Assignment 2 - Velocity Fields

Name: Aman Bhardwaj

Student ID: 12333472

Introduction

This report covers the implementation of various simulation tasks of semiconductor device fabrication using the ViennaLS library and the Level Set Method. The ViennaLS library provides an efficient and flexible platform for the simulation of level set-based operations like etching, deposition, and advection in a computational domain. The tasks outlined involve understanding the basics of level set methods, simulating physical processes such as the melting of materials and etching via the Bosch process, and finally creating a FinFET transistor structure by deposition and etching steps. The simulation aims to demonstrate the effectiveness of level set methods in modelling the fabrication processes in semiconductor devices.

Task Overview

1.1 Creating a Sphere/Snowball

The first task involved creating a simulation domain using the ViennaLS library and extracting the surface mesh and visualizing it in ParaView. In this task, a simple sphere is created placed at the origin. The sphere serves as the basic unit of the simulation, analogous to a snowball in the real world. This task serves as an introduction to understanding how ViennaLS represents surfaces and geometries using level sets.

1.2 Melting the Sphere

To simulate the melting of the sphere, advection of the level set was performed using the lsAdvect class. This class requires a level set domain and a velocity field that describes how the surface evolves over time. A custom velocity field was implemented, returning a constant velocity of -1 at every point on the surface, simulating the melting process. This task introduces the concept of advection in level set methods and how ViennaLS handles velocity fields and material evolution.

1.3 Multiple Materials

The next step involved handling multiple materials, as seen in the construction of a snowman comprising multiple spheres. The simulation required layer wrapping during advection to maintain stable numerics and the use of Boolean operations to combine level sets.

2.2 Emulating a Real Process and using the Bosch Process

This task emulates the Bosch process for deep via etching, including:

- Creating a 3D domain with two reflective and one infinite boundaries.
- Defining a substrate and depositing a mask layer using Boolean operations.
- Simulating isotropic deposition and directional etching cycles using custom velocity fields.

This task is designed to simulate complex real-world processes and demonstrate how changing process parameters affects the results, such as sidewall roughness and etching depth.

3.2 Creating Transistor Structures (FinFET Transistor)

This task involves the creation of a FinFET structure using the following steps:

- Depositing layers of oxide, silicon, and creating mask by Boolean operation.
- Etching and depositing materials to create the fin and gate structure.
- Using Boolean operations for Chemical Mechanical Planarization (CMP) to flatten the gate material.
- Finally, etching Gate and Spacer directionally.

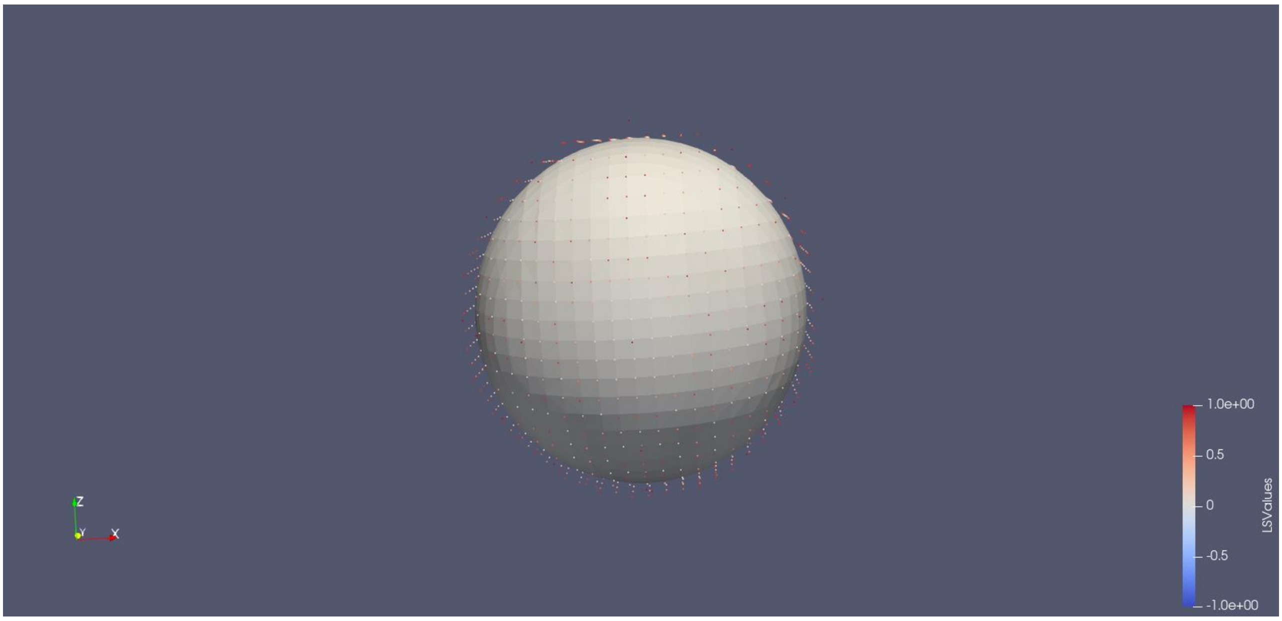
This task highlights the ability to model complex semiconductor devices and emphasizes the influence of process parameters on generating the final FinFET structure.

Results

Task 1.1: In this task, a sphere (snowball) was successfully created within a 3D simulation domain. The sphere was centered at the origin, and the surface mesh was extracted and visualized in ParaView. The result is a perfectly spherical surface represented by level sets. The sparse level set representation of ViennaLS ensures efficient handling of this geometry. Visualizing the sphere in ParaView confirmed that the surface mesh matched the defined radius and grid spacing.

The files snowball_surface.vtk and snowball_grid.vtk were generated successfully by using the command line argument: python Task1.1.py. The simulation domain and Sphere of radius 10.0 units were created.

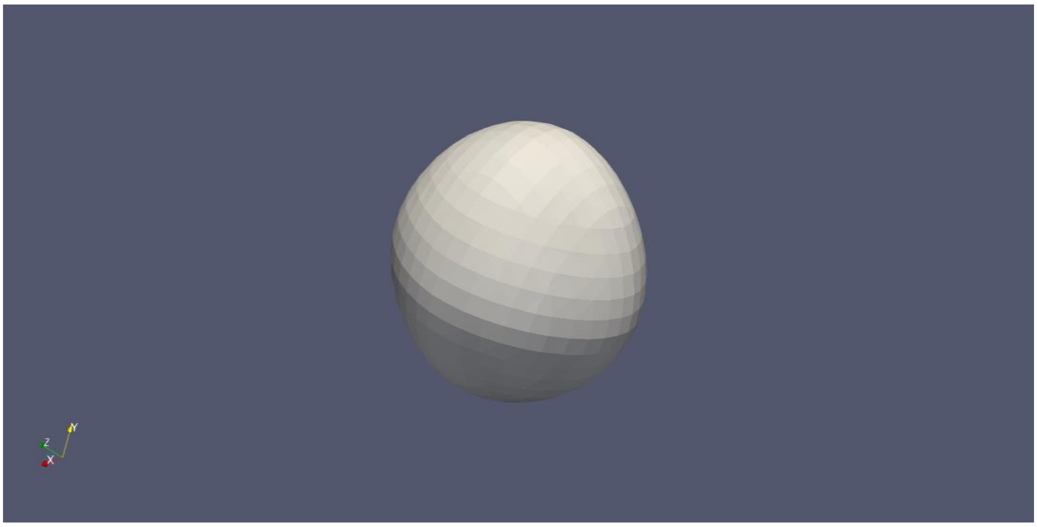
Below is the visualization of how the surface looks like in ParaView:

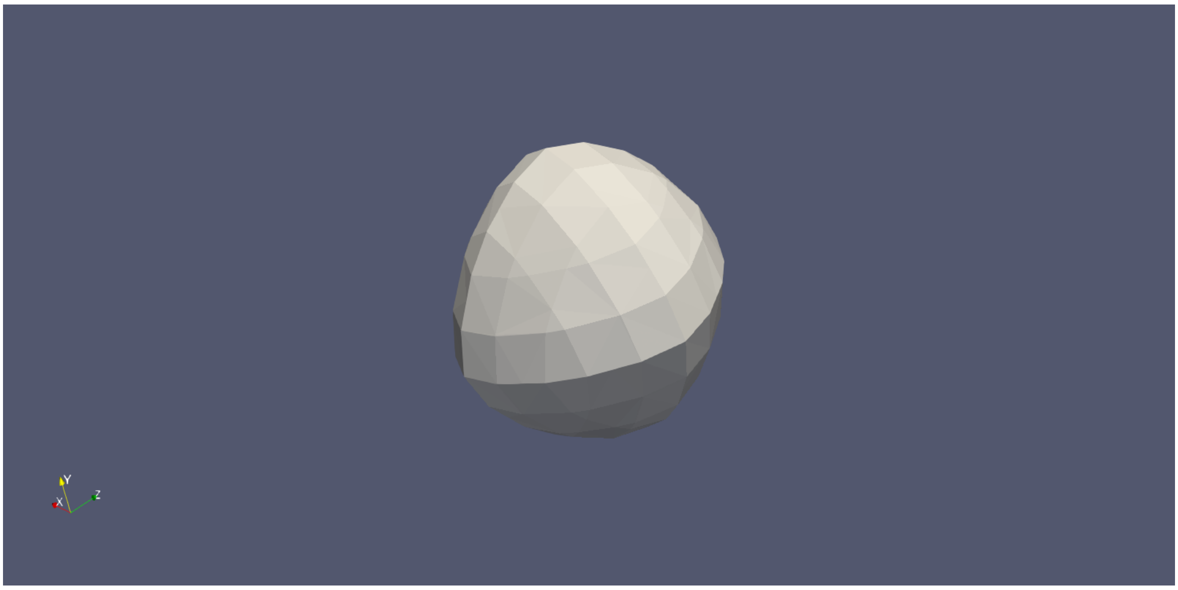


Task 1.2: This task involved melting the sphere by simulating an advection process using a constant velocity field of -1. The snowball gradually shrank over time due to isotropic advection, simulating the effect of melting. Each advection step was stable, adhering to the CFL condition, ensuring no numerical instability.

After running the advection for a given time, the sphere reduced in size, and the final surface was visualized in ParaView. The shrinking process was smooth and uniform, confirming that the velocity field correctly drove the surface inward. Key observations:

- The melting was uniform, as expected from an isotropic velocity field.
- ViennaLS handled the advection steps efficiently, ensuring stability.
- The process could be visualized dynamically, showing real-time evolution.
- I used two times for advection, one for **t = 6 and other for t = 8** to visualize the melting of sphere at two given times. Here are the visualizations at different times for the melting of sphere, through cmd command python Task1.2.py:



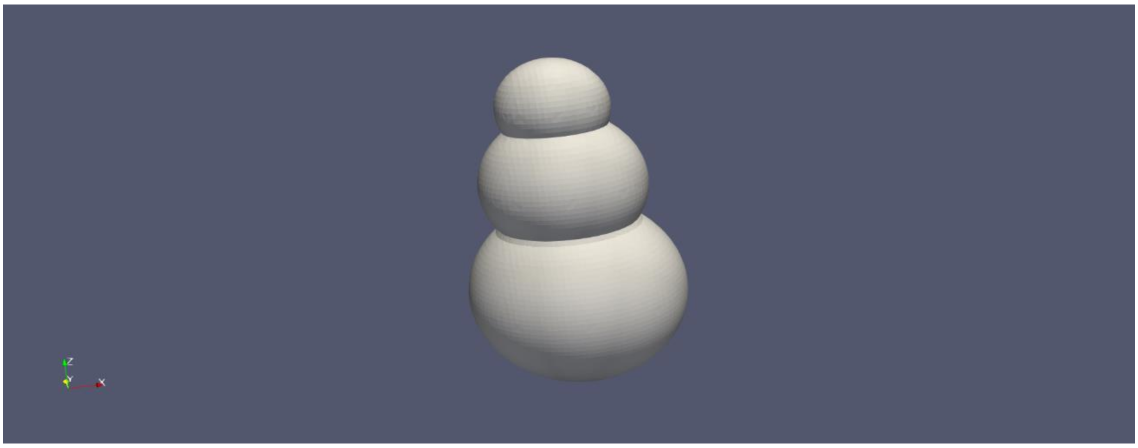


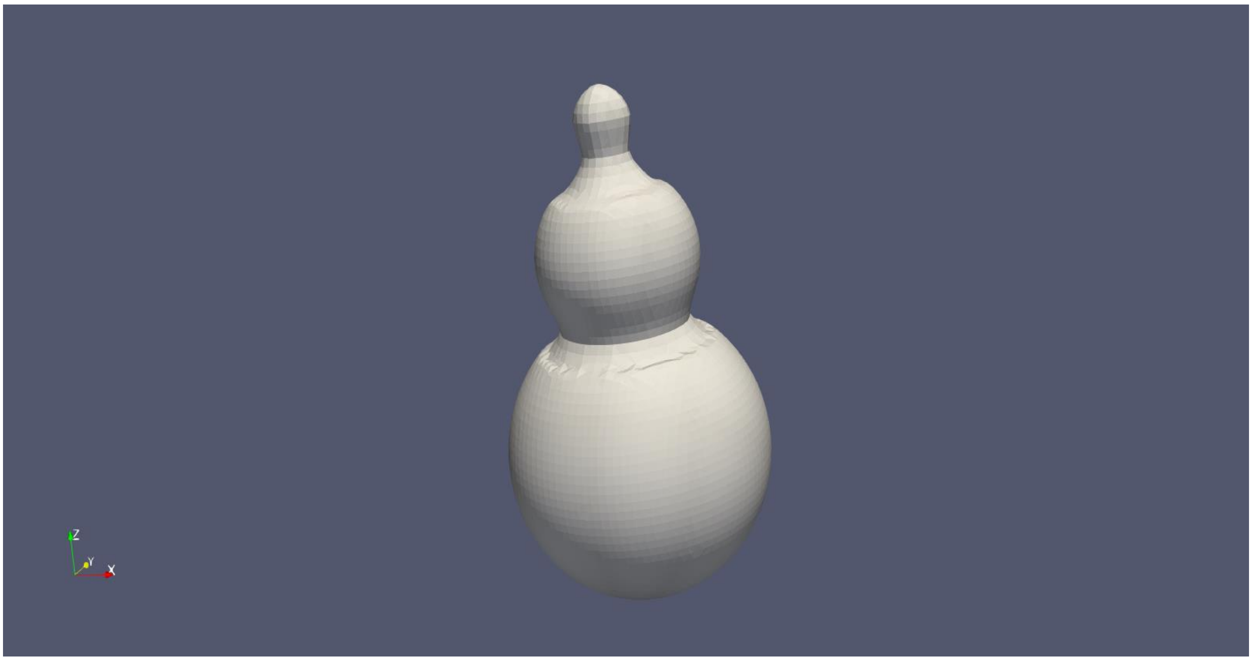
Task 1.3: In this task, multiple materials were simulated using ViennaLS's capability to combine level sets through layer wrapping and Boolean operations. This is used for melting the Snowman which is made up of three spheres. Two different materials (spheres) were combined into a single level set using Boolean Operation, allowing for stable numerics during advection.

The velocity fields for each material were defined separately, allowing only the topmost material to move while lower layers adjusted accordingly.

Key observations:

- Boolean operations worked seamlessly to combine level sets.
- Different velocity fields could be applied for each material, allowing for realistic simulations of material behaviour.
- Visualization confirmed that the wrapping method was stable and efficient.
- The advection time was set at t = 2 and it showed that the snowman melted, and the top layer melted the most. The program is run through cmd by python Task1.3.py. Here is the visualization of the melting of snowman at **t = 0 and t = 2**:

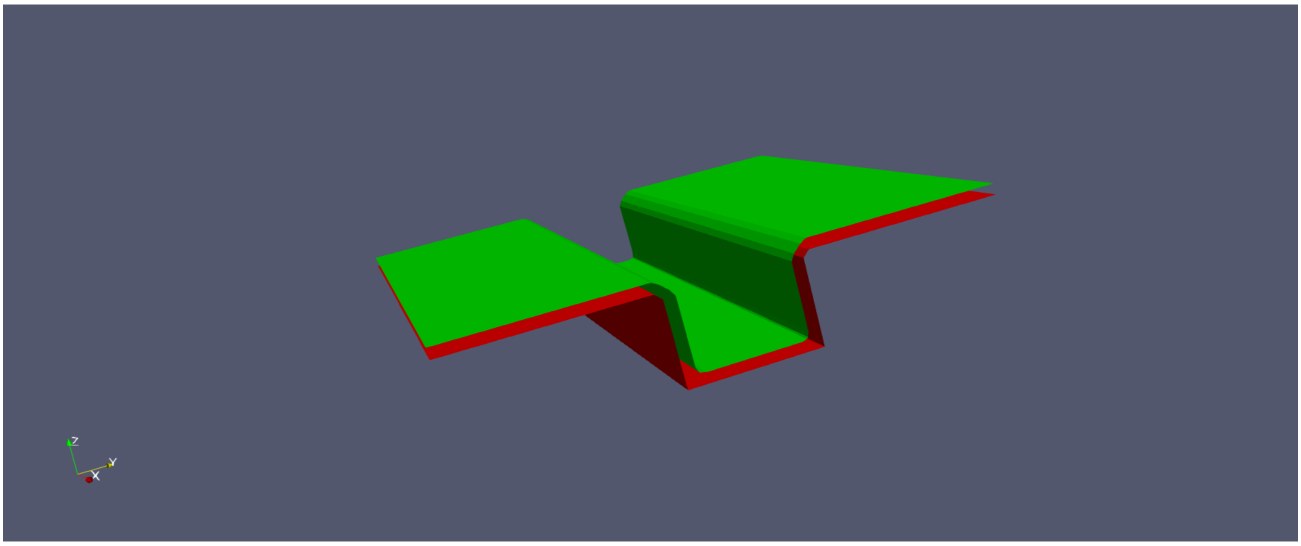




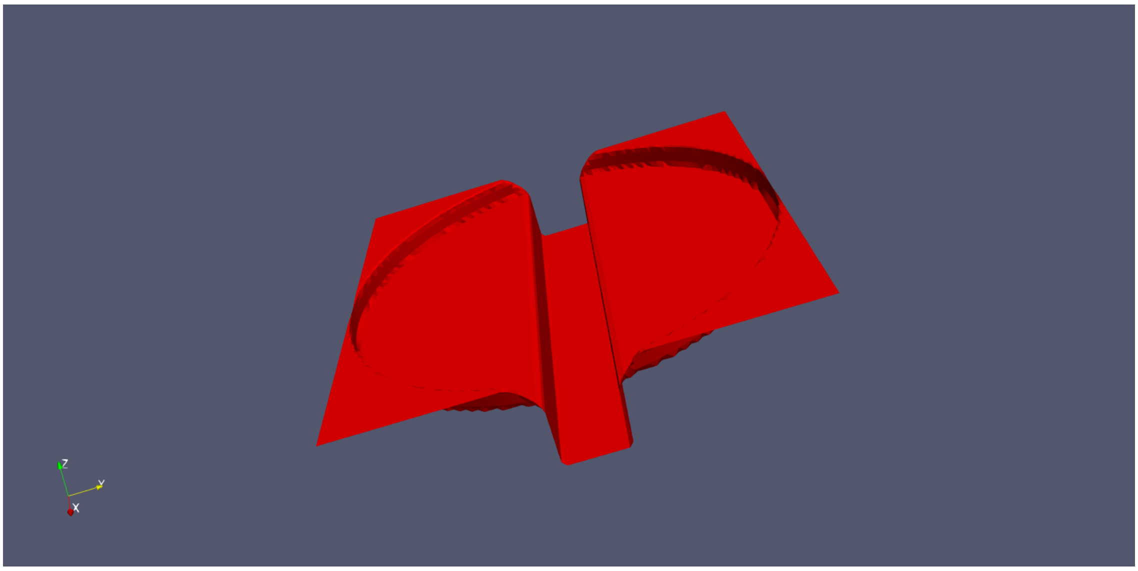
Task 2.2: In this task, the Bosch process was simulated by creating a substrate with a mask layer and applying alternating cycles of isotropic deposition and directional etching. The resulting via had a deep hole with scalloped sidewalls, characteristic of the Bosch process. First the circular hole was made in the mask layer, then emulating several cycles of the Bosch process created a deep hole in the substrate. Through cmd command, the task is executed: python Task2.2.py

Initial Mask Creation

- **Substrate and Mask:** A substrate was created as a trench and a mask material was deposited on top. The mask is designed to cover the substrate surface, acting as a protective layer during the etching process. The visualization of substrate and mask layer looks as under:



- **Circular Hole Generation:** A Boolean operation is performed to generate a circular hole in the mask but not in the substrate. This allows the etching process to be localized to the exposed part of the substrate, with the hole spanning most of the simulation domain. For this, a Cylinder shape is used to define the cylindrical hole. The visualization looks like this:



Emulation of the Bosch Process

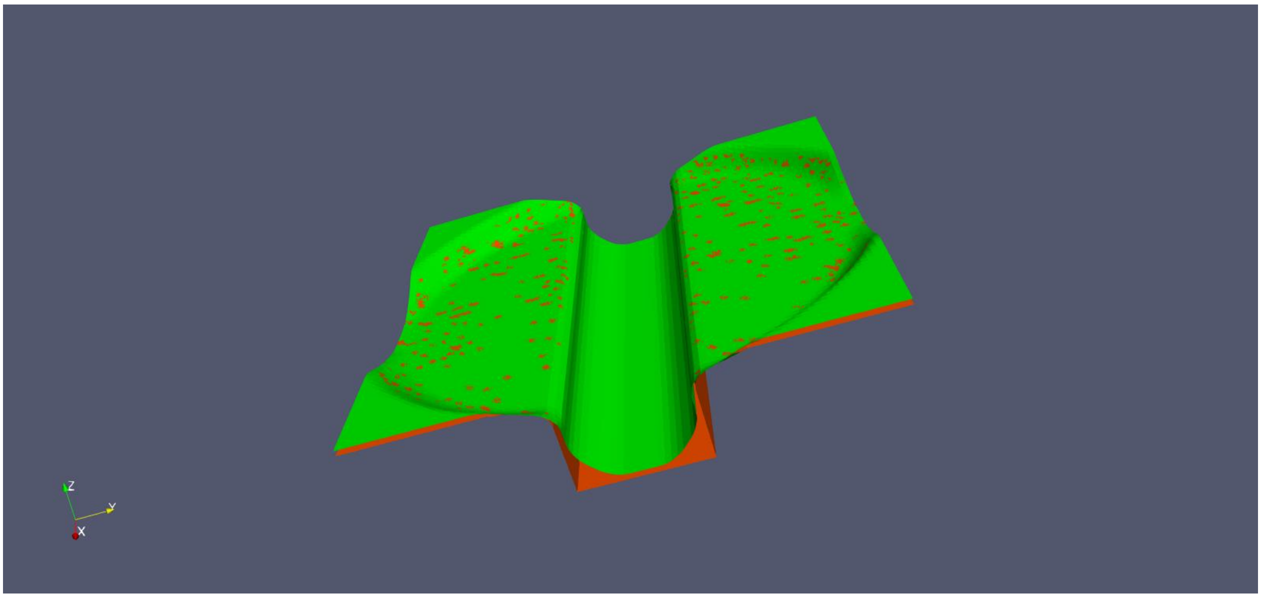
- The Bosch process consists of alternating perfectly isotropic deposition directional etching steps, creating the deep hole in the substrate through several cycles. The deposition is used to passivate the sidewalls, while the etching removes the passivation layer in a directional manner and etches the substrate.

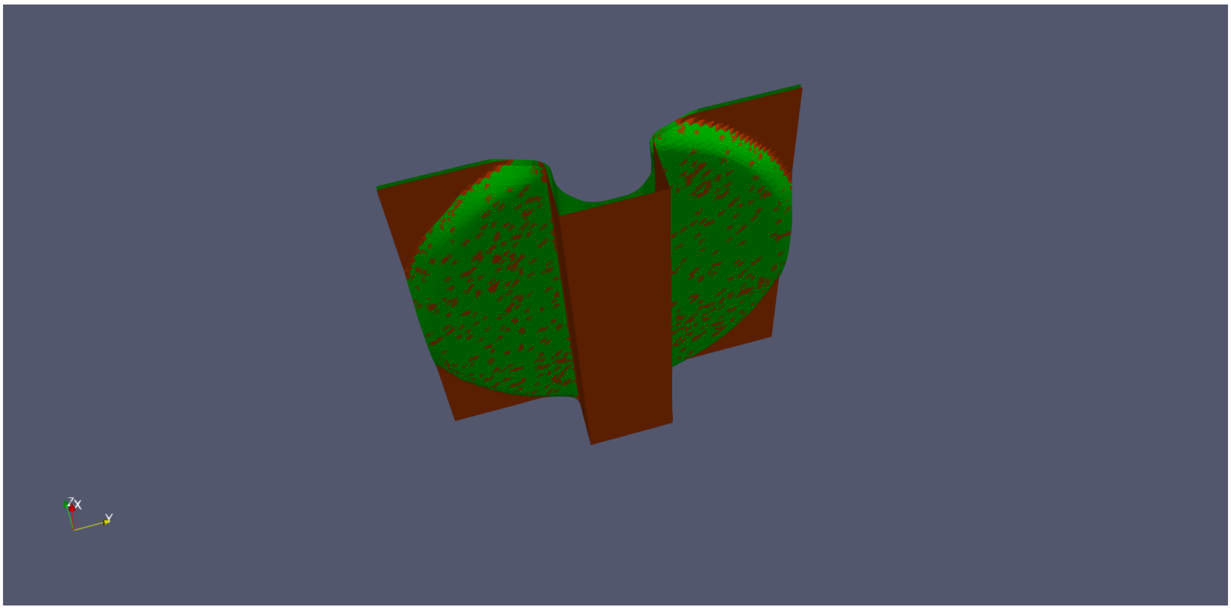
  <u>Deposition and Etch Rates</u>:

- The rate of deposition controls how much material is laid down during each cycle, affecting the thickness of the passivation layer. The substrate etching is assumed to be perfectly isotropic, meaning the etch rate is uniform in all directions. The velocity field used for etching depends on the material, with different velocities assigned to the mask, silicon substrate, and deposited material. I used both deposition and etch rates as 1 which created a deep circular hole in the substrate with **number of cycles = 12**, which gave satisfactory results. Here is the visualization of the process:

  Time passed during the advection process = 4.241792194330132

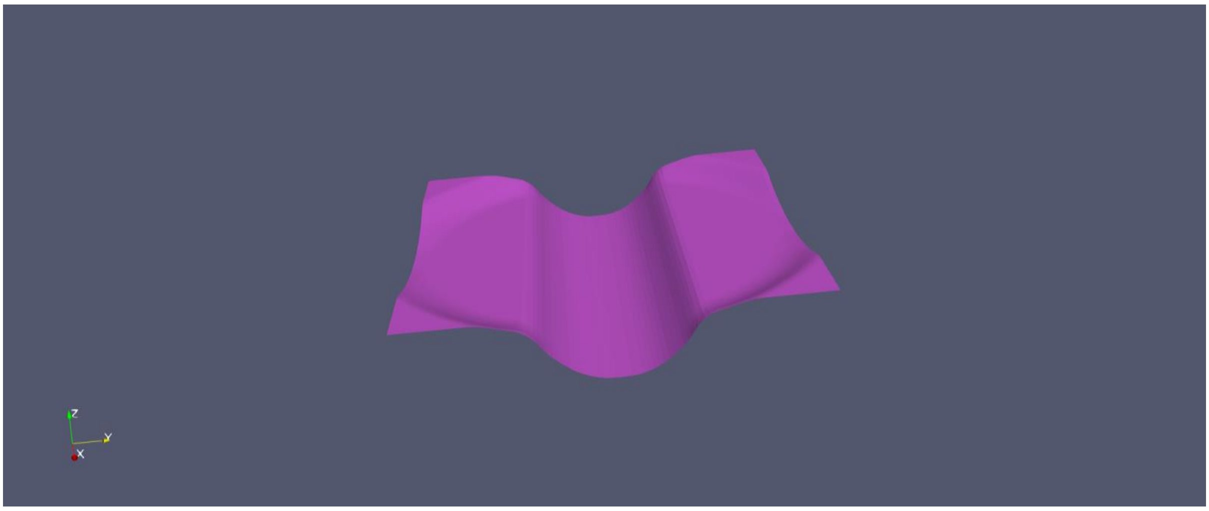
  I used colours in ParaView to describe the surfaces. These two images show the deep hole in the substrate using the rates:





Cycle Times and Profile Evolution:

Now, if I increased the **number of cycles to 30**, the visualization looks like this:



This shows that the hole has been etched deeper into the substrate, but the hole is less uniform as compared to that of lower number of cycles. It can be concluded that:

If the cycle time is less like 12, the resulting hole profile has smaller grooves along the sidewalls due to shorter deposition/etch durations. However, a shorter cycle time resulted in comparatively less depth per cycle but more uniform hole.

With longer cycle times like 30, each cycle etched deeper into the substrate, and it made the substrate curved along with the mask. Also, this increased the roughness on the sidewalls as the larger grooves make the hole less uniform.

The sidewall roughness is influenced by the balance between the deposition and etching times. Here it can be seen that sidewall roughness is increased with longer cycle, as the groove size is larger.

The depth of the etched hole per cycle depends on the rates of deposition and etching. Faster etching with slower deposition resulted in deeper etching per cycle, while slower etching with faster deposition reduces the etch depth with small hole.

Effect of Increased Isotropy in Etching:

If the etching process is made more isotropic (i.e., the etch rate becomes more uniform in all directions), the hole profile becomes more rounded, and the sidewalls will lose their vertical character. This can lead to less anisotropic profiles, but more rounded, bowl-shaped features as can be seen from the figures.

When the etch process is perfectly directional (e.g. z-direction only), the sidewalls remain steep and vertical, with grooves caused by the deposition/etch cycles.

So, by making etch process to be more isotropic, the final structure's sharpness can be controlled. More directional etching results in vertical sidewalls, while increased isotropy leads to sloped, more rounded sidewalls as can be seen from the visualization.

Properties Affecting Scallop Size:

Deposition Rate: The rate of isotropic deposition plays a crucial role in determining the size of the scallops along the sidewalls. Faster deposition rates lead to thicker layers as more material is deposited, resulting in larger scallops after etching.

Etch Rate: A higher directional etch rate creates deeper etching per cycle, which influences scallop size as well. Faster etching rate results in smaller scallops, while slower etching rate leads to larger scallops.

Cycle Duration: The length of each deposition/etch cycle also impacts scallop size. Longer cycles lead to large scallops due to the greater amount of material deposited and etched away per cycle as can be seen from the figures.

Task 3.2: Creating Transistor Structures (FinFET Transistor):

1) In the final task, a FinFET transistor was successfully created using multiple steps of deposition, etching, and planarization. The structure consisted of a silicon fin on an oxide layer with a gate material deposited around it. I also created a silicon substrate of height 50 nm below the oxide layer to resemble the structure as given in the assignment description. The process followed the steps outlined, including mask creation, directional etching, and CMP for gate flattening.

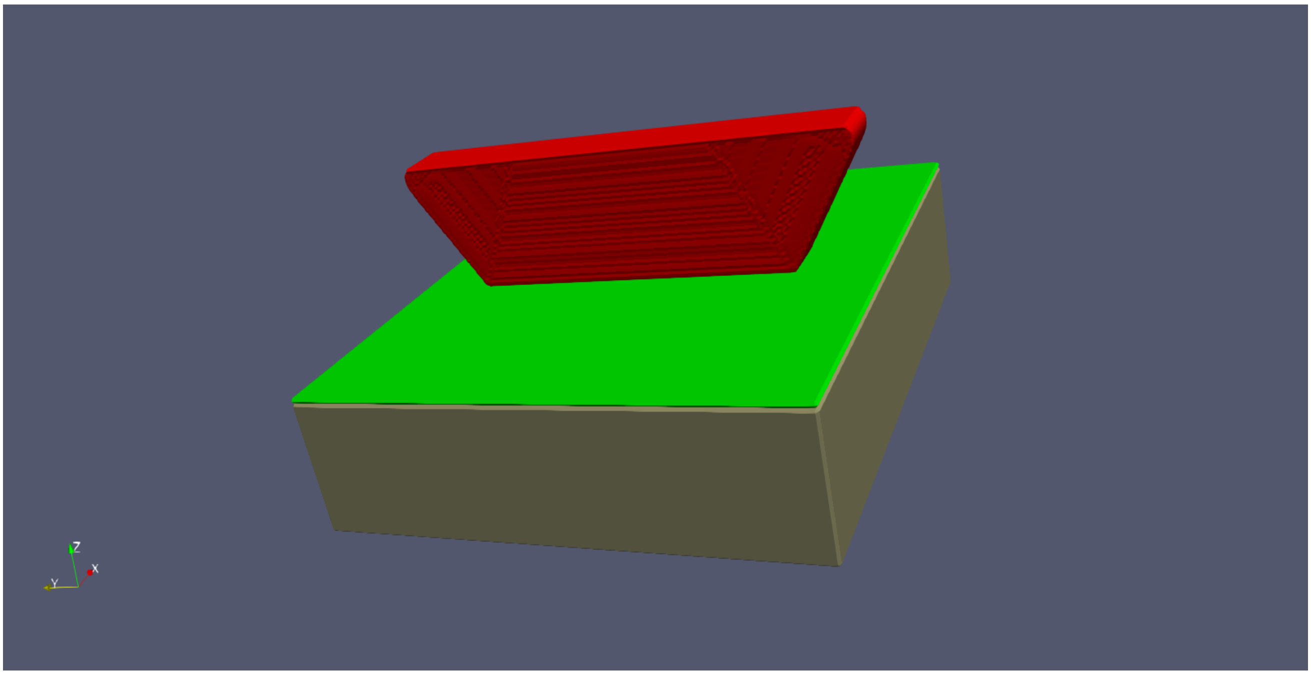
I created an oxide layer of 200x200nm and deposited 50 nm of silicon on it isotropically. I used **deposition steps = 50**, so that the material is deposited evenly in all directions. It was observed that more steps resulted in smoother deposition and allowed for finer control over the thickness. Fewer steps lead to uneven surface so using more deposition steps was better.

The function getScalarVelocity controls the deposition velocity on certain regions (e.g., top of the oxide). A value of positive value = 1.0 here means that silicon grows only on the top, preventing sidewall growth. A higher velocity value results in faster deposition.

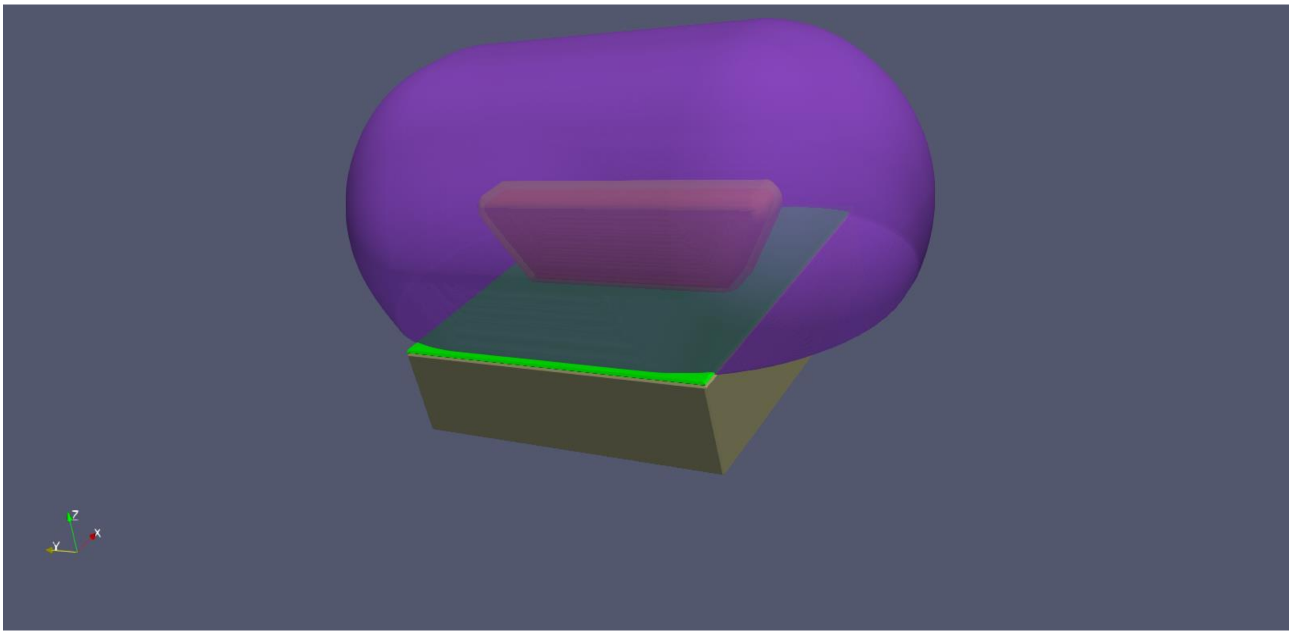
Then a mask of 20nm width was created on the deposited silicon.

Now, for etching, Directional Etching was controlled by the velocity fields, where a negative velocity (-1.0) removes material. The etching steps, set by **etch_steps = 6** with a **depth of 60.0 nm**, control the degree of material removal. Increasing the number of steps allows for finer resolution, but takes time for processing, while fewer steps result in comparatively less accurate etch profiles. The command to run this file on cmd is python Task3.2.py.

The use of a simple Advection scheme (vls.Advect()) with default advection ensured a stable solution. After the etching, mask was removed and final_silicon_fin_etched.vtp file was generated with created a fin on top of the oxide layer after the etch process. The visualization looks like this in ParaView:

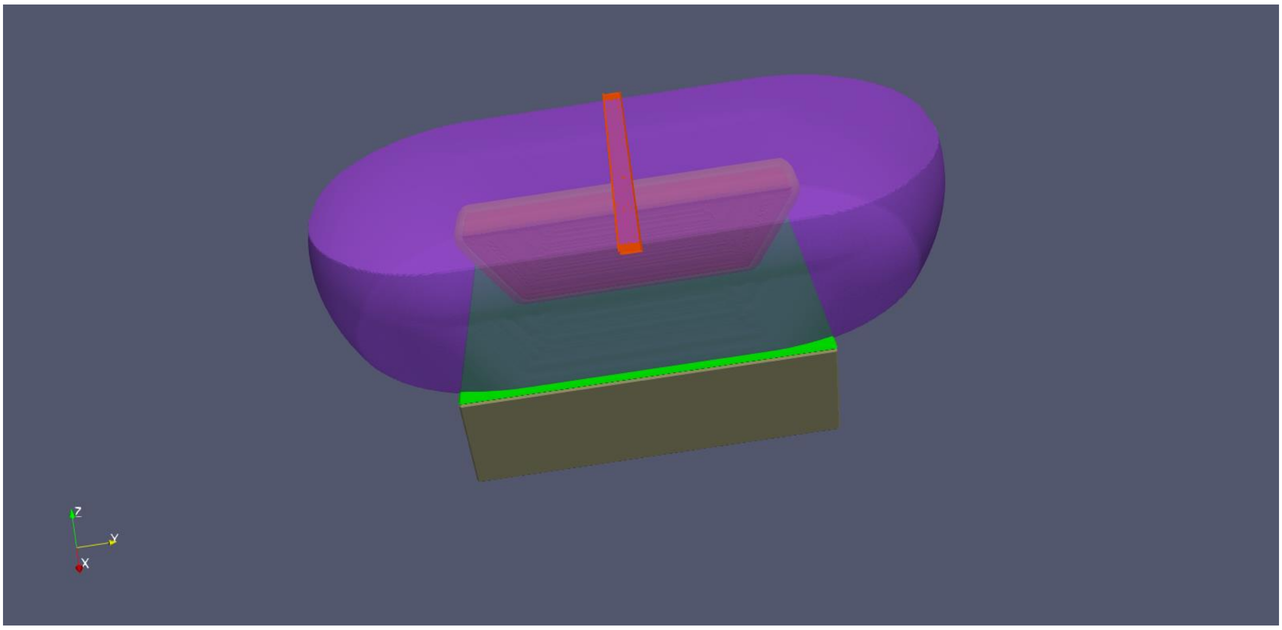


Now, after obtaining the fin, spacer of 5 nm thickness is deposited everywhere with **spacer_steps = 5** for finer spacer deposition. After this, 80 nm of Gate material was deposited everywhere. I used **gate_steps = 8**, for depositing gate material of 80 nm thickness. After the deposition of gate, the structure looks like this:



For CMP, a plane at height of 70nm above the Oxide was created to flatten the Gate Material above it. Then, a 10nm wide mask (box) perpendicular to the etched fin on top of the Gate after the CMP was made. The result is saved as cmp_result.vtp.

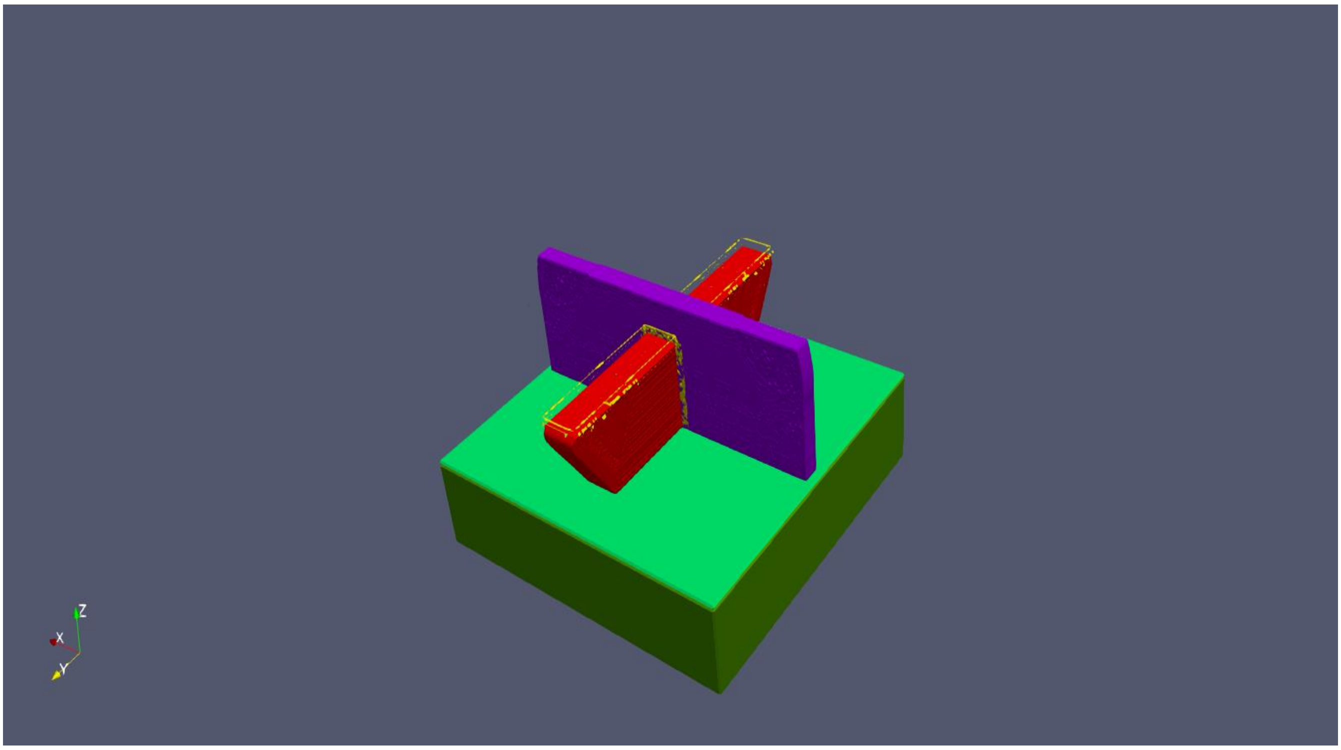
The structure looks like this after the CMP using Boolean operation and the attribute **RELATIVE_COMPLEMENT**, and thus, the gate material was removed above 70 nm height from the oxide layer:



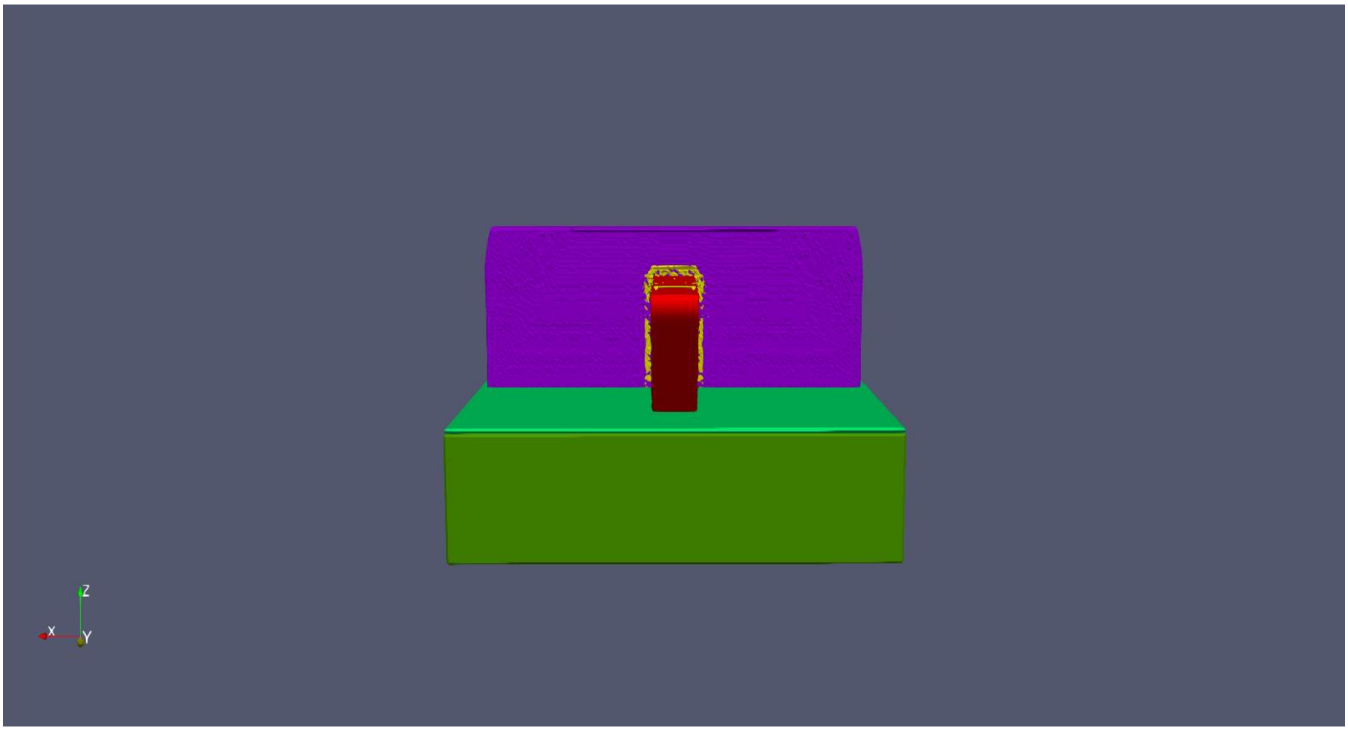
Now finally, the Gate and Spacer are etched directionally, resulting in a 10nm wide Gate. Also, the gate mask is removed before saving the final patterned gate. The intermediate files with cycles for deposition and etching of material are also saved. The final FinFET structure is saved as FinFET_structure.vtp

All the intermediate results are combined into one structure by Boolean operation Union.

The final structure of the FinFET obtained looks like this below:

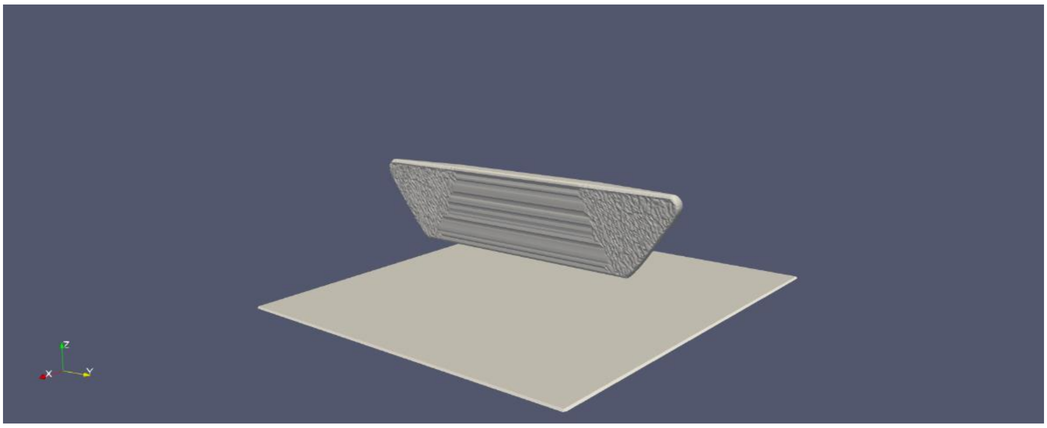


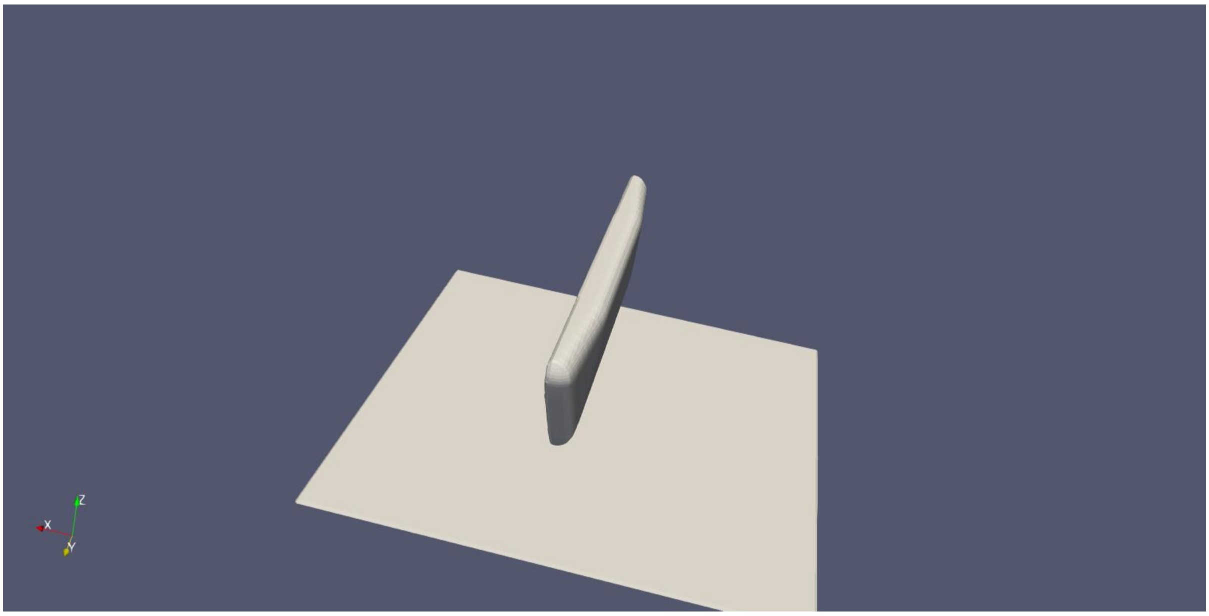
The cross section view looks like this, with yellow showing the spacer, red the etched fin, purple the etched gate and green the oxide layer:



2) Impact of Isotropy in Directional Etching:

- Adding isotropy to an otherwise perfectly directional etch causes lateral etching, leading to undercuts or sidewall erosion. In FinFET creation, isotropic components cause the fin structure to shrink laterally, making it narrower and potentially less defined.
- In the etching process, to add isotropy, the etch velocity field should have a small isotropic component, I changed the vector velocity value of the EtchVelocityField class in the code for Fin creation from (0.0, 0.0, 0.0) to (0.1, 0.0, 0.2) for slightly isotropic in the etch step for the deposited silicon to obtain the fin. It can be seen that the etch was not perfectly vertical, and sidewalls are also affected. Also, it can be seen that the fin structure is shrunk with sidewall erosion and not perfect finish after etching, the surface of the sidewall looks uneven. In the second image, it can be seen that the other sidewall is a bit curved due to some isotropy.





3) When directional etch processes such as **Fin creation, Spacer Etch, and Gate Patterning** include **positive isotropic components**, the etch begins to remove material not only in the vertical (z-axis) direction but also in lateral (x and y-axis) directions. I used this in the code by changing the velocity vector from (0.0, 0.0, 0.0) to (0.1, 0.0, 0.2). This introduces some unintended side effects such as:

Impact of Positive Isotropic Components:

1. Fin Creation: The fin structure will become narrower and more rounded at the top edges and sidewalls. The originally sharp vertical sidewalls of the fin will be eroded, and undercutting occurs, where the base of the fin is etched more than the top. This results in a loss of fin height and a smaller fin width, which would affect transistor performance.

2. Spacer Etch: In spacer etching, the isotropic component would cause lateral erosion, leading to thinning of the spacers on the sides of the gate.

3. Gate Patterning: The gate region would be more rounded rather than sharply defined. This can affect the gate length and alter the electrical characteristics of the transistor.

A process that introduces isotropic etching while still retaining some directional behaviour is called **Reactive Ion Etching (RIE)** which can achieve this, with specific adjustments. RIE can be made slightly isotropic by adjusting the balance between chemical and physical etching components.

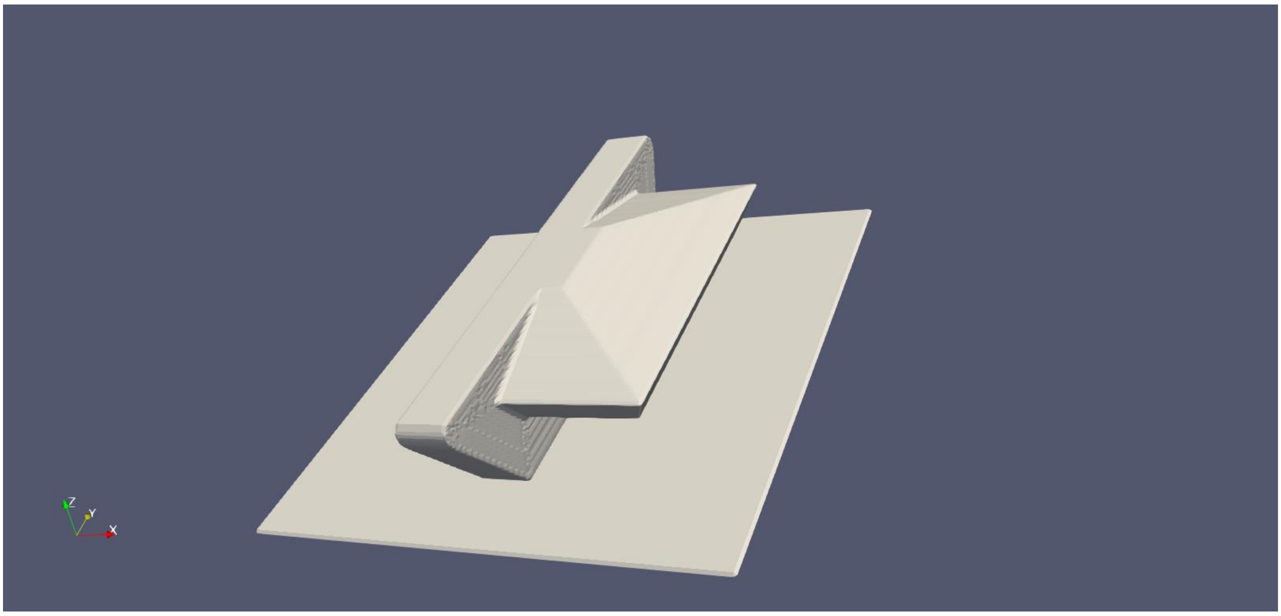
4) When the direction of directional processes (such as plasma etching) is skewed, simulating non-uniform plasma behaviour on the edges of wafers, the following changes and effects occur in the etched structures:

1. The etching profile will no longer be perfectly vertical. Instead of clean vertical sidewalls, there are slanted or uneven profiles. The skew causes the etch to lean in one direction, resulting in asymmetry in the structure.

2. Skewing the etch direction creates deeper or shallower etch rates, which could lead to incomplete or over-etched structure.

3. If gate patterning is done under skewed etch conditions, the gate's width or height will not be consistent. This non-uniformity can affect the electrical performance, as the gate controls the flow of current through the channel.

Here in the code, I changed the normal direction of getScalarVelocity from normal[2] < 0 to normal[2] < 1 for the EtchVelocityField class for the etching of silicon fin and it can be observed that the etching process is not uniform and perfectly vertical, it creates a skewed structure with slant edges and sides while etching the silicon fin as shown below in the visualization in ParaView:



5) Advanced Advection Scheme: Lax-Friedrichs:

The Lax-Friedrichs scheme is a more advanced integration method that better handles sharp transitions and varying velocity fields. It can provide more accurate results when modelling directional etch steps with complex velocity fields (like those with isotropic components).

Using Lax-Friedrichs will result in more accurate etch profiles, especially in cases where curvature or sharp features are present. The etch steps will better align with the intended velocity fields, leading to more precise patterning and less rounding of edges.

In the code I used this scheme in the etching step of silicon fin to configure the advection kernel for directional etching use the code in python script and using vls.Advect():

```
etch_kernel.setIntegrationScheme(vls.IntegrationSchemeEnum.LAX_FRIEDRICHS_1ST_O
RDER)
```

Here is how the visualization of the etched fin using this scheme looks like: