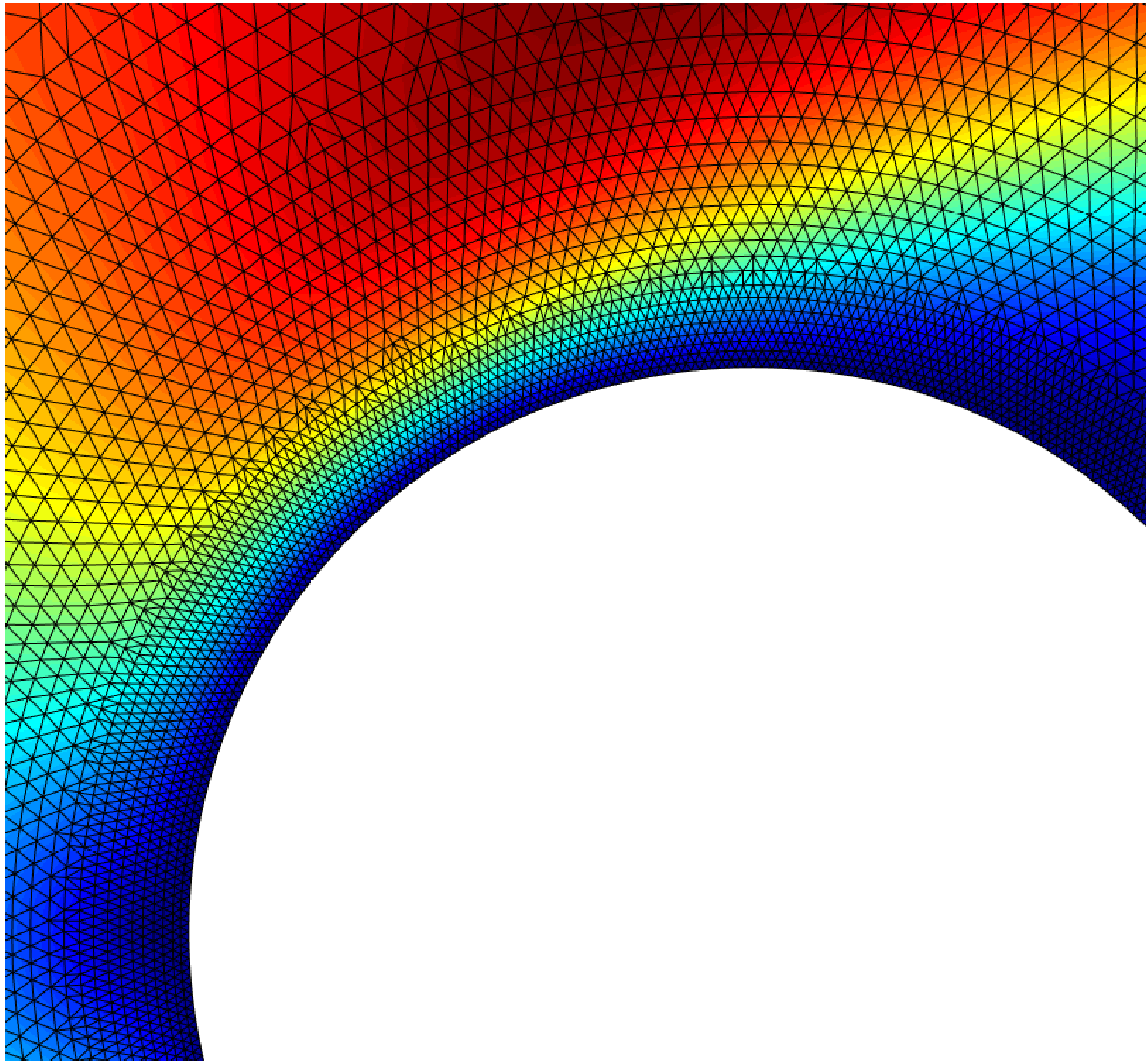
Finite Element Solver for the 2D Steady, Laminar Navier-Stokes Equations



Faisal As'ad

2.29 Project

Goal

To create from scratch a 2D finite element Navier-Stokes solver for incompressible, laminar, steady flows.

Motivation

Combined interest in:

- Finite element methods
- External fluid flows
- Unstructured grids

Programming platform:

MATLAB

Meshing Software:

SU2 (Stanford University Unstructred)

Test Case:

External flow over a cylinder (Re<~100)

Governing Equations

Navier-Stokes conservation of mass, x-momentum, y-momentum:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot [\rho \mathbf{u} \otimes \mathbf{u}] + \nabla p - \nabla \cdot \overline{\overline{\boldsymbol{\tau}}} = 0$$

- Incompressible flow $\rightarrow$ divide out ρ
- Steady $\rightarrow \dfrac{\partial()}{\partial t} = 0$
- Assume Newtonian Fluid $\rightarrow \boldsymbol{\nabla} \cdot \overline{\overline{\tau}} = \mu \boldsymbol{\nabla}^2 \mathbf{u}$

Now we have:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0$$

$$\boldsymbol{\nabla} \cdot [\mathbf{u} \otimes \mathbf{u}] + \frac{1}{\rho} \nabla p - \nu \boldsymbol{\nabla}^2 \mathbf{u} = 0$$

After non-dimensionalizing with $V_{ref} = V_\infty$, $L_{ref} = D$, $p_{ref} = \frac{\mu V_\infty}{D}$, finally:

$$\nabla \cdot \mathbf{u} = 0$$

$$Re(\nabla \cdot [\mathbf{u} \otimes \mathbf{u}]) + \nabla p - \nabla^2 \mathbf{u} = 0$$

Three equations, three unknowns: $u(x,y)$,$v(x,y)$, $p(x,y)$

Weak Form of Governing Equations

$\phi(x, y)$: scalar perturbation (test function) to the solutions on the interior s.t. $\phi|_{\delta\Omega} = 0$.

Multiply both sides of equations by ϕ & integrate over domain:

$$\int_\Omega \Big(\nabla \cdot \mathbf{u}\Big) \phi \, d\Omega = 0$$

$$\int_\Omega \Big(Re(\nabla \cdot [\mathbf{u} \otimes \mathbf{u}]) + \nabla p - \nabla^2 \mathbf{u}\Big) \phi \, d\Omega = 0$$

Applying vector integration by parts, with $\phi = 0$ at boundaries, weak form is:

$$\int_\Omega \Big(\nabla \cdot \mathbf{u}\Big)\phi \, d\Omega = 0$$

$$\int_\Omega \Big(Re(\nabla \cdot [\mathbf{u} \otimes \mathbf{u}]) + \nabla p\Big)\phi - \nabla \mathbf{u} \cdot \nabla \phi \, d\Omega = 0$$

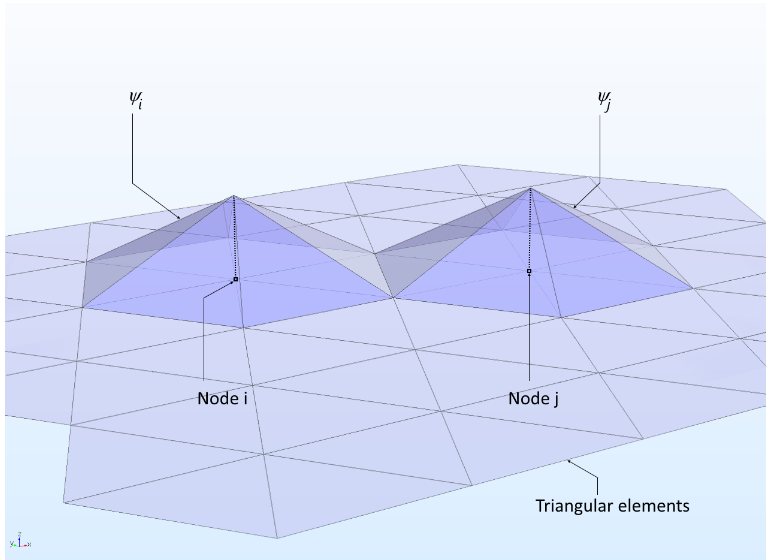
Now assume:

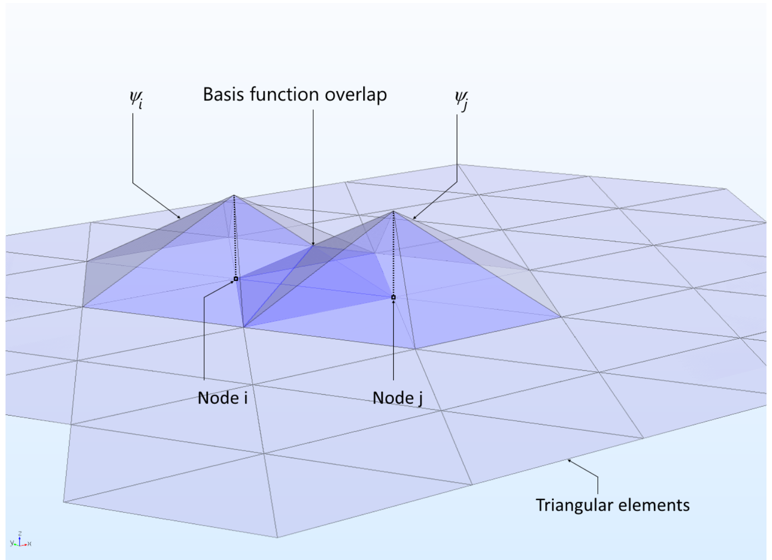
$$u(x,y) = \sum_{i=1}^{N} u_i \psi_i(x,y)$$

$$v(x,y) = \sum_{i=1}^{N} v_i \psi_i(x,y)$$

$$p(x,y) = \sum_{i=1}^{N} p_i \psi_i(x,y)$$

We choose a linear piecewise continuous nodal basis function, s.t. $\psi_i(x,y) = 1$ @ node i and =0 at all other nodes.

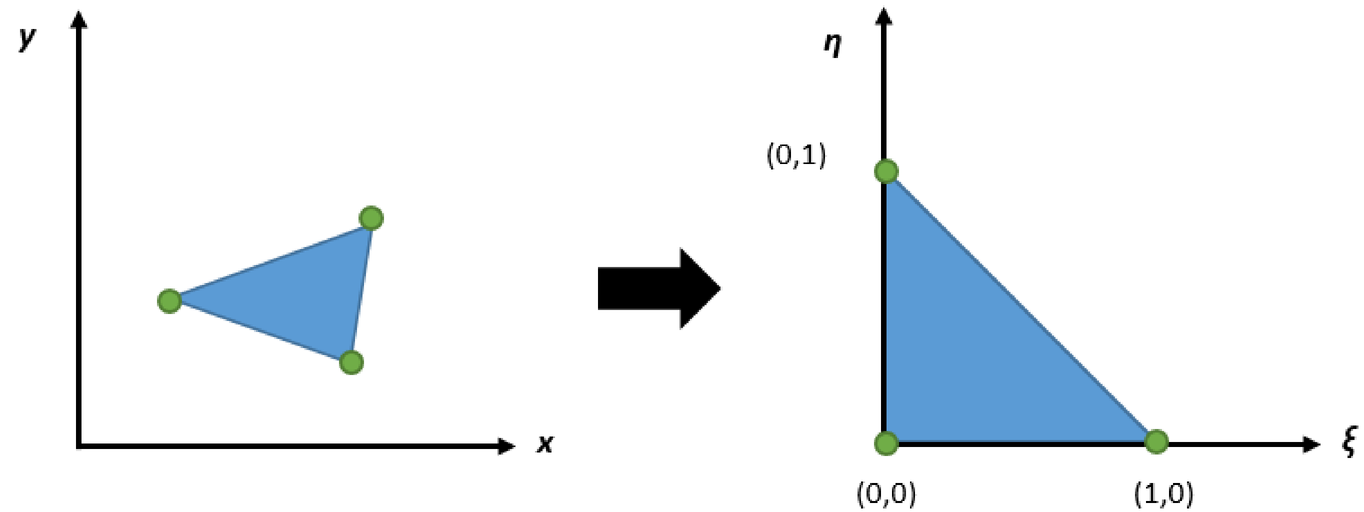




Galerkin Method

- Try each of the N basis functions as test functions in the weak form
- Gives $3N$ equations for the $3N$ unknown u_i ,v_i ,p_i coefficients.
- Simplifies integrals since each basis function only non-zero over adjacet elements.

Chosen basis functions are linear, so weak form is simple to differentiate and integrate in local coordinates ξ, η over adjacent elements.



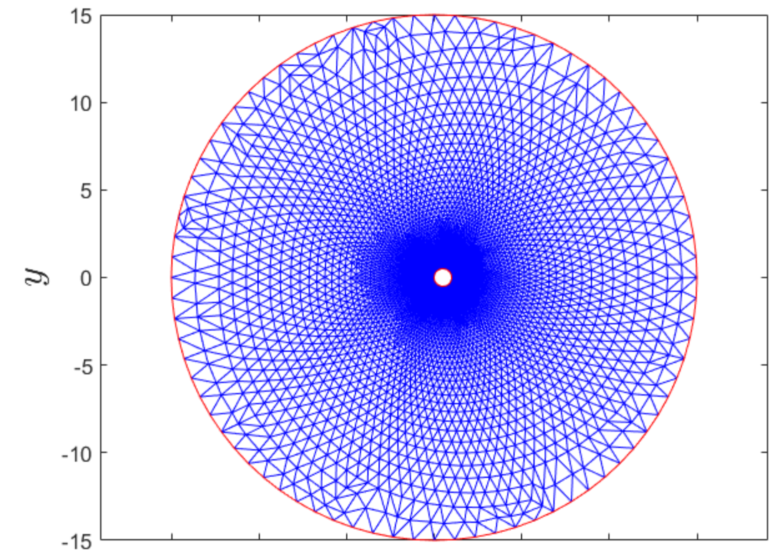
$$\bar{\psi}_1 = 1 - \xi - \eta \quad \bar{\psi}_2 = \xi \ , \bar{\psi}_3 = \eta$$

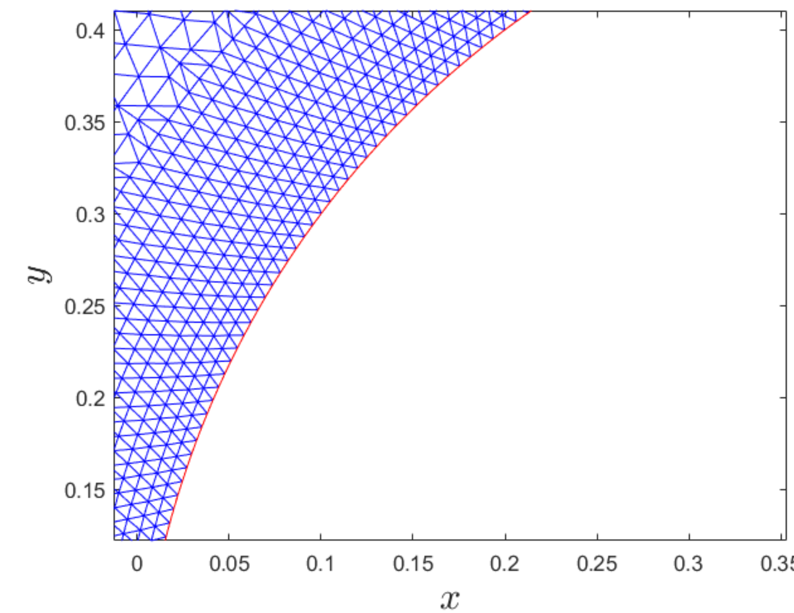
Then transform the integrals and derivatives to global coordinates x, y using Jacobians of elements $J_{elem} = \frac{\partial(\xi,\eta)}{\partial(x,y)}$

Mesh

- Unstructured triangular mesh (26,192 elements & 13,336 nodes)
- Refined near the cylinder (large gradients in BL, spacing= $\frac{D}{200}$ @ wall)
- Coarse away from cylinder to reduce # of unknowns
- Outer domain $15D$ away from from cylinder surface

Mesh imported into MATLAB using node coordinates and connectivity information.





Boundary Conditions

On the wall (cylinder):

- Dirichlet BC on velocity: $u_{wall} = 0, v_{wall} = 0$
- Neumann BC on pressure: $\frac{\partial p}{\partial n}\Big|_{wall} = 0$

On left half outer domain:

Freestream Dirichlet BC on velocity:

- $u_L = V_\infty, v_L = 0$
- Neumann BC on pressure: $\frac{\partial p}{\partial n}|_L = 0$

On right half outer domain:

- Dirichlet BC on pressure (zero guage pressure): $p_R = 0$
- Neumann BC on velocities: $\frac{\partial u}{\partial n}|_R = 0$, $\frac{\partial v}{\partial n}|_R = 0$

Outer domain BCs not immediately obvious, only physically approximate, but domain is far from cylinder so influence is minimal.

FE treatment of BCs

- Neumann BCs are 'Natural Boundary Conditions' (zero gradient BCs automatically satisfied)
    - *Result of dropping term in integration by parts*

- Dirichlet BCs are 'Essential Boundary Conditions': they must be specified explicitly
    - *Equation replaces the respective governing equation at the boundary node*

Solution Method

$3N$ equations to solve, of the form:

$\mathbf{R}_{m_i}(\mathbf{u}_j, \mathbf{v}_j, \mathbf{p}_j) = 0$ (Conservation of mass + p BCs)

$\mathbf{R}_{x_i}(\mathbf{u}_j, \mathbf{v}_j, \mathbf{p}_j) = 0$ (Conservation of x–momentum + u BCs)

$\mathbf{R}_{y_i}(\mathbf{u}_j, \mathbf{v}_j, \mathbf{p}_j) = 0$ (Conservation of y–momentum + v BCs)

Compactly : $\mathbf{R}(\mathbf{x}) = 0$ with $\mathbf{x} \equiv \{\mathbf{u}_j, \mathbf{v}_j, \mathbf{p}_j\}$

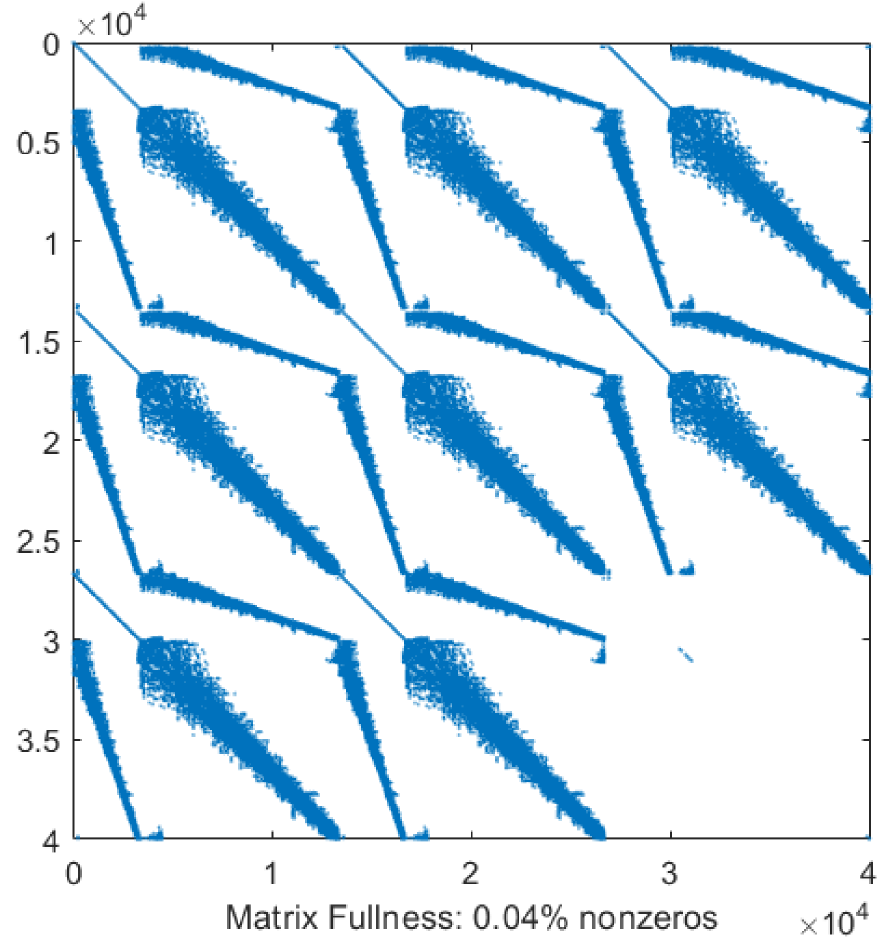
Non-linear in the u_j, v_j, p_j coefficients, and so must be solved using an iterative method.

Newton method chosen for fast convergence, but requires additional effort to code the Jacobian matrix.

$$J = \frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \left[\begin{array}{c|c|c} \dfrac{\partial R_x}{\partial u} & \dfrac{\partial R_x}{\partial v} & \dfrac{\partial R_x}{\partial p} \\ \hline \dfrac{\partial R_y}{\partial u} & \dfrac{\partial R_y}{\partial v} & \dfrac{\partial R_y}{\partial p} \\ \hline \dfrac{\partial R_m}{\partial u} & \dfrac{\partial R_y}{\partial v} & \dfrac{\partial R_y}{\partial p} \end{array}\right]$$

Where each block is of size NxN.

Each residual only depends on adjacent elements, so the Jacobian is sparse:



Then, at each iteration k: $\mathbf{x}^{k+1} = \mathbf{x}^{k} - \mathbf{J}^{-1}(\mathbf{x}^{k})\mathbf{R}(\mathbf{x}^{k})$.
Iteration stops when:

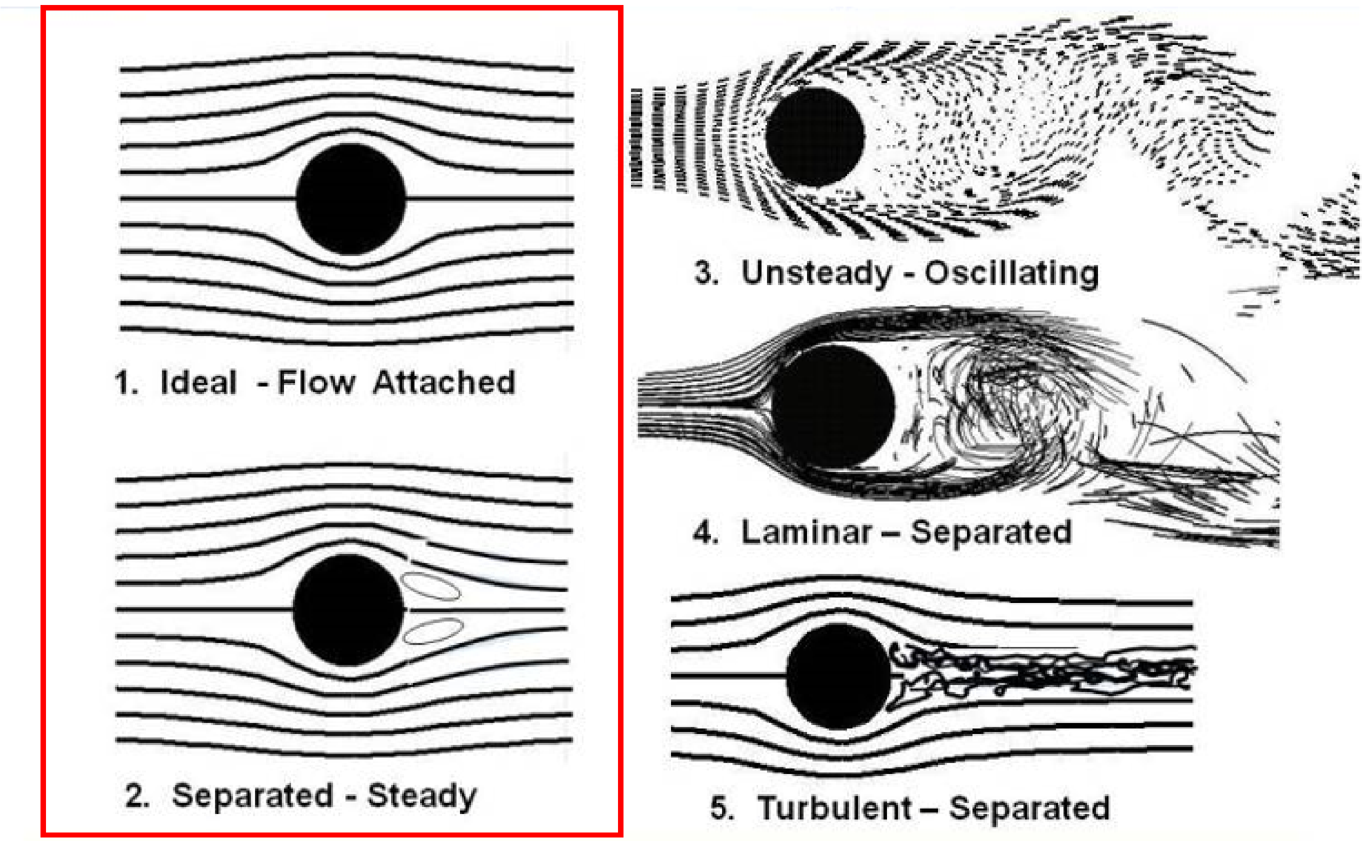
$||\Delta\mathbf{x}|| < tol$ or $k = k_{max}$

In MATLAB, more efficient to store the Jacobian Matrix values, row indicies, column indicies in lists, and then form sparse matrix from lists.

On average each iteration took ~15 seconds.

Solutions converged to machine precision with <10 iterations (quicker convergence at low Re due to smaller gradients)

Regimes of flow over a cylinder



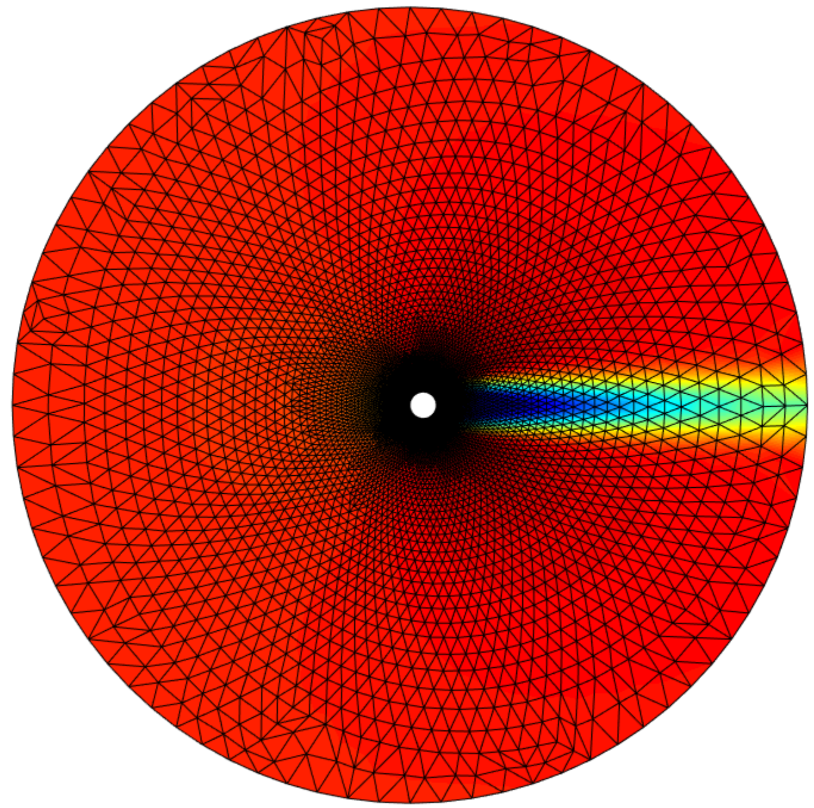
Creeping flow (Re<6.5):

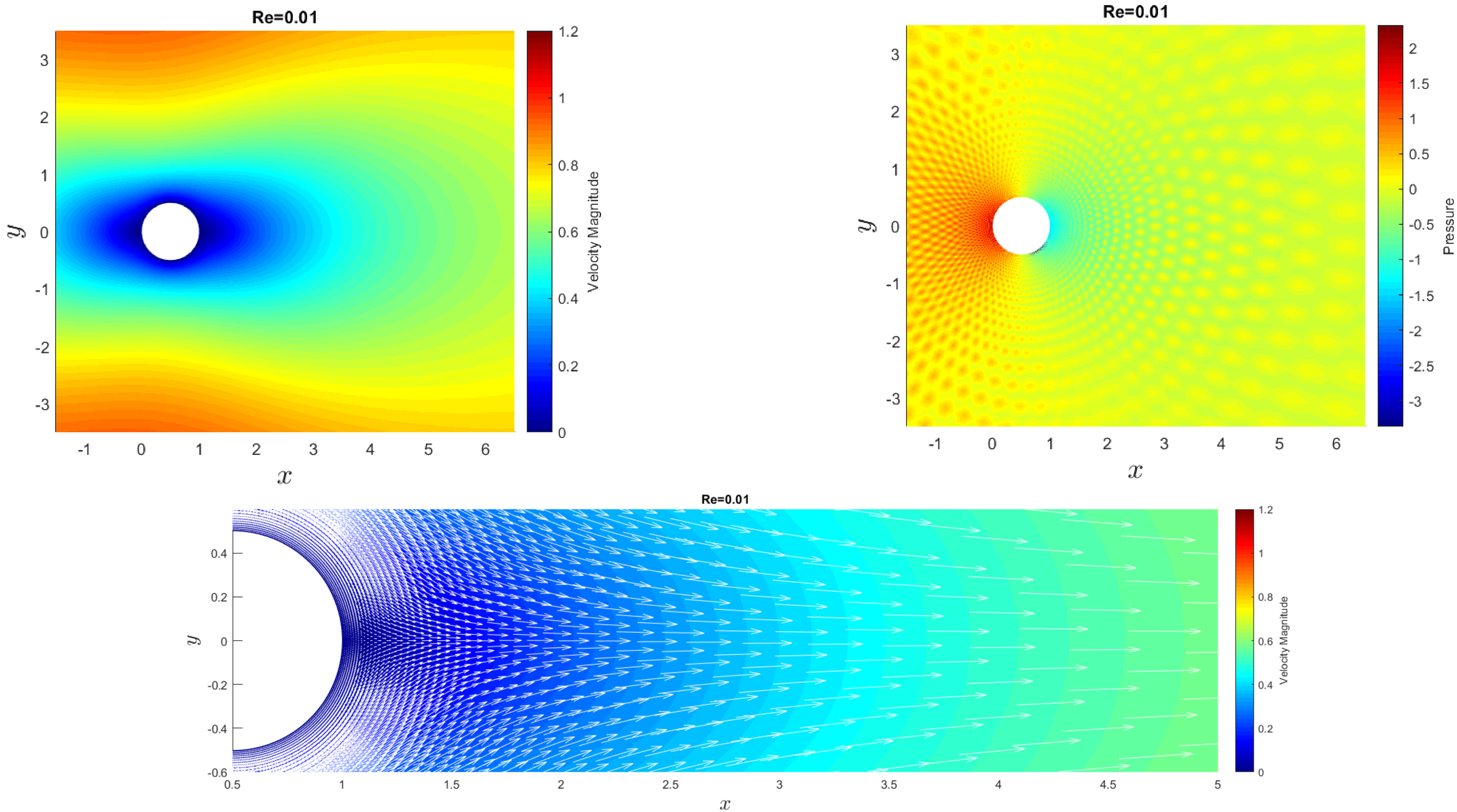
Advective intertial forces small compared to viscous forces (no seperation/recirculaton region).

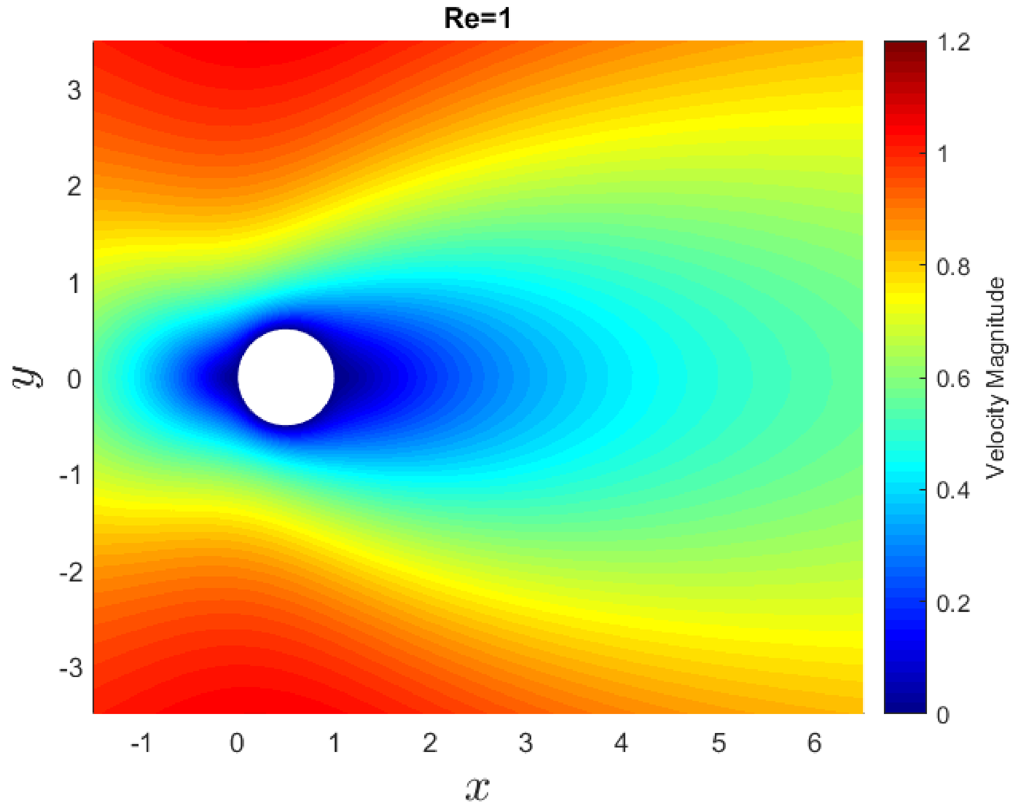
Laminar Separation Region (Re <~100)

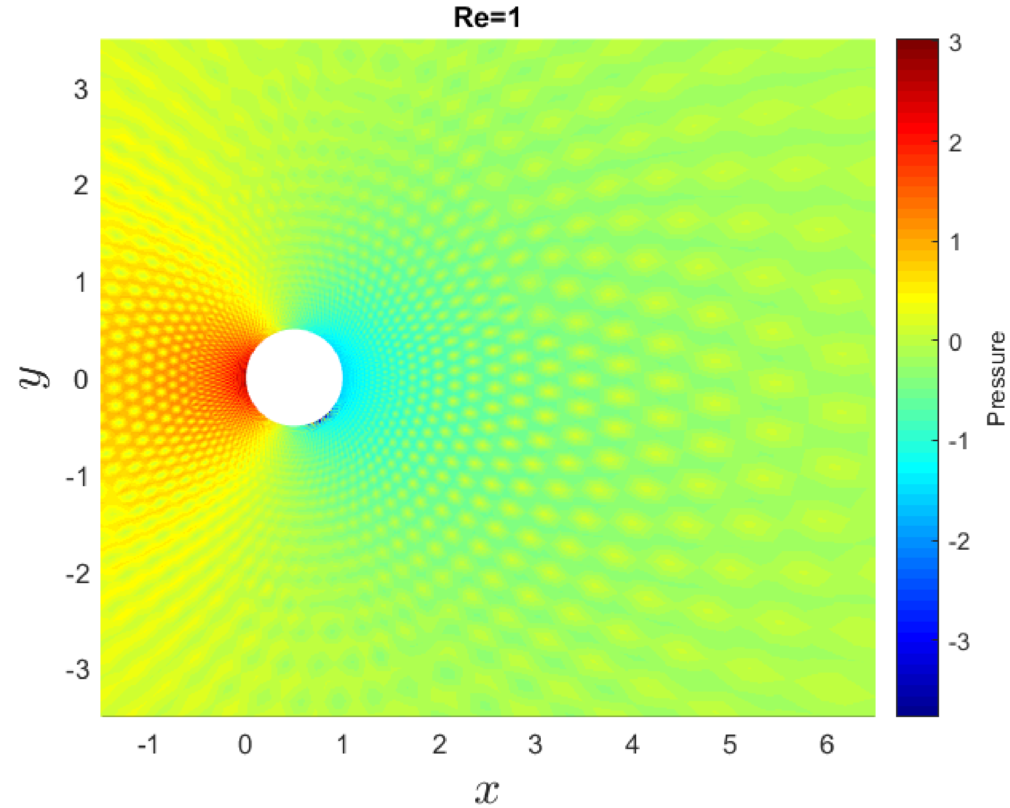
Advective intertial forces dominate, separation is evident downsteam of the cylinder.

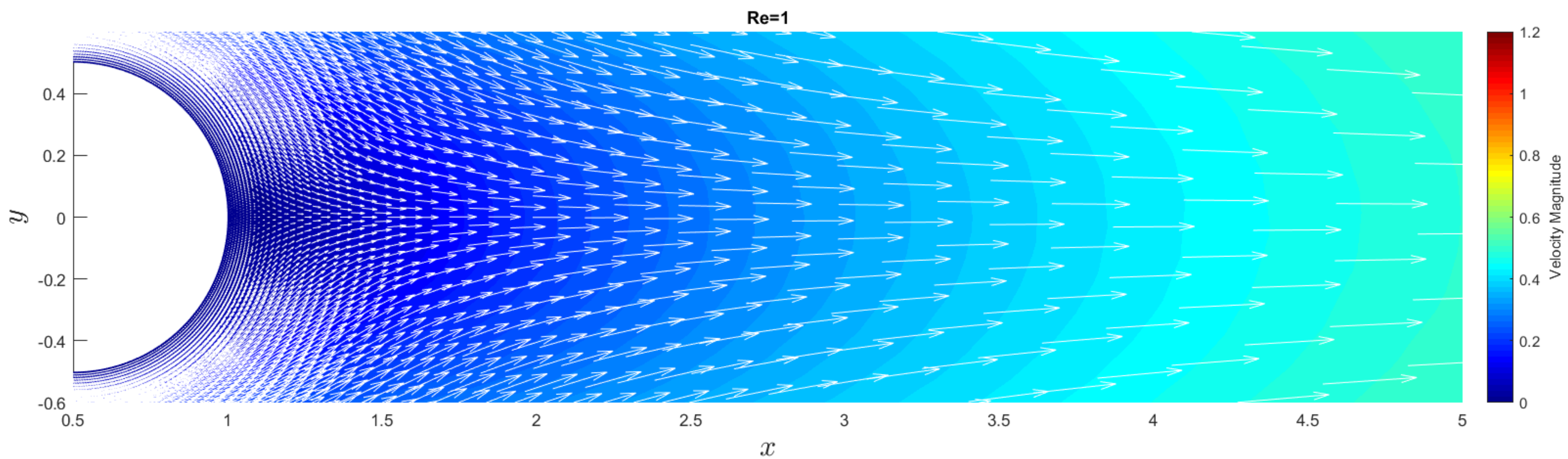
Results

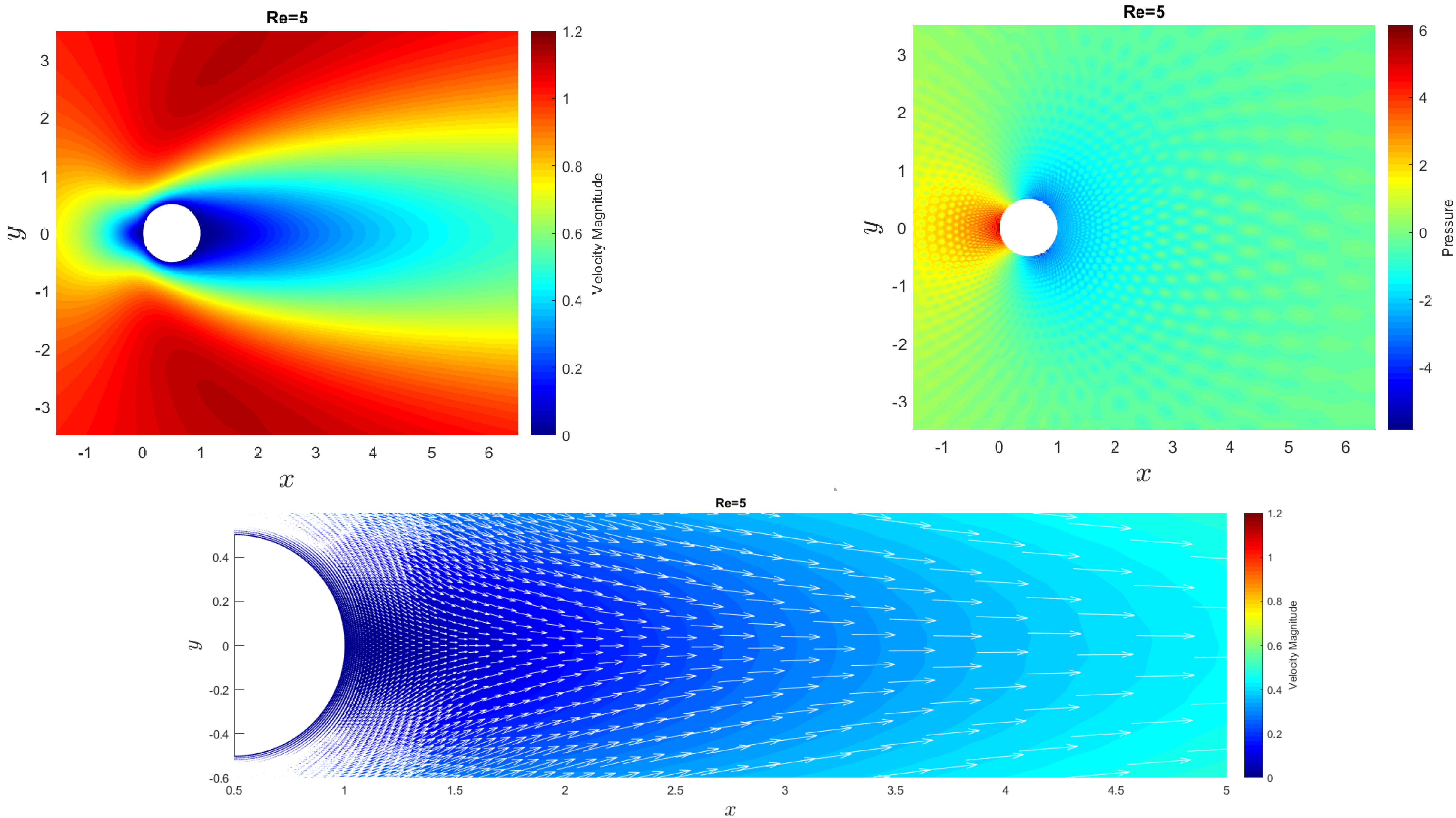


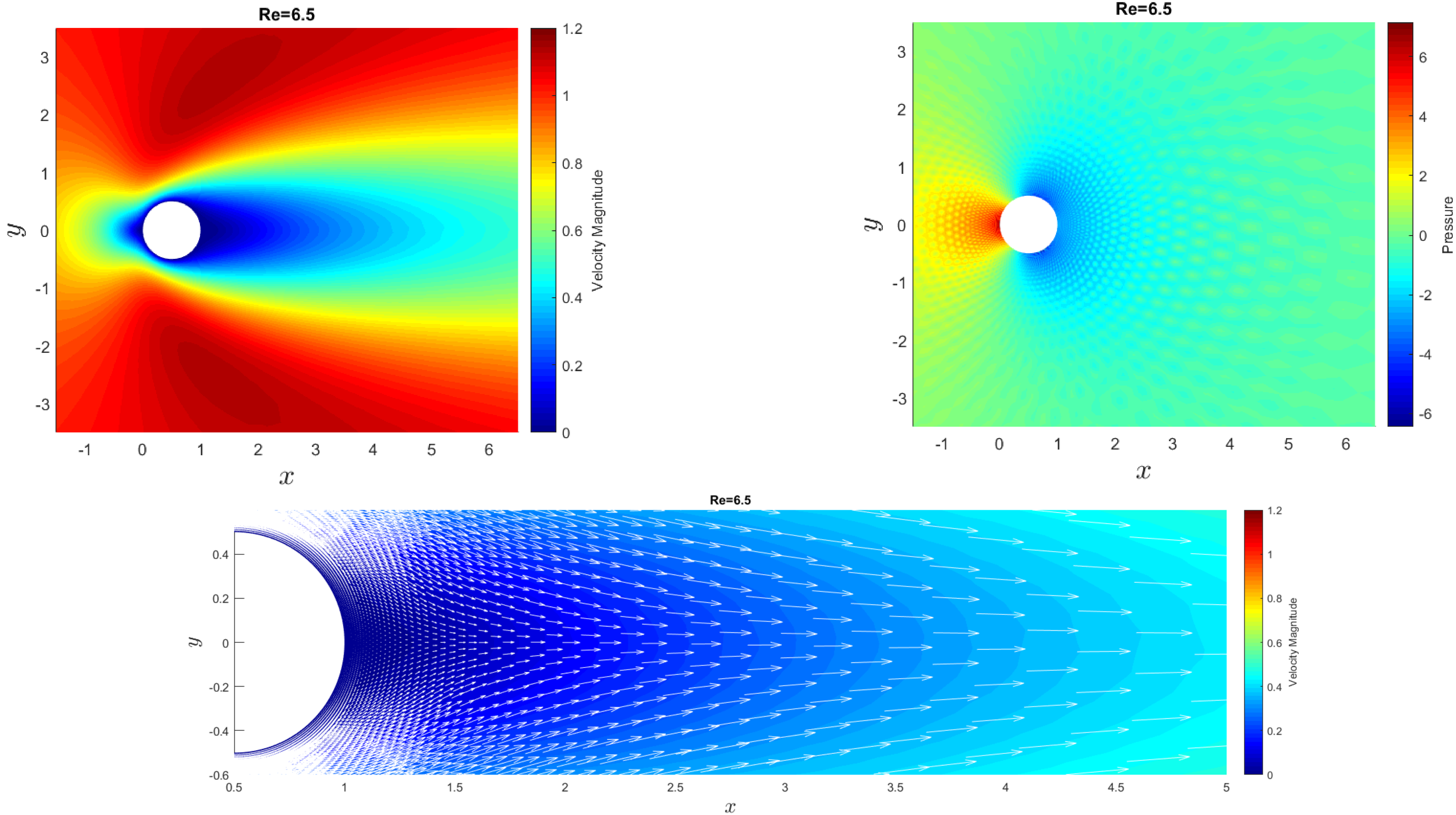

Re=0.01
Velocity Magnitude
Re=0.01
Pressure
Re=0.01
Velocity Magnitude
x
y

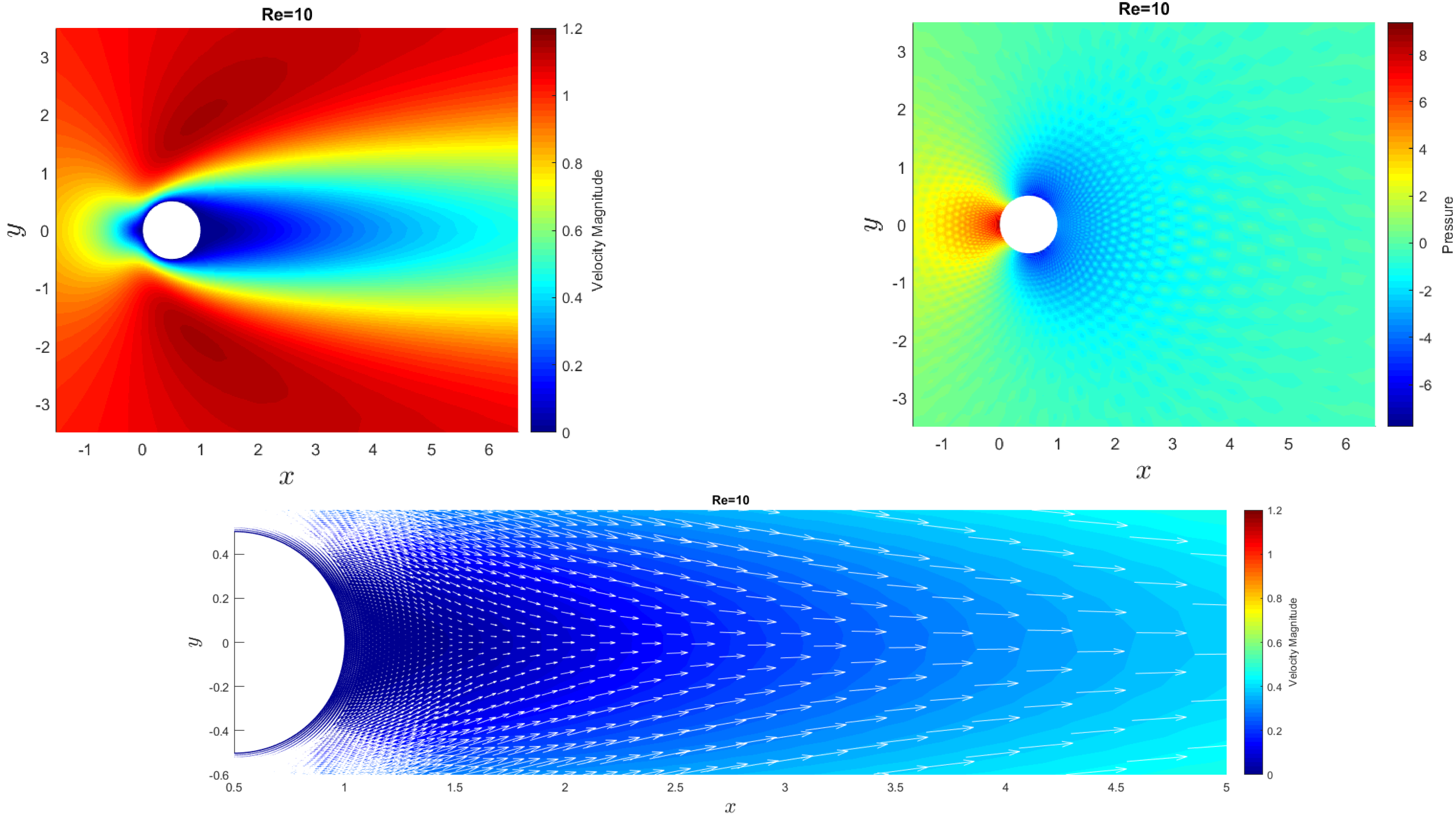

Re=1
Velocity Magnitude
y
x
3
2
1
0
-1
-2
-3
-1
0
1
2
3
4
5
6
1.2
1
0.8
0.6
0.4
0.2
0


Re=1
Pressure
y
x
3
2
1
0
-1
-2
-3
-1
0
1
2
3
4
5
6

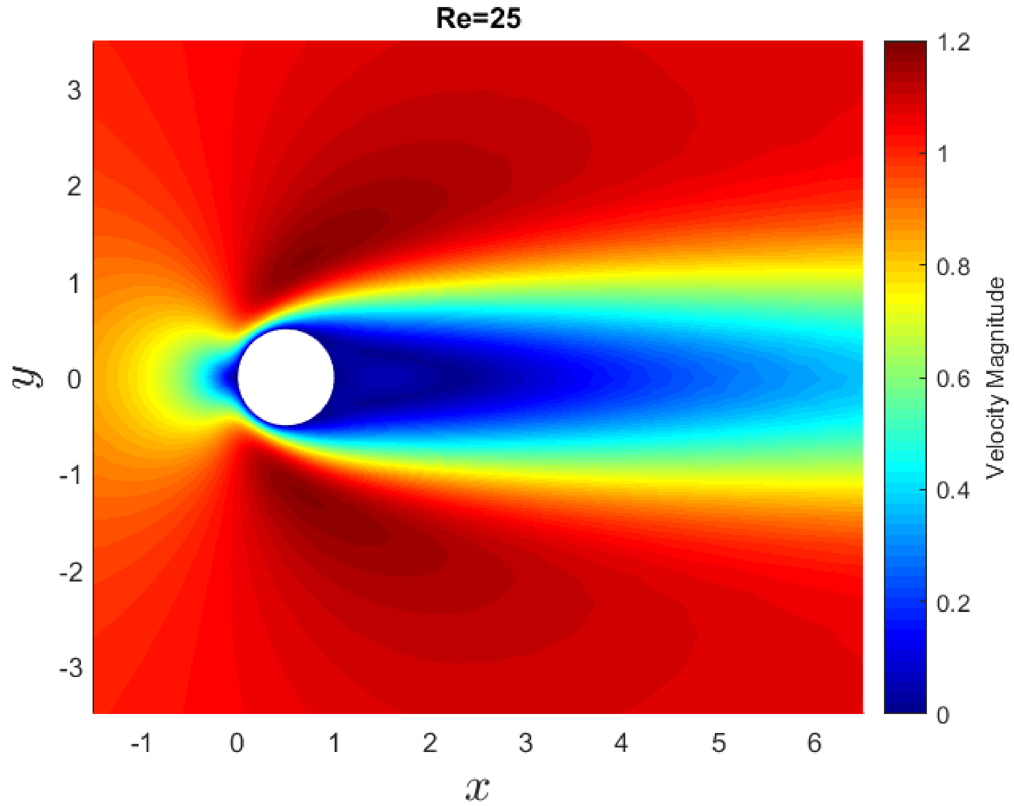

Re=1
Velocity Magnitude
y
x
0.4
0.2
0
-0.2
-0.4
-0.6
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
1.2
1
0.8
0.6
0.4
0.2
0

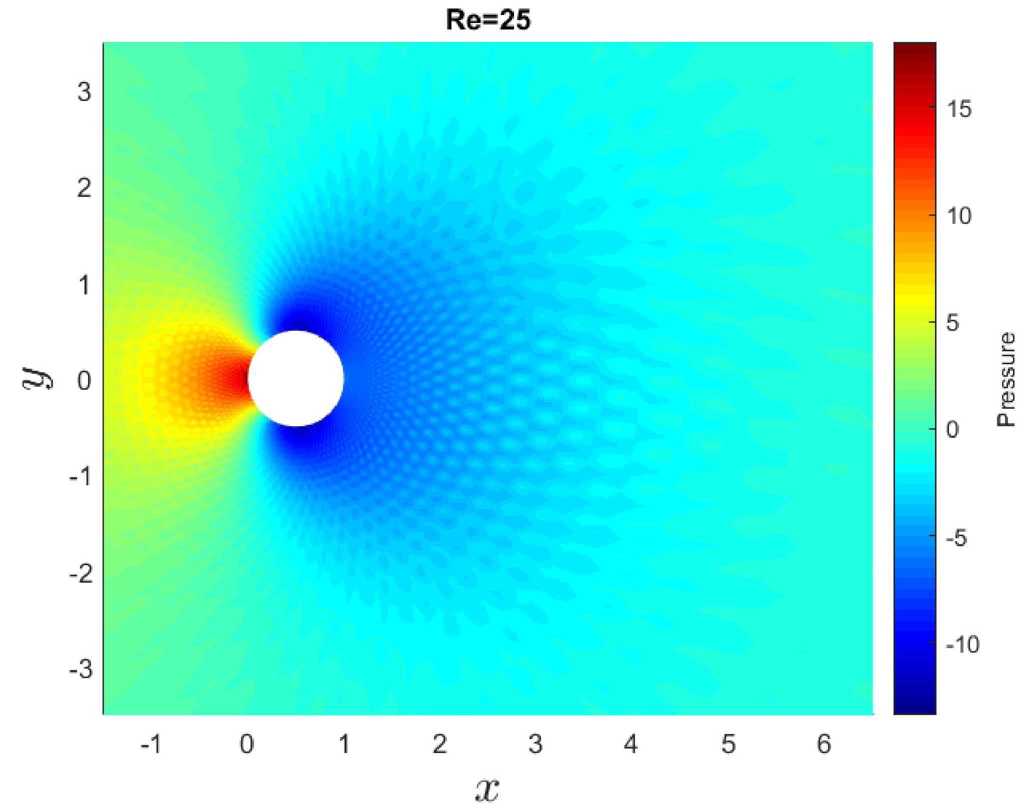

Re=5
Velocity Magnitude
Re=5
Pressure
Re=5
Velocity Magnitude
x
y

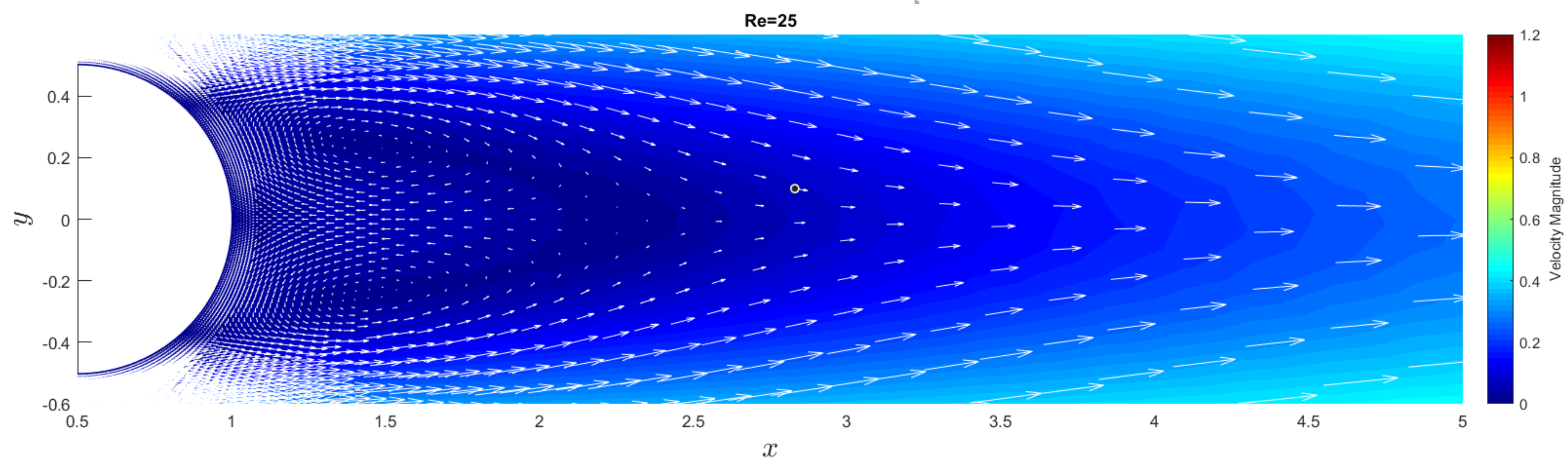

Re=6.5
Velocity Magnitude
Re=6.5
Pressure
Re=6.5
Velocity Magnitude
x
y

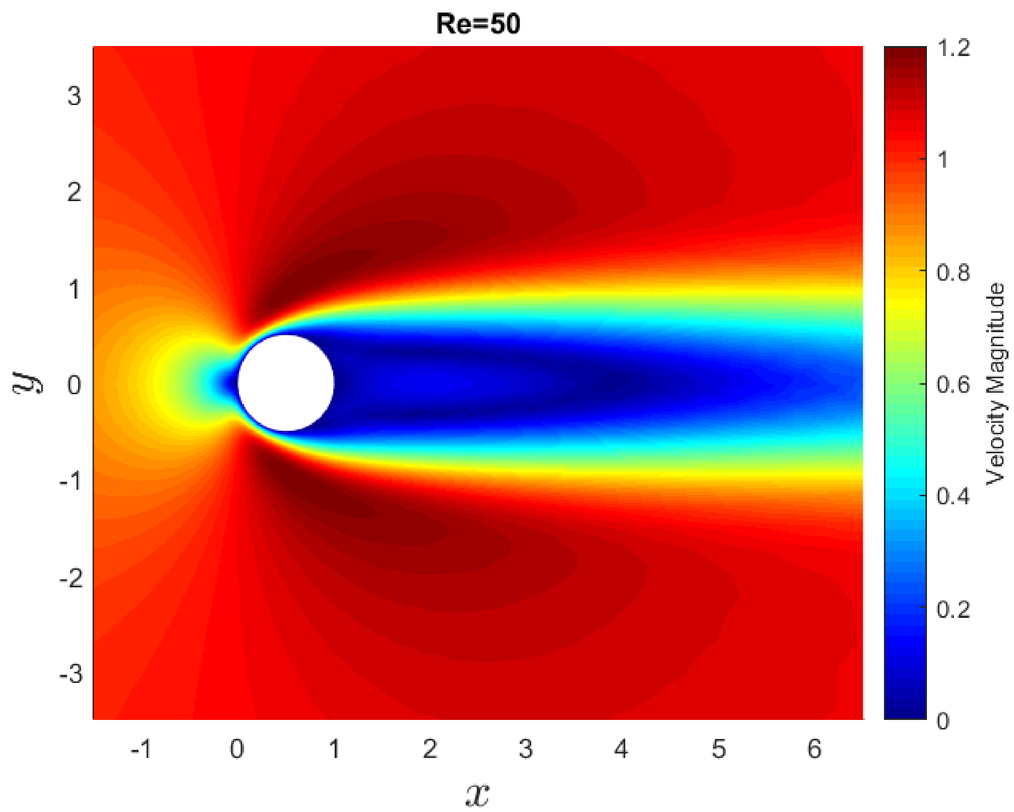


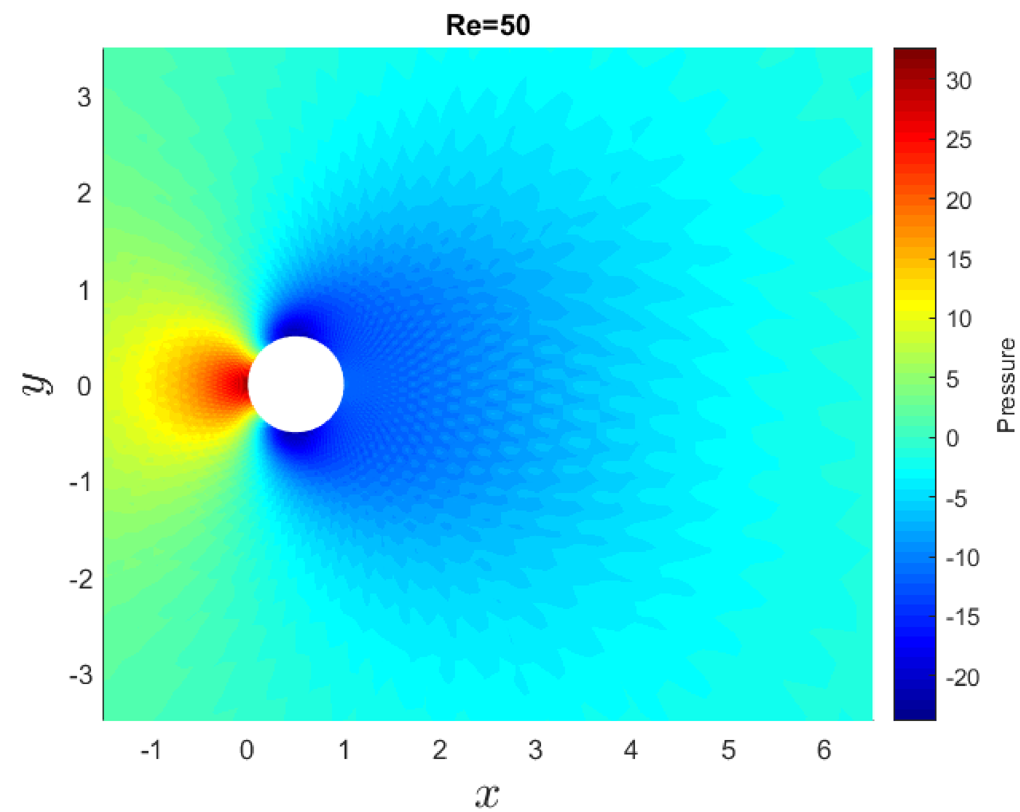
Re=10
Velocity Magnitude
y
x
Re=10
Pressure
y
x
Re=10
Velocity Magnitude
y
x

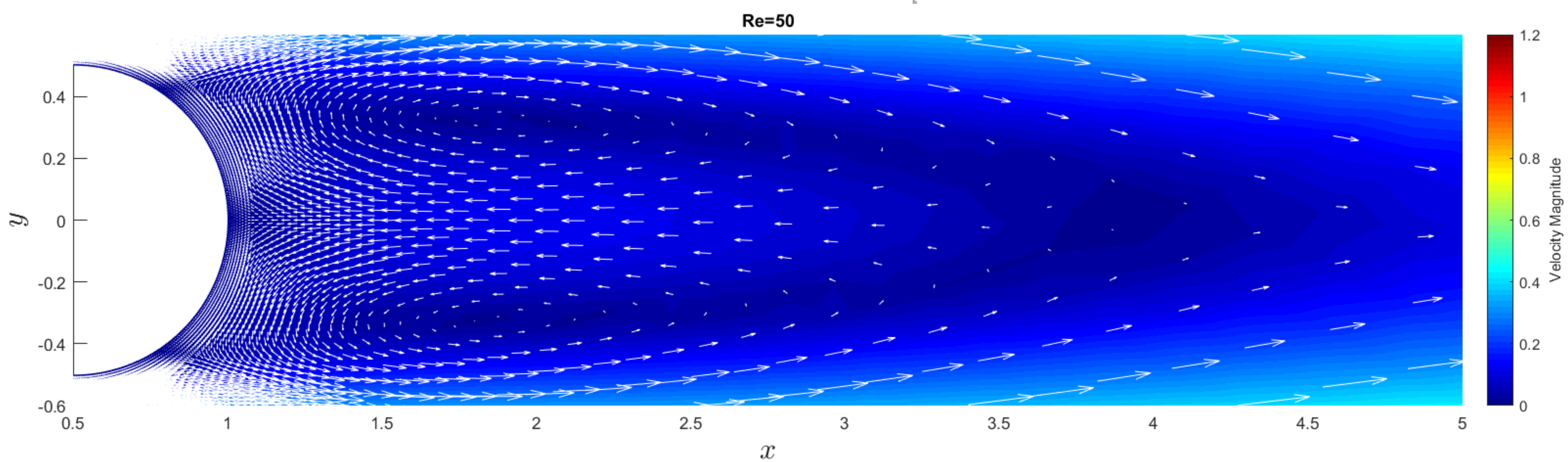

Re=25
Velocity Magnitude
x
y

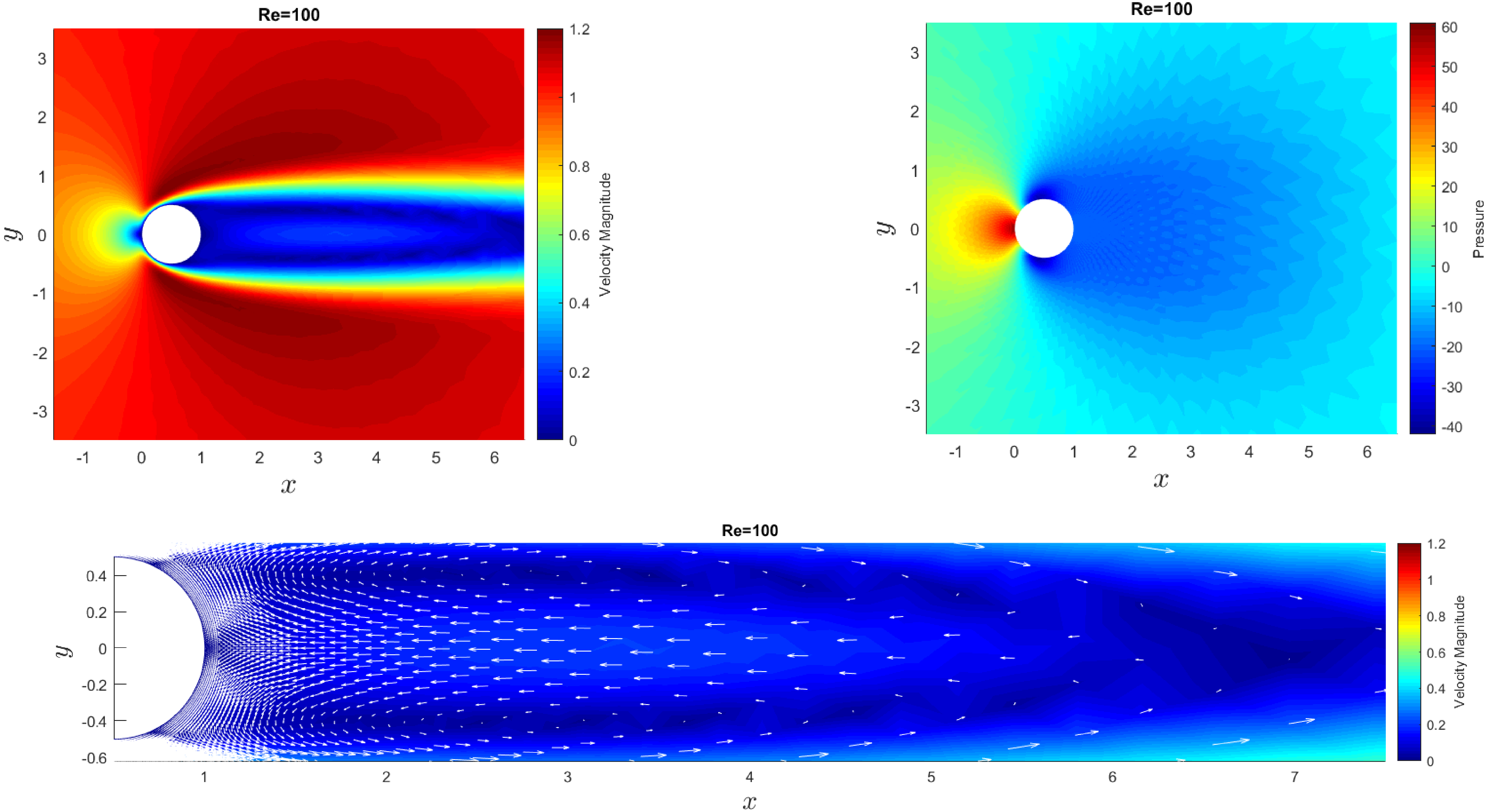

Re=25
Pressure
x
y


Re=25
Velocity Magnitude
x
y

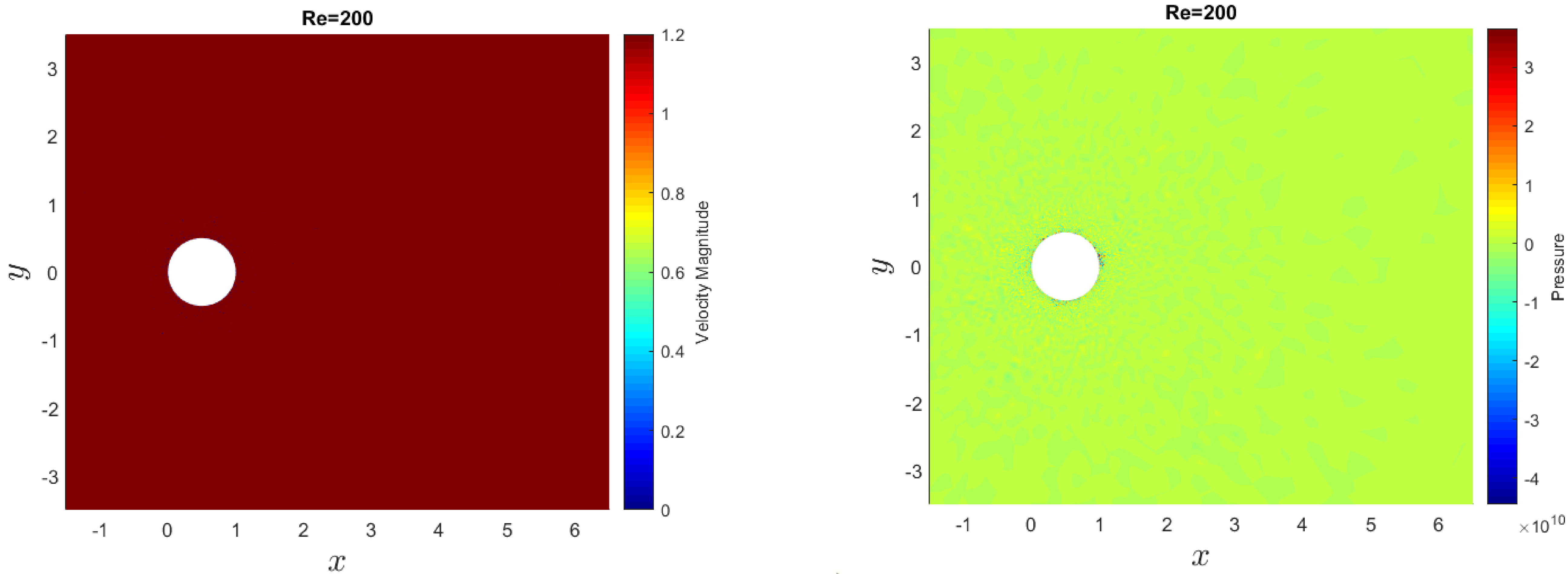

Re=50
Velocity Magnitude
x
y


Re=50
Pressure
x
y


Re=50
Velocity Magnitude
x
y



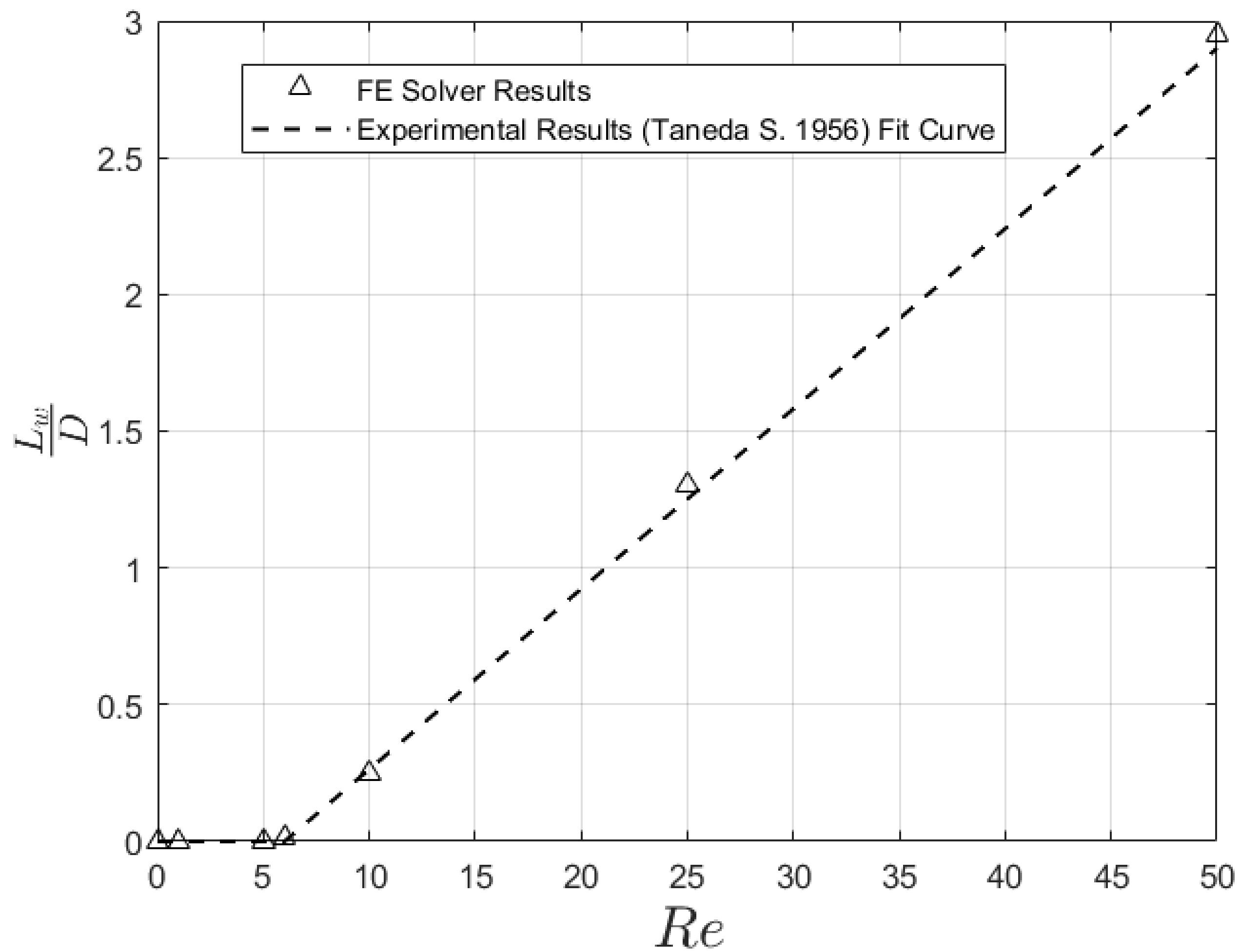
Re=100
Velocity Magnitude
x
y
Re=100
Pressure
x
y
Re=100
Velocity Magnitude
x
y



As expected, catastrophic failure of the solver!

(does not converge at all, residual explodes)

Comparison of recirculation zone length with experimental data



Pressure Oscillations

Use of continuous linear elements for both velocity and pressure basis functions leads to pressure oscillations (proved by Logg and Mardal). This velocity profile is still well-captured because:

- Average gradients are still captured accurately because of fine mesh
- At low Re, pressure force dominated by viscous forces

Solution: Quadratic elements in velocity, linear elements in pressure (but increases complexity).

Future Improvements

- Quadaratic velocity basis functions to eliminate pressur oscillations
- Implement unsteadiness by adding the time dervative term
- Add turbulence model ($k - \epsilon$, SA) by adding Reynolds stress term in RANS equations
- Validation of C_D values with experimental data

These changes will allow going to higher Reynolds numbers and make the code more robust.