5 Results and Validation

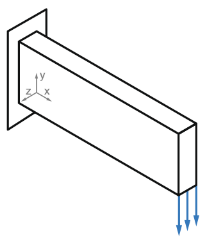
5.1 Validation

The physical density variable vector $\tilde{x}_e$ obtained for elements through the implemented parallel framework has been validated with the physical density variable vectors obtained through an independent sequential topology optimization platform worked upon by $K.Liu.$[[9]]. The design domain and its corresponding optimized structure obtained through this framework have been validated with examples from $K.Liu.$[9],$Hunter$[7]

5.2 Results

5.2.1 Cantilever Beam Topology Optimization with Line Load Along Edge

Platform Execution Parameters:
Number of elements($X - Direction$):**120**
Number of elements($Y - Direction$):**20**
Number of elements($Z - Direction$):**10**
Penalization Factor(p):**3**
Filter Radius($rmin$):**1.5**
Volfraction(f):**0.15**
Number Of Cores Used:**10**



(a) Initial Design Domain

(b) Topo Optimized Domain

Figure 4: From Reference Paper[9]-elements(x,y,z):(60,20,4) ,p=3,f=0.3,r=1.5

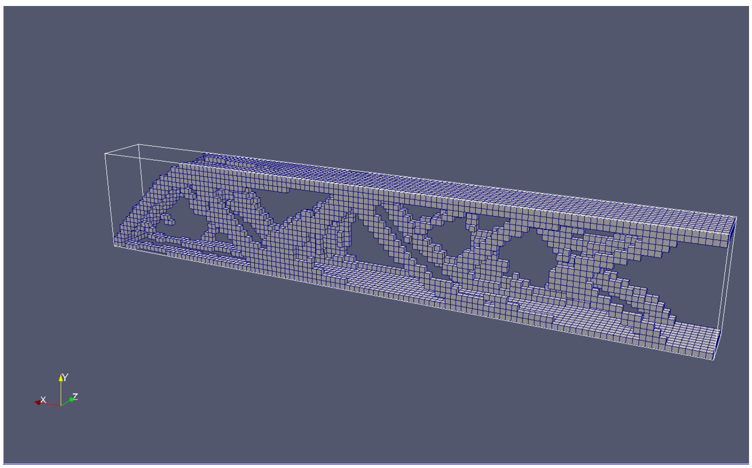


Figure 5: Optimized Design Domain visualized with Paraview

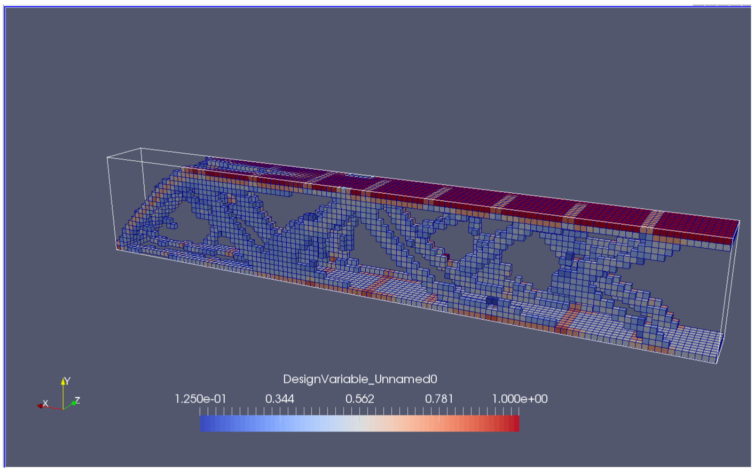
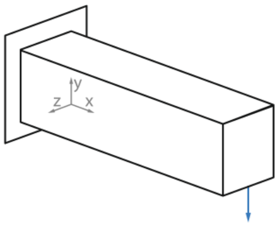


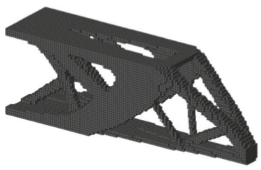
Figure 6: Optimized Design Domain with Gradient Distribution of Density Variables

5.2.2 Cantilever Beam Topology Optimization with Single Load at Lower Edge

Number of elements($X - Direction$):**100**
Number of elements($Y - Direction$):**40**
Number of elements($Z - Direction$):**10**
Penalization Factor(p):**3**
Filter Radius($rmin$):**1.5**
Volfraction(f):**0.15**
Number Of Cores Used:**5**



(a) Initial Design Domain



(b) Topo Optimized Domain

Figure 7: From Reference Paper[9]-elements(x,y,z):(120,40,30),p=3,f=0.15,r=1.5

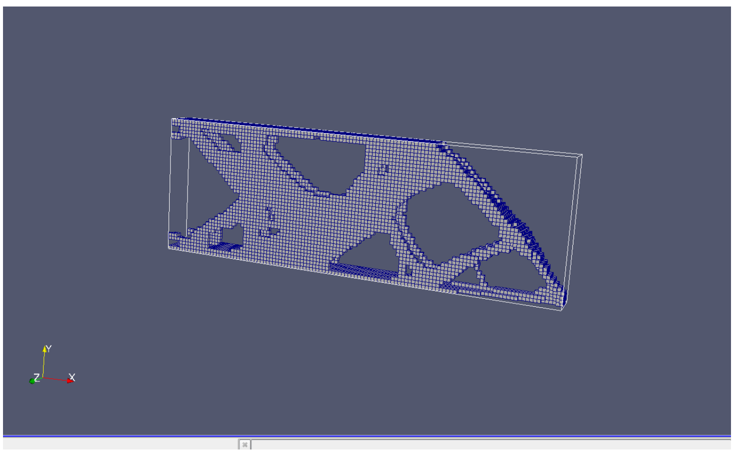


Figure 8: Optimized Design Domain visualized with Paraview

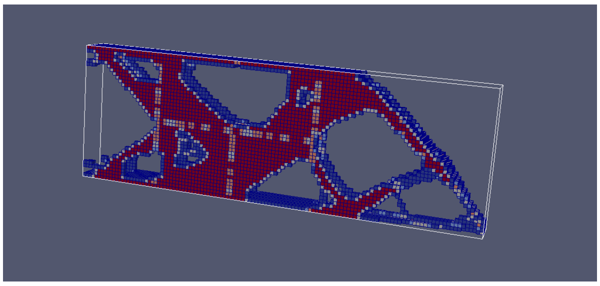
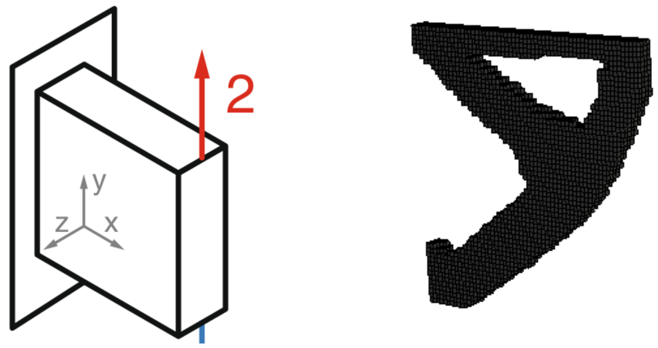


Figure 9: Optimized Design Domain with Gradient Distribution of Density Variables

5.2.3 Cantilever Beam Topology Optimization with Single Upper Load

Number of elements($X - Direction$):**60**
Number of elements($Y - Direction$):**60**
Number of elements($Z - Direction$):**4**
Penalization Factor(p):**3**
Filter Radius($rmin$):**1.5**
Volfraction(f):**0.4**
Number Of Cores Used:**10**



(a) Initial Design Domain

(b) Topo Optimized Domain

Figure 10: From Reference Paper[9]-elements(x,y,z):(60,60,4) ,p=3,f=0.4,r=1.5

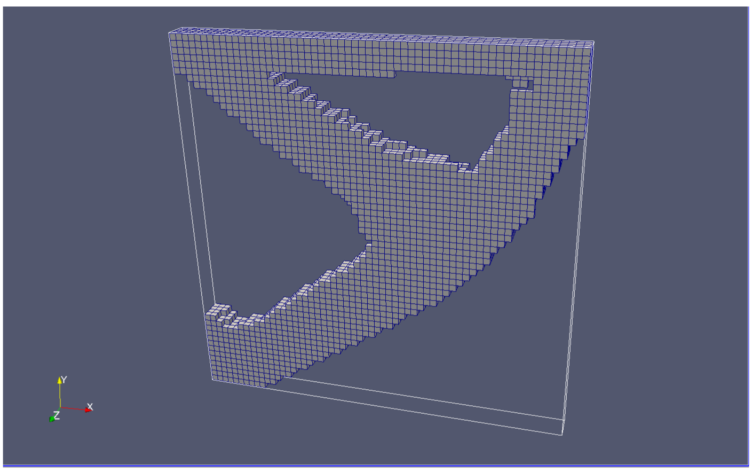


Figure 11: Optimized Design Domain visualized with Paraview

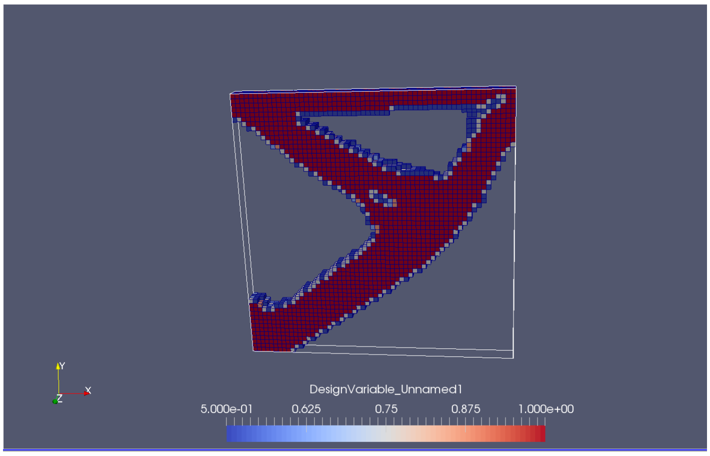


Figure 12: Optimized Design Domain with Gradient Distribution of Density Variables

5.3 Parallel Computing Analysis

The Topology Optimization framework established with PETSc was made to execute on TU Freiberg Cluster where at each program run ,around 4-12 cores stood used.
Two approaches were taken to analyze the Parallel Computing Capability of the framework:

5.3.1 Keeping the number of elements same and varying the cores to check speed scaling

The framework was executed for 3000 elements,by varying the number of cores.The graph and table (1) gives us an inference where the time taken for code execution completion **reduced** with increase in cores.

Cores	Time(in seconds)
1	22.831
3	10.624
6	7.704
10	5.64
12	5.634

Table 1: Number of Cores vs Execution time for 3000 elements

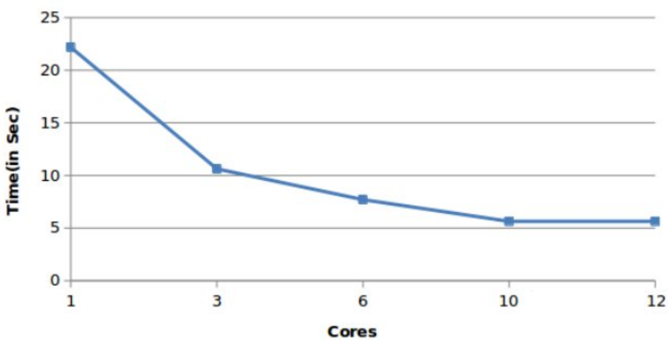


Figure 13: Number of Cores vs Execution time for 3000 elements

5.3.2 Sequential vs Parallel Execution Time Scale Comparison

The time taken to do a sequential run was compared with that of a Parallel run for varying element sets.The observations stand stated below:

Number of Elements	Sequential Run Time (seconds)	Parallel Run Time(seconds)
60	0.2457	0.1937
100	1.1506	1.408
250	1.478	1.393
900	35.814	3.24
1500	91.814	6.24
5000	240.23	8.78
10000	600.23	13.78
15000	3150.13	105.23
30000	6096.23	254.6
60000	8009.23	312.78
150000	14537.9	912.3

Table 2: Sequential vs Parallel(6-10 cores) Time Scale Comparison

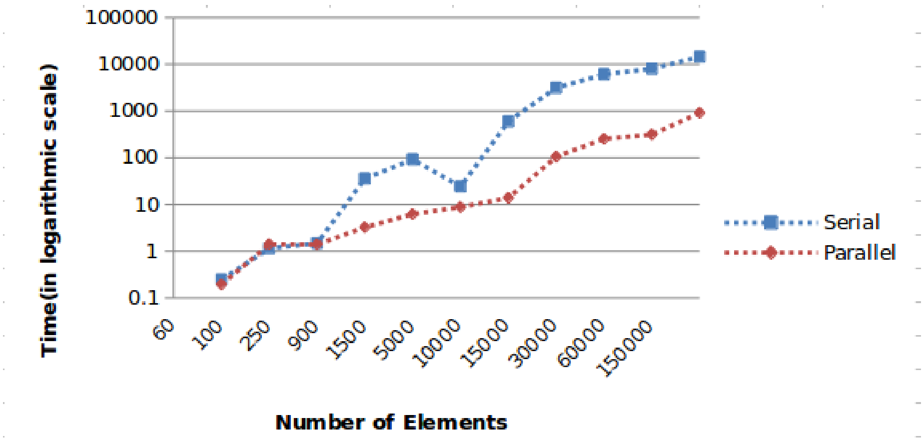


Figure 14: Sequential vs Parallel(6-10 cores) Time Scale Comparison

We observe that till 900 elements there is no significant difference between Sequential and Parallel Execution.But,as we increase the element sets ,we observe parallel computing working faster than sequential runs.

6 Summary and Conclusion

The framework implemented with PETSc presents an easy-to-use ,fully parallelized base for conducting large scale topology optimizations. The framework includes: FEM principles,sensitivity analysis,density filter and optimality criterion optimizer,to help solve minimum compliance problems.

The validity of the framework is demonstrated through case examples and inferences on performance with parallel computing were discussed.Clearly,for large element sets, parallel computing has a significant faster execution time as compared to sequential runs.

Future work,could be the implementation of different optimization algorithms or filters to assess and compare optimization results.Also, various new analytical approaches could be worked upon with large data set generation capability of PETSc.