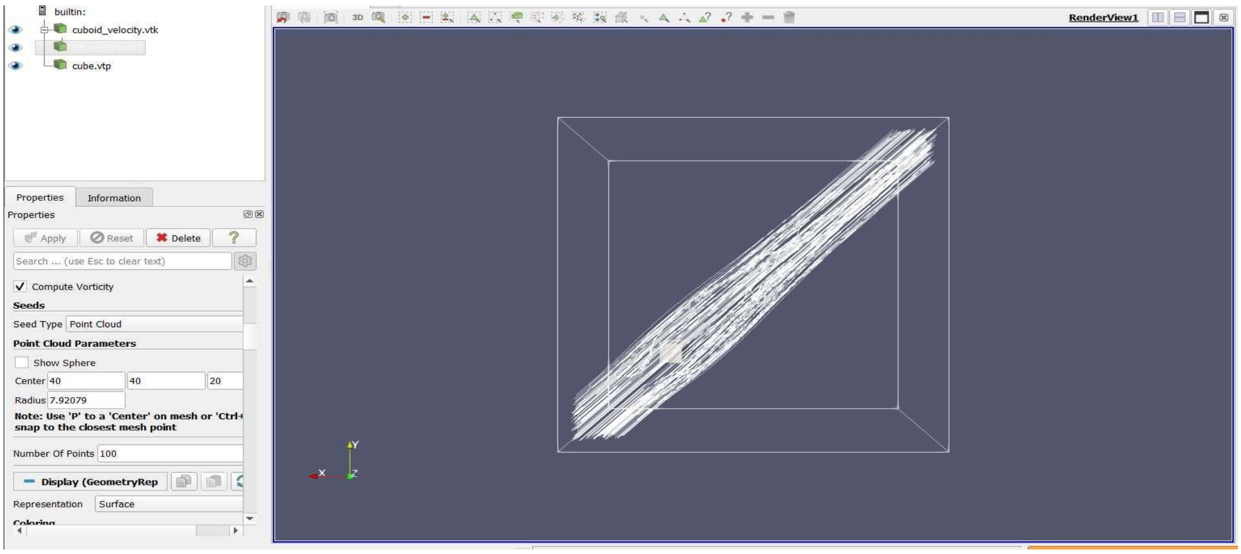
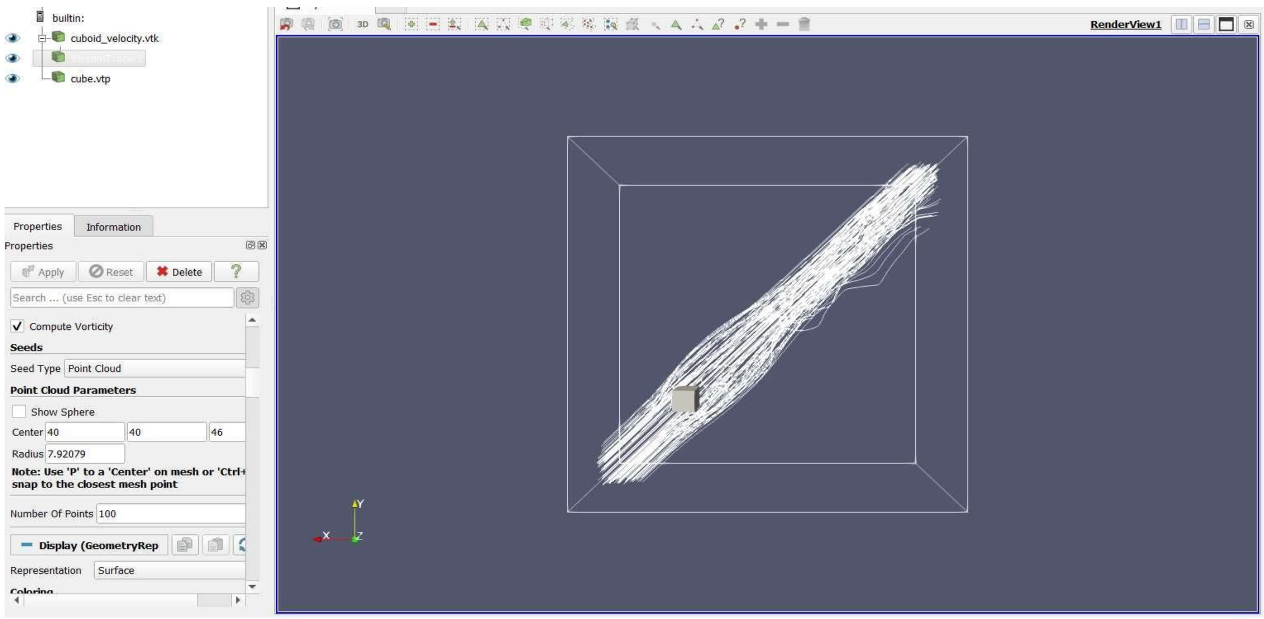
Vector Field Visualization in ParaView

(a)

Seeding in front of the obstacle

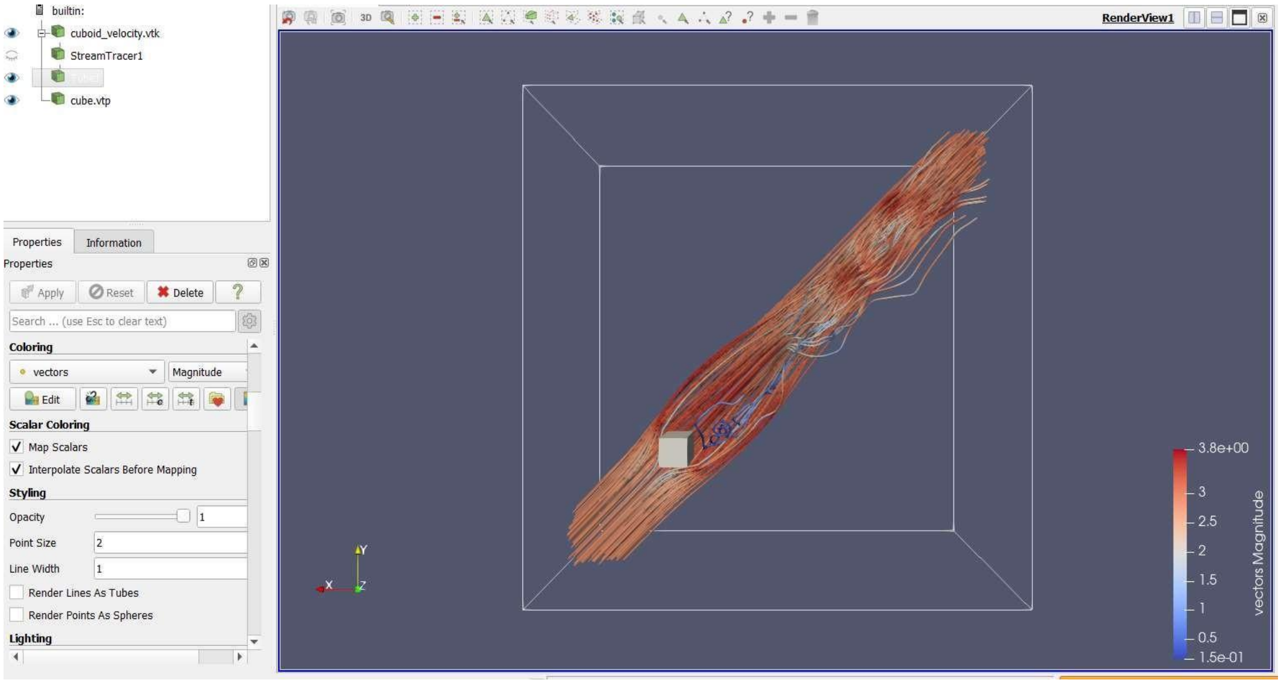


Seeding from behind the obstacle



Regarding the reason to prefer one seeding strategy over the other, it depends on the specific flow and the information you want to visualize. Seeding in front of the obstacle can help you visualize how the flow interacts with the obstacle and how the velocity vectors change as they encounter the cube but the cube is not very clearly visible. Seeding behind the obstacle can provide insights into the wake region and how the flow behaves after passing the obstacle as the cube is very clearly visible

(b)



The flow speed regions can be easily observed with the respective color from the color map given in the bottom right corner of the above picture. The region right after the obstacle show the increase in flow speed in the majority of the streamlines but very few of them show the low flow sneed as well indicating the color blue.

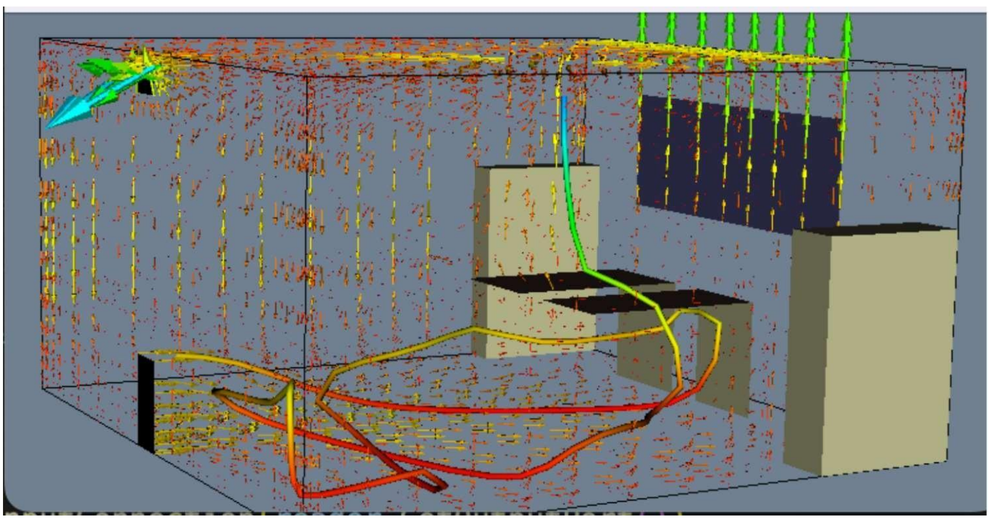
(c)

Streamline integration is terminating very few streamlines on maximum streamline length. And it seems like the very few streamlines are terminating because the maximum number of integration steps are reached. This can be changed by the "Maximum Steps" parameter in the StreamTracer properties. Once the maximum number of steps is reached, the integration is halted.

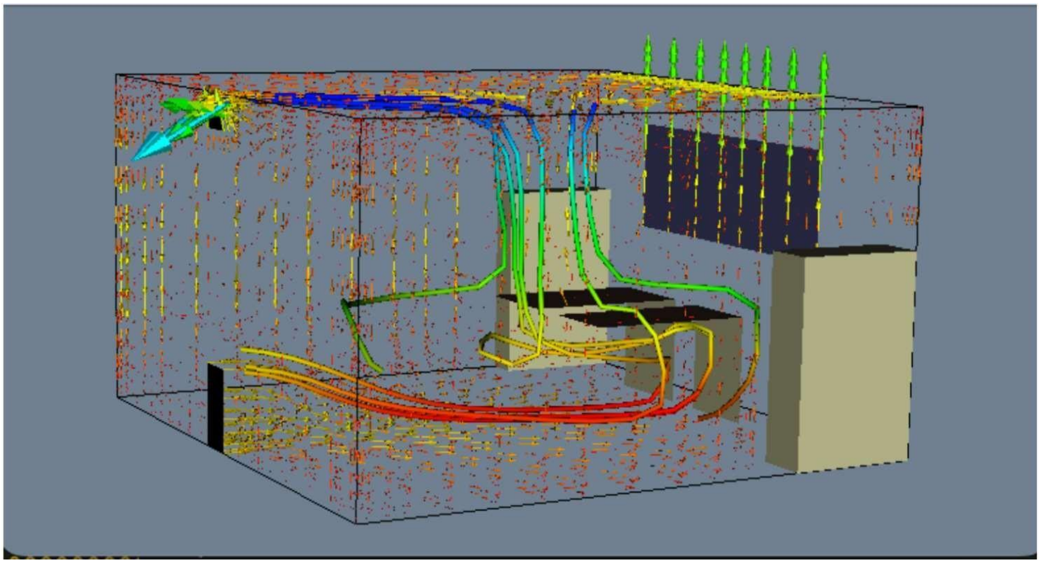
When we substantially increase the maximum stream line length parameter, the reason for termination value drops gradually.

Vector Field Visualization in VTK

(a)



(b)



(c)

By changing InitialIntegrationStep from smaller to larger values (0.1, 0.4, 0.6, 1) incase of smaller values streamlines were smoother and more detailed visualization and especially in regions with rapidly changing flow patterns. In case of large values e.g. "1" resulted in fewer samples along the streamlines. Thus leading to less smoother visualization.

(d)

Compared to the fourth order Runge-Kutta scheme, Runge-Kutta45 shows very less effect of InitialIntegrationStep, there was almost no big difference between the first two initial step values 0.1, 0.4 and the last two initial step values. And the reason for that is: The fourth-order Runge-Kutta scheme uses a constant step size throughout the integration process. In this case, the InitialIntegrationStep determines the initial step size for all the streamlines. On the other hand, the embedded scheme Runge-Kutta45 is an adaptive-step integration method. It dynamically adjusts the step size based on the local conditions of the vector field, aiming to maintain a desired level of accuracy. This adaptive behavior allows the scheme to adaptively refine the step size in regions where the flow is complex or rapidly changing and increase the step size in regions with smoother flow behavior.