Body Cell Tree Code for Viscous Flow over Cylinder

Naveen Himthani (120010001)

November 2015

1 Aim

Implement the body-cell tree-code for point vortices for flow over a cylinder

1. Consider a distribution of random points in a square. Compare the time taken to compute the velocity for all vortices on each other using the direct $O(N^2)$ method and the tree code. Also check that the accuracy is maintained. Plot the time taken versus the number of particles. Calculate the cross-over-point" when it is faster to compute with FMM than with the direct method.
2. Use the FMM for the velocity evaluation for the point vortices in the flow past circular cylinder and do a higher-resolution computation using it to get better results.

2 Method Used

The tree code approach used by Prof. Prabhu Ramachandran in his PhD thesis **CITE HERE** has been used to compute the velocity of each vortex blob.

3 Results

3.1 Tree Generation

Random points following the normal distribution were generated using a random number generator in python. Then the body cell tree was generated which looks like

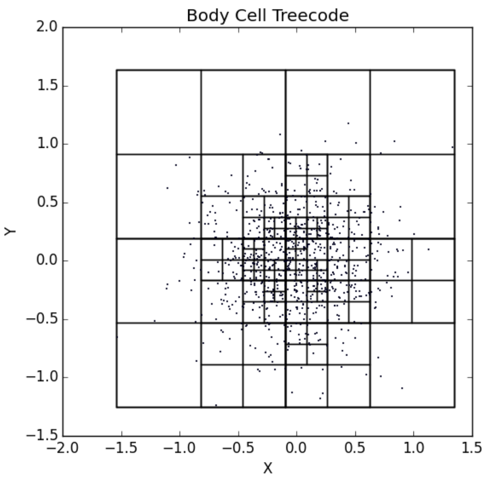

Figure 1: Body cell tree

As we can clearly see that the number of smaller cells are more in the center than towards the outside. This is because we have used a normal distribution to generate the points with $\mu = 0$ and $\sigma = 0.2$ and hence more density in the center and consequently more cells in the center. The creiteria used for dividing the cells is that no childless cell should contain more than 10 particles.

3.2 Comparison with $O(N^2)$ algorithm

The velocity for each particle was computed using the tree code which is supposed to be $O(NlogN)$ and also with the direct $O(N^2)$ algorithm. The time taken by the two was compared for different number of particles and then plotted on a log-log scale plot.

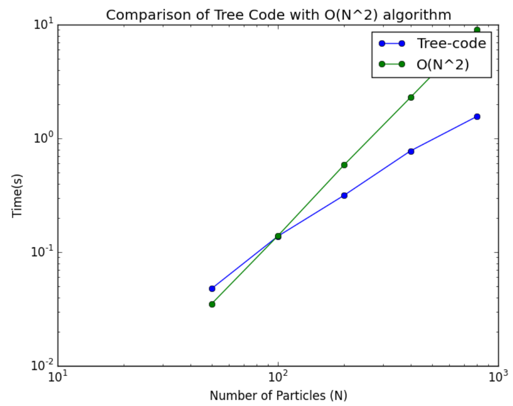

Figure 2: Computational time comparison

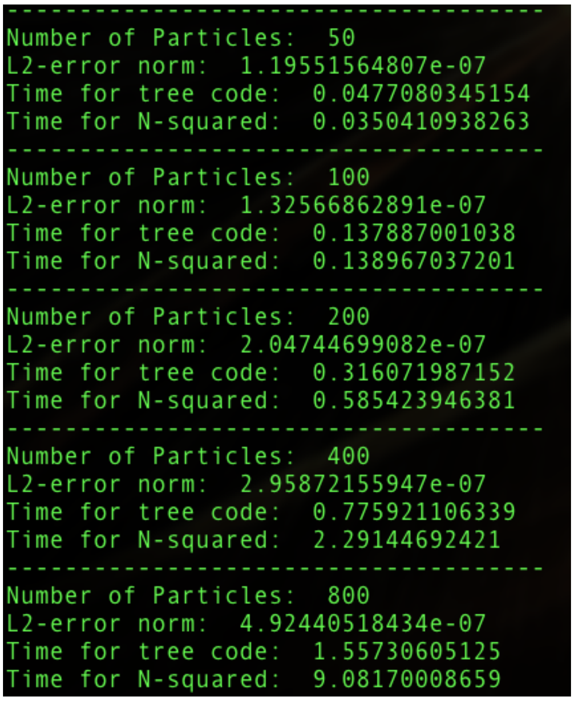

Figure 3: Error for different Number of particles

In Fig 2, the $O(N^2)$ algorithm is faster to compute for less than approximately 100 particles, but for more than 100 particles, the time taken by tree code algorithm is less. This is the crossover point. The parameter used in the tree code were:

- $N_{p_{max}} = 5$
- Number of terms in the multipole expansion = 30 for comparable accuracy. L2 error norm is of the order of 1e-7 as is evident from Fig 3

3.3 Tree code for flow over cylinder for 3 seconds - Particle Plot

Using the Tree Code, the flow over a cylinder was simulated with the following parameters:

- $N_{p_{max}} = 20$
- Order of multipole expansion = 15 for good enough accuracy
- $\gamma_{max} = 0.05$ for good resolution
- $N_{panels} = 60$ for good resolution

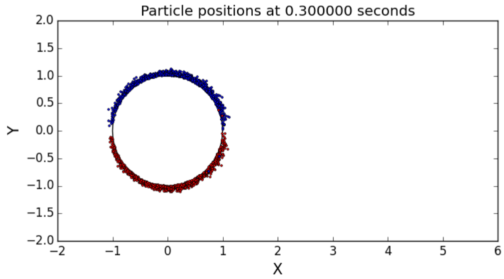

Figure 4: 0.3 sec

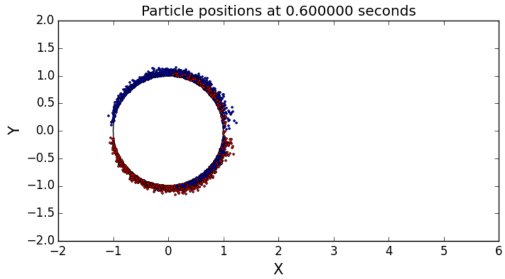

Figure 5: 0.6 sec

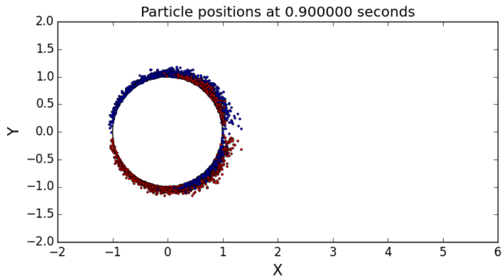

Figure 6: 0.9 sec

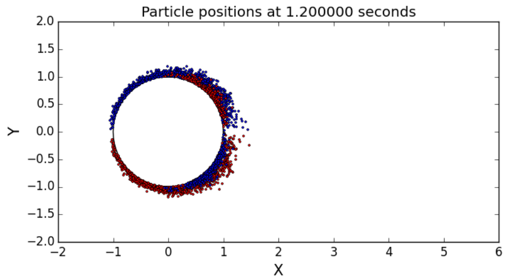

Figure 7: 1.2 sec

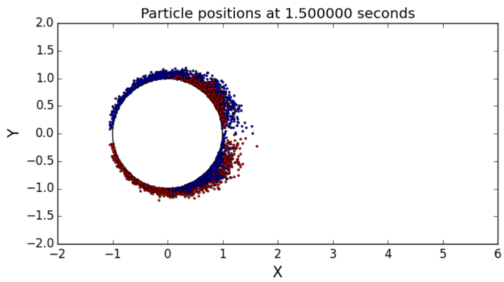

Figure 8: 1.5 sec

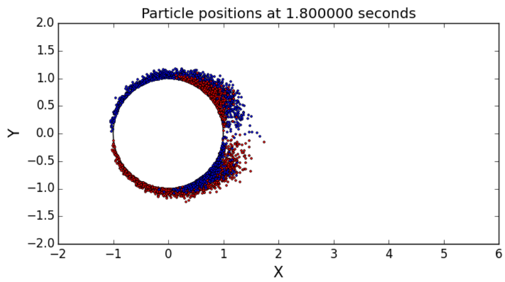

Figure 9: 1.8 sec

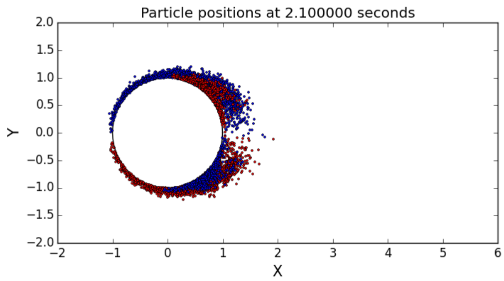

Figure 10: 2.1 sec

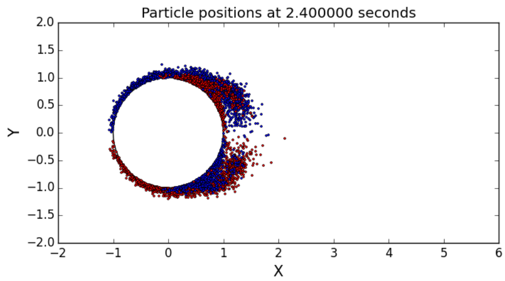

Figure 11: 2.4 sec

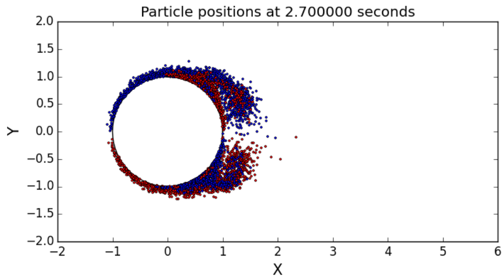

Figure 12: 2.7 sec

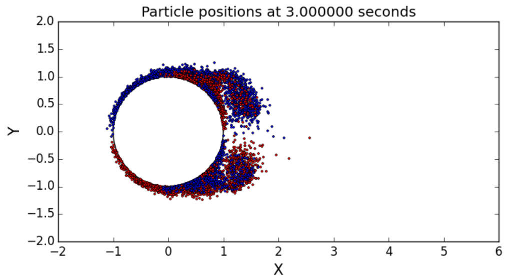

Figure 13: 3.0 sec

As is evident from Fig 13, the flow is of quite high resolution. The time taken for this simulation was approximately 15 min with over 5200 particles being created at the end of 3 seconds.

3.4 Velocity Quiver Plot at 3 seconds

Below is a velocity quiver plot for the flow at 3 seconds. The formation of vortices behind the cylinder is clearly visible from the figure.

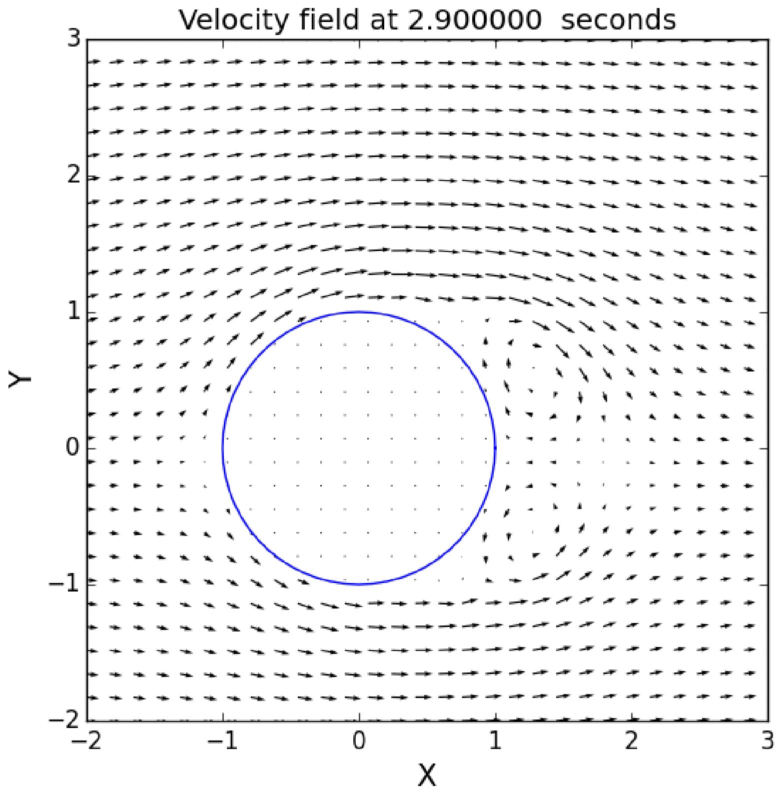

Figure 14: Velocity Quiver plot

3.5 Drag Computation

The drag coefficient is computed in the same way it was done assignment 4. The profile is very much similar.

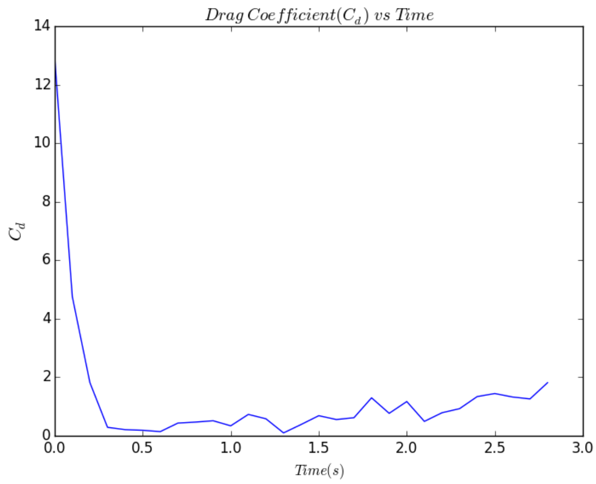

Figure 15: Drag Coefficient with time

4 Observations and Conclusions

The tree code algorithm is a quite fascinating algorithm which gives a lot of performance improvement for a very small trade-off for accuracy. The simulation run-time is greatly reduced from over 3 hrs to less than 15 min, where the 3 hr version used parameters which resulted in loss of accuracy. The main aim of the algorithm has been achieved i.e. to give greatly improved performance so that our simulation can be run with greater resolution for greater accuracy as is required by the Random Vortex Method.