Incompressible Inviscid Flow using FreeFEM++

Gaurav Gupta

June 14, 2024

1 Governing Equations

Incompressible inviscid flow is simulated by solving the govering equations in FreeFEM++ using artifical compressibility method. For 2D flow, the mesh is created in FreeFEM++, whereas in 3D flow, the mesh is created with GMSH and imported in FreeFEM++.

Continuity Equation

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

Momentum Equation

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\nabla P$$

Note: The density (ρ) is taken as one for simplicity.

Boundary Conditions

Boundary	**Type**
Inlet	Dirchlet ($\vec{u} = [1\ 0\ 0]$)
Outlet	$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$
Domain Walls	Free-slip
Cylinder Wall	Free-slip

2 Artificial Compressibility Method

The artificial compressibility method is used to solve incompressible flow equations with as hyperbolic equations in a time-marching fashion like compressible flow equations. It is achieved by introducing an artificial parameter in the continuity equation in terms of presssure related through a constant β.

Modified Continuity Equation

$$\beta \frac{\partial p}{\partial t} + \nabla \cdot \vec{u} = 0$$

Momentum Equation

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\nabla P$$

The set of equations are solved until a steady solution is obtained. This steady solution is also the solution of the original govering equations as $\frac{\partial p}{\partial t}$ tends to 0 at steady state.

Weak Form

The time derivatives are written using Backward Euler and the weak form is derived for the above modified equations.

$$\int_\Omega \left(\frac{\vec{u}^{n+1} \cdot \vec{v}}{\Delta t} + (\vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1}) \cdot \vec{v} - p\nabla \cdot \vec{v} - q\nabla \cdot \vec{u}^{n+1} - \frac{\beta p^{n+1} q}{\Delta t} \right) d\mathcal{V} - \int_\Omega \left(\frac{\vec{u}^n \cdot \vec{v}}{\Delta t} - \frac{\beta p^n q}{\Delta t} \right) d\mathcal{V} = 0$$

The non-linear term of the equation is computed using the approximation $u^n \cdot \nabla \vec{u}^{n+1}$ in FreeFEM++.

Listing 1: Problem definition in FreeFEM++

```
solve bdryn([normalappx,normalappy],[w1,w2])
=int1d(Th)(w1*normalappx+w2*normalappy)
-int1d(Th)(w1*N.x+w2*N.y)
+int2d(Th)(1.e-8*(w1*normalappx+w2*normalappy));

macro grad(u) [dx(u), dy(u)]//
macro UGrad(un, u) ((un#1)*grad(u#1) + (un#2)*grad(u#2))//
macro con(un,dt) [convect([un#1,un#2],-dt,un#1),convect([un#1,un#2],-dt,un#2)]//
macro div(u) (dx(u#1) + dy(u#2))//

solve GE([u1,u2,p],[v1,v2,q])
=int2d(Th)
([u1,u2]'*[v1,v2]/dt
+un1*(grad(u1)'*[v1,v2])
+un2*(grad(u2)'*[v1,v2])
-div(v)*p
-div(u)*q
-Beta*p*q/dt
)
-int2d(Th)(
[un1,un2]'*[v1,v2]/dt
-Beta*pn*q/dt
)
+int1d(Th,2)(u1*v1/dt + un1*dx(u1)*v1 +u2*v2/dt + un1*v2*dx(u2))
-int1d(Th,2)(un1*v1/dt + un2*v2/dt)
+on(1, u1=1, u2=0)
+int1d(Th,qft=qf1pTlump,5)(1e10*(u1*normalappx+u2*normalappy)*(v1*normalappx+v2*normalappy))
+int1d(Th,qfe=qf1pE,5)(1e10*(u1*N.x+u2*N.y)*(v1*N.x+v2*N.y))
+on(3, u2=0)
+on(4, u2=0);
```

3 Results of 2D Computation

The code is tested using the benchmark case of flow around a cylinder. The solution of the flow is already known from potential flow theory. β is a user-defined parameter whose value is varied along with the number of points on the cylinder boundary. An initial domain of [-10D,10D] in x and [-6D, 6D] in y-direction. The unstructured mesh used for computation is presented in **Figure 1**.

3.1 Number of points on cylinder

Increasing the mesh resolution on the surface of the cylinder by increasing the number of points clearly increases the accuracy of the solution (**Figure 2**). Here, we kept the domain unchanged and $\beta = 10^{-9}$ for all cases. The solution immediately converges for N=100 and N=150 whereas it blows up in case of N=50. For the case of N=50, spurious oscillation is observed at the top and bottom points which is corrected with mesh refinement.

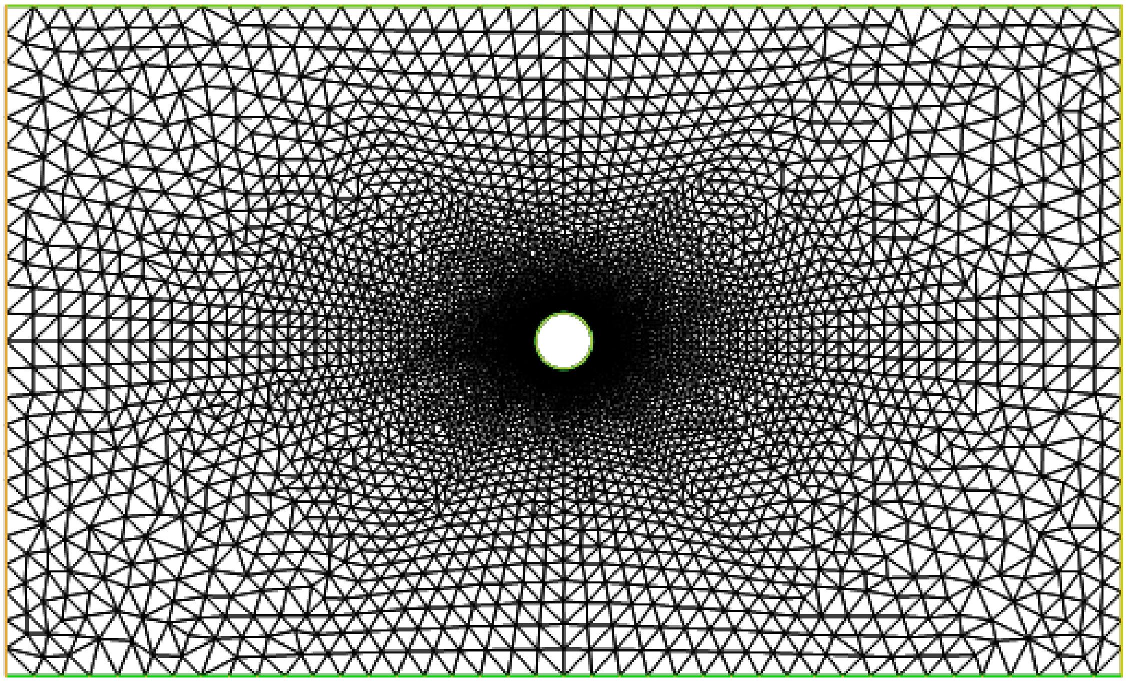

Figure 1: Unstructured mesh contructed in FreeFEM++ with 100 points on the cylinder body

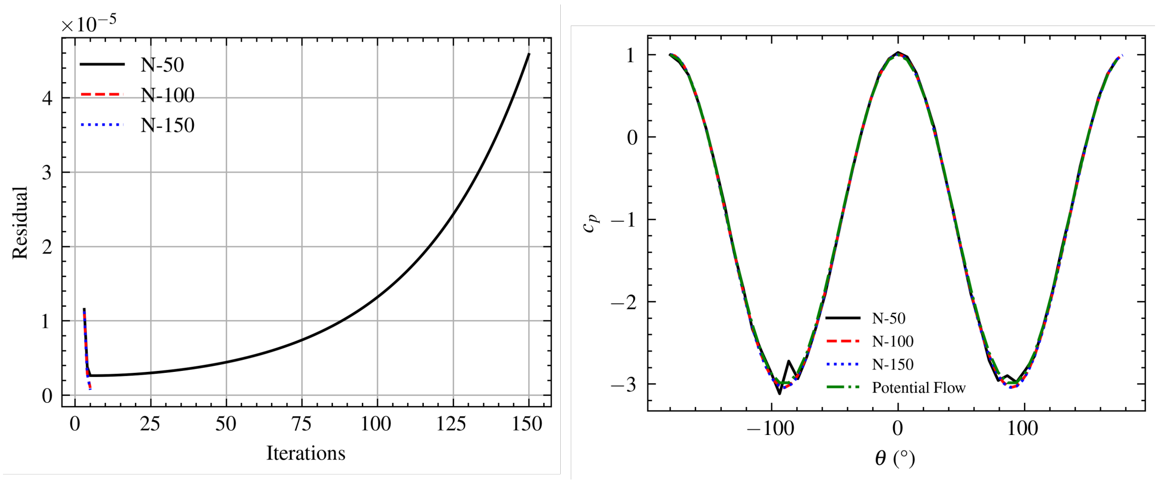

Figure 2: Variation in the residual and pressure distribution over the surface with time for different number of points on the cylinder boundary

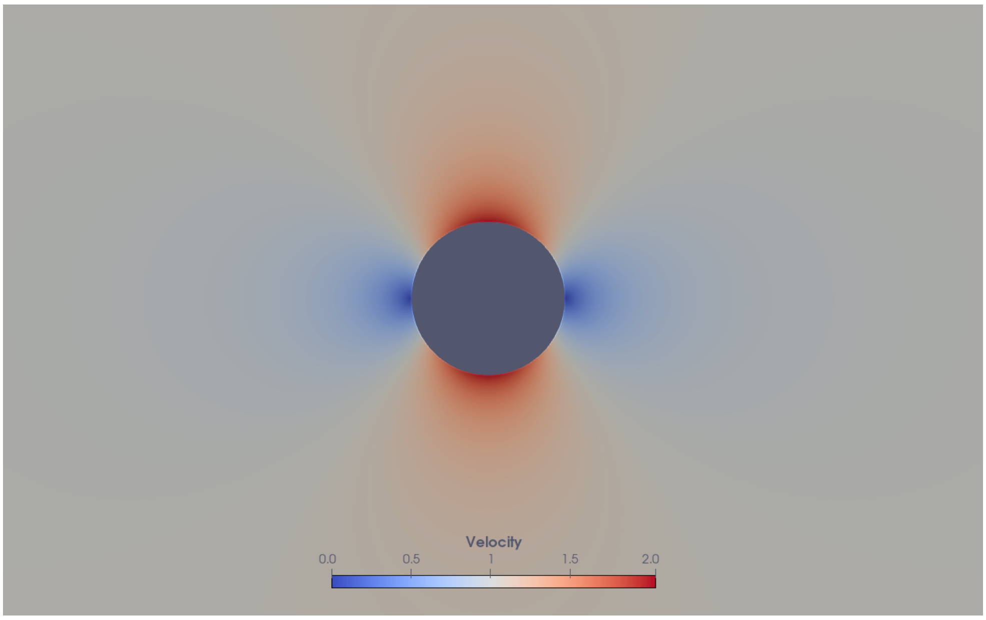

Figure 3: Velocity contour over the cylinder for 50 points on the cylinder surface.

3.2 Variation of β

β is the user-defined parameter to control the artifical compressibility introduced in the equation. In general, the value of β is choosen less than one and a smaller value helps in faster convergence. Here, the domain size is kept unchanged with 100 points on the surface of cylinder. The residual convergence is faster with increase in β. For $\beta = 10^{-5}$ the c_p plot is way off the potential solution, whereas for the case of $\beta = 10^{-7}$ and $\beta = 10^{-9}$, the c_p plot coincides with the potential flow solution. (**Figure 4**). The solution converges for $\beta = 10^{-9}$ in 7 timesteps along with accuracy.

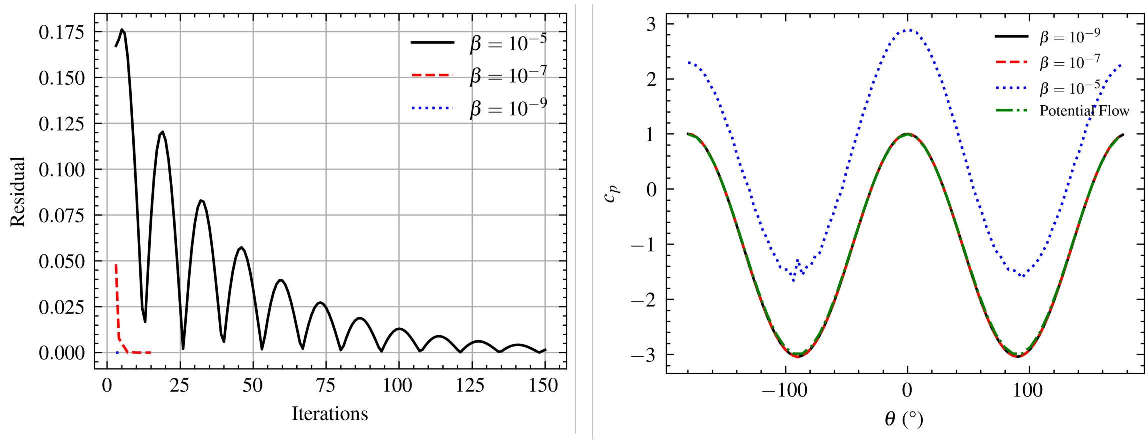

Figure 4: Variation in the residual and pressure distribution over the surface with time for different value of β.

3.3 Variation with Domain

The computational domain is enlarged from the original size of $20D \times 12D$ to $30D \times 16D$ and reduced to $10D \times 8D$. The mesh refinement i.e. number of points on the surface is fixed as 100 whereas the β is fixed as 10^{-9}. The solution rapidly diverges for the smaller domain of $10D \times 8D$ whereas it converges for the other two domains quickly. Spurious oscillations are also observed at the top and bottom points of the cylinder i.e. $\theta = 90$ and $\theta = -90$ for the smaller domain due to interference from the top and bottom walls. (**Figure 5**).

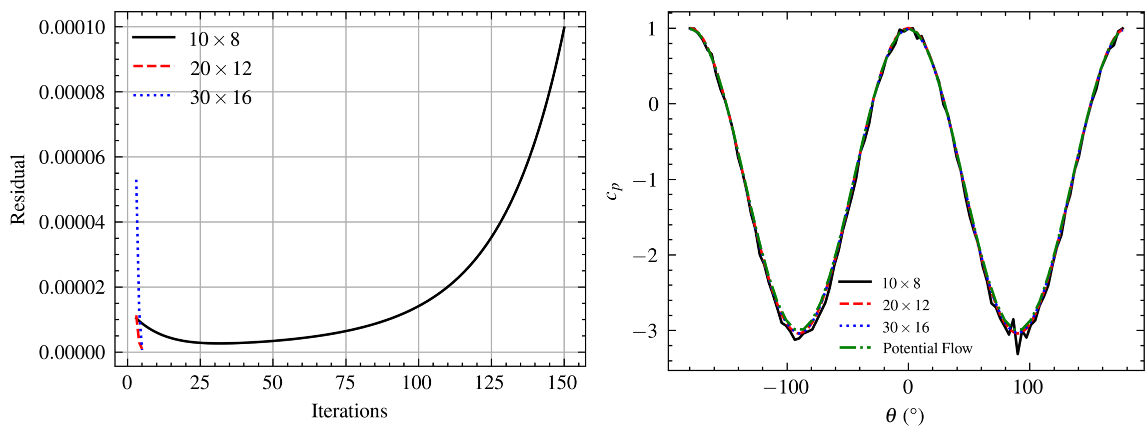

Figure 5: Variation in the residual and pressure distribution over the surface with time for domain.

4 Conclusion

The computation converges quickly to an accurate steady-state solution for a large domain with mesh refinement near the body and a small value of β. The domain should be large enough such that it doesn't interfere with the flow around the simulation and far-field conditions are achieved.