Software Report

Nguyen Quan Ba Hong

February 4, 2021

Contents

Chapter 1

Done

1.1 STAR-CCM+

1. **Website.** https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html
2. **Documentation.**
   - Siemens Digital Industries Software. *Simcenter STAR-CCM+ Release Notes*. Version 2020.2 (70 pages).
     → I have read some important features of STAR-CCM+ compared to its competitors.
   - Siemens Digital Industries Software. *Simcenter STAR-CCM+ Installation Guide*. Version 2020.2 (105 pages).
     → I have not read it yet, and unintended to, since Mr. Henrik Pletat has handled the installation step for me.
   - Simcenter STAR-CCM+ Documentation Version 2020.2 (i.e., User Guide 15.04) (12512 pages).
     → I have read the following parts: **Product Overview**, **User Guide**, some parts of **Theory**, and successfully run all simulations in **Tutorials** for *incompressible flow*.
3. **Installed versions.**
   - Notebook `nguyen@fg8nb1`: Installed versions and their installation directories:
     - `13.04.010-R8`: `/opt/CD-adapco/13.04.010-R8/STAR-CCM+13.04.010-R8/star/bin/starccm+`
     - `15.04.008-R8`: `/opt/Siemens/15.04.008-R8/STAR-CCM+15.04.008-R8/star/bin/starccm+`

     Remark 1.1.1. *Before running on the notebook* ***nguyen@fg8nb1****, especially after rebooting, I need to run*

     ```
     nguyen@fg8nb1:~$ flexlm
     ```

     1st to activate the license manager in order to provide a license(s) to run STAR-CCM+.
   - Compute server `escher-01`[1]: Installed versions and their installation directories:
     - `11.06.011-R8`: `/opt/CD-adapco/11.06.011-R8/STAR-CCM+11.06.011-R8/star/bin/starccm+`
     - `13.04.010-R8`: `/opt/CD-adapco/13.04.010-R8/STAR-CCM+13.04.010-R8/star/bin/starccm+`
4. **Release running/floating STAR-CCM+ Licenses on e.g., `escher-01`.** 3 ways:
   - Check if the process(es) is/(are) active with 1 of the following commands[2]:
     - `ps -A | grep ccm` or `ps -e | grep ccm`, used by Caroline Löbhard, usage: to see every process on the system involving `ccm` using standard syntax.
     - `ps aux | grep ccm`, used by Henrik Pletat, usage: to see every process involving `ccm` on the system using BSD syntax.

     After obtaining the process number(s) of STAR-CCM+, kill the running/floating process(es) by the command:

     ```
     kill <STAR-CCM+ process number(s)>
     ```

   - `killall starccm+` → easiest way, especially after multi-thread running STAR-CCM+.

[1] I have also the access to the compute servers `escher-03, escher-04, leonhard-01`, but none of them has STAR-CCM+ installed.
[2] The 2nd one is identical with the 1st one. The 3rd one is

- Use the licence server `FLEXlm`
  - List all the floating licence(s) by

    ```
    nguyen@escher-01:/opt/CD-adapco/11.06.011-R8/FLEXlm/11_13_0_0/bin> ./lmutil lmstat -a
    ```

    In the terminal output, look for e.g.,

    ```
    nguyen escher-01 /dev/tty (v2016.10) (agnesi.wias-berlin.de/11744 503), start Mon 8/27 17:24
    nguyen escher-01 /dev/tty (v2016.10) (agnesi.wias-berlin.de/11744 402), start Mon 8/27 17:24
    nguyen escher-01 /dev/tty (v2016.10) (agnesi.wias-berlin.de/11744 302), start Mon 8/27 17:24
    nguyen escher-01 /dev/tty (v2016.10) (agnesi.wias-berlin.de/11744 202), start Mon 8/27 17:24
    nguyen escher-01 /dev/tty (v2016.10) (agnesi.wias-berlin.de/11744 104), start Mon 8/27 17:25
    ```

    to obtain the number(s) of floating license, here, e.g., `503, 402, 302, 202, 104`[3].
  - Then use `FLEXlm` again to kill these by executing in the directory

    ```
    nguyen@escher-01:/opt/CD-adapco/11.06.011-R8/FLEXlm/11_13_0_0/bin>
    ```

    the following corresponding commands:

    ```
    nguyen@escher-01:/opt/.../bin> ./lmutil lmremove -h ccmpsuite agnesi.wias-berlin.de 11744 503
    nguyen@escher-01:/opt/.../bin> ./lmutil lmremove -h ccmpsuite agnesi.wias-berlin.de 11744 402
    nguyen@escher-01:/opt/.../bin> ./lmutil lmremove -h ccmpsuite agnesi.wias-berlin.de 11744 302
    nguyen@escher-01:/opt/.../bin> ./lmutil lmremove -h ccmpsuite agnesi.wias-berlin.de 11744 202
    nguyen@escher-01:/opt/.../bin> ./lmutil lmremove -h ccmpsuite agnesi.wias-berlin.de 11744 104
    ```

5. **Java macros.** "Simcenter STAR-CCM+ uses the tutorial macros to construct the same model that the tutorial instructions describe."

   → I am familiar to playing the Java macros[4] in STAR-CCM+ (same for all versions).

6. `TheStevePortal/Documentation/`**Tutorials.** I have run successfully in Sect. **Incompressible Flow**:

   (a) Steady Flow: Lid-Driven Cavity Flow

   (b) Steady Flow: Channel Flow with Multiple Meshes

   (c) Steady Flow: Laminar and Turbulent Flow in an S-Bend

   (d) Steady Flow: Backward Facing Step

   (e) Adjoint Flow Solver: External Flow over a Dual Element Wing

   (f) Adjoint Flow Solver: Wing Shape Optimization

   (g) Simulation Operations: S-Bend Shape Optimization

   Numerical results. Optimized S-bend shape with the deformation field illustrated:

[3] They will change each time you execute STAR-CCM+, but they are usually be 3-digits numbers of the form $\overline{10*}, \overline{20*}, \overline{30*}, \overline{40*}, \overline{50*}$ where the last digit $* \in \{1, 2, 3, 4, 5, 6\}$ for the case of 5 floating, but 6 in total (1 license for notebook `nguyen@fg8nb1`), licenses here.

[4] In the group `SOFTube` of OpenFOAM-based applications which will be addressed later, Martin Kanitsar used 12 Java macros:

(a) For Topology Optimization `topoOpt`: `shapeGradientCCM`: the Java macros `ccmReplace.java, mbmw3host.java, mbmw3.java, mbmw3ds.java, mbmw3re.java, mbmw3rs.java, porous.java, porous0.java, porous1.java, wrapGeom.java` are used to interact with STAR-CCM+.

(b) For Shape Optimization `shapeGradientCCM`: the Java macros `ccmReplace.java, mbmw3host.java, mbmw3.java, mbmw3ds.java, mbmw3re.java, mbmw3rs.java, solvePrimal.java, solveAdjoint.java` are used to interact with STAR-CCM+.

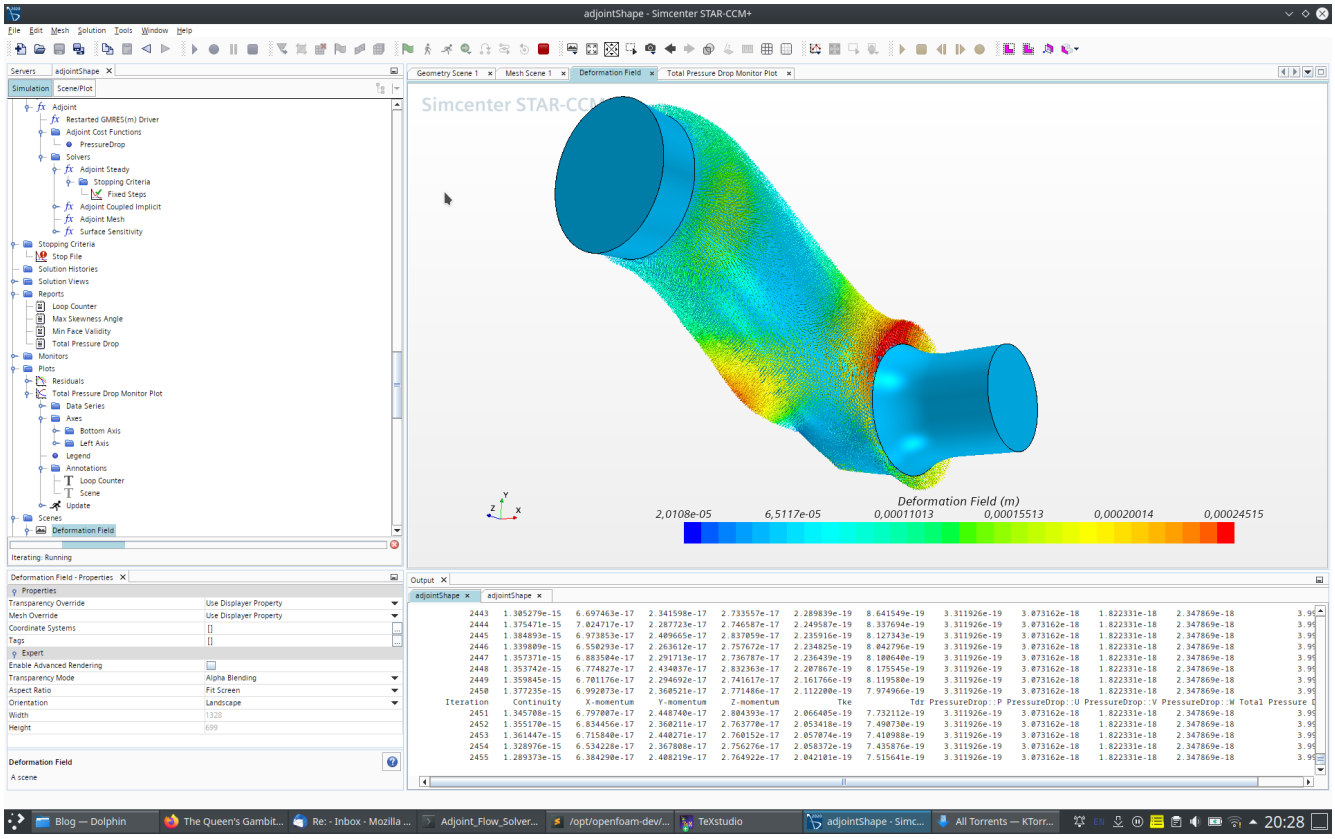

Figure 1.1: Optimized S-bend shape with the deformation field illustrated.

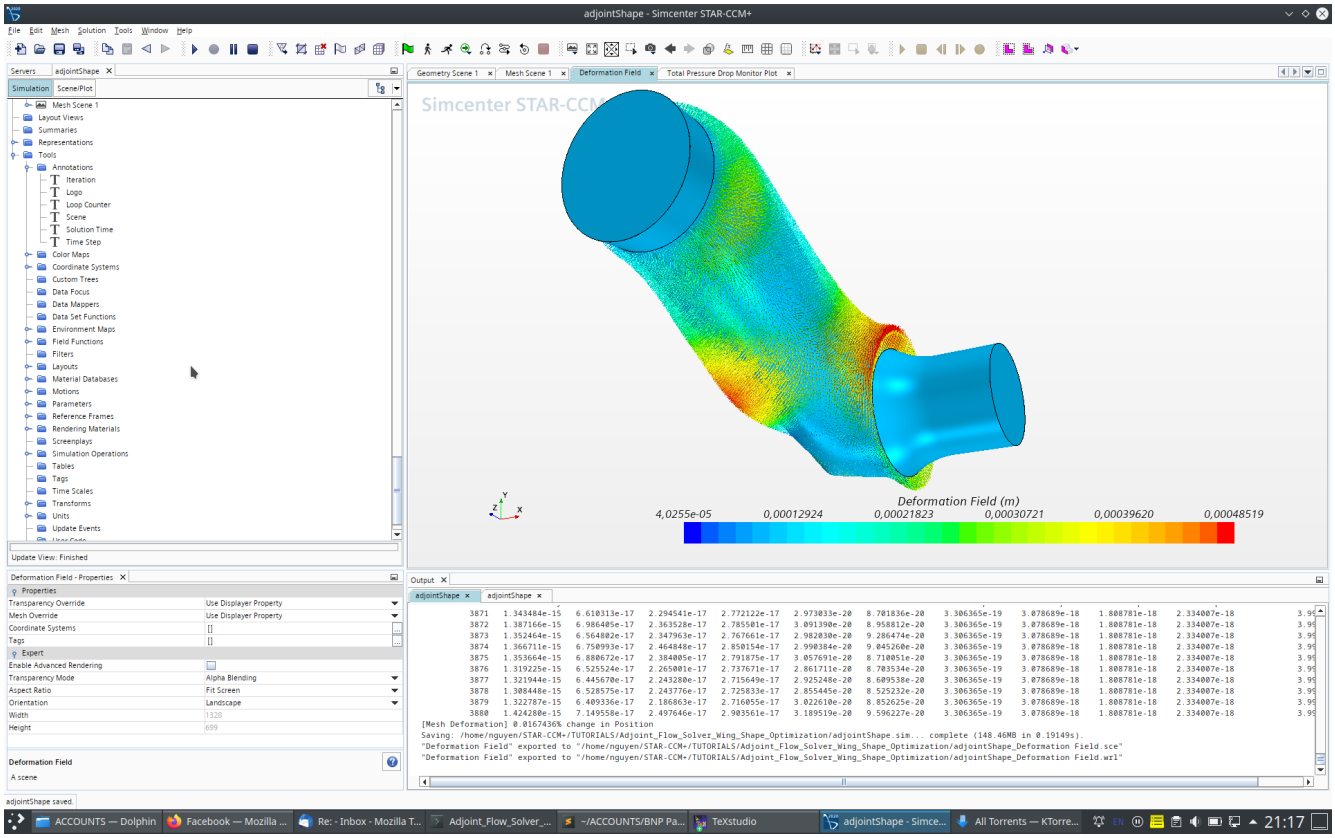

Figure 1.2: Optimized S-bend shape with the deformation field illustrated.

7. **Materials.** I have downloaded some heavy materials from the website `TheStevePortal/Support`. Besides `STAR-CCM+ Tutorial Files 15.04` and `STAR-CCM+ Verification Suite 15.04` for **Tutorials** mentioned above, these materials contains the Java macros `*.java` and predefined simulation files `*.sim` for simulation in: DIESEL, GASOLINE, NATURALGAS, NORA.

   → It seems to me these materials are unnecessary because of their irrelevant themes and we will eliminate STAR-CCM+ at the end anyway. Thus, I ignored all these materials.

1.2 OpenFOAM

1. **Variants.** In Wikipedia/OpenFOAM: "There are 3 main variants of OpenFOAM software that are released as free and open-source software under the GNU General Public License Version 3."

   (a) "OpenFOAM variant by OpenCFD Ltd. (with the name trademarked since 2007) first released as open-source in 2004. (Note that since 2012, OpenCFD Ltd is an affiliate of ESI Group.)" → or, in short, openfoam.com

   (b) "OpenFOAM Foundation Inc. variant, released by The OpenFOAM Foundation Inc. (since 2012), and transferred in 2015 to the English company The OpenFOAM Foundation Ltd." → or, in short, openfoam.org

   (c) "FOAM-Extend variant by Wikki Ltd. (since 2009)" → or, in short, `foam-extended`.

   Remark 1.2.1. *In my opinion, there is no need to pay much attention to this variant at the moment, due to the "unofficial" label of this variant and also the target of our project.*

Remark 1.2.2. *I have installed and currently used the following versions of OpenFOAM:*

- *Notebook* ***nguyen@fg8nb1:***
  - *For openfoam.com variant:* ***OpenFOAM-v2006*** *(release date: Jun 2020). This is not the latest version of this variant, since* ***OpenFOAM-v2012*** *(release date: Dec 23, 2020) released recently.*
  - *For openfoam.org variant:* ***openfoam8*** *(latest stable version), and* ***openfoam-dev*** *(cutting-edge development, updated weekly by* ***sudo apt-get update*** *in (K)Ubuntu).*
- *Compute server* ***escher-01:***
  - ***OpenFOAM-2.1.1*** → *Martin Kanitsar's* ***SOFTube*** *is only compatible with this version.*
  - ***OpenFOAM-v2006.***

    Remark 1.2.3. *Due to old version of* ***gcc, make,*** *and* ***cmake*** *on* ***escher-01,***

```
nguyen@escher-01:~> gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib64/gcc/x86_64-suse-linux/4.8/lto-wrapper
Target: x86_64-suse-linux
...
Thread model: posix
gcc version 4.8.5 (SUSE Linux)

nguyen@escher-01:~> make -v
GNU Make 4.0
Built for x86_64-unknown-linux-gnu
Copyright (C) 1988-2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

nguyen@escher-01:~> cmake --version
     cmake version 3.5.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

    I cannot install ***openfoam-dev*** *on* ***escher-01:*** *it requires the newer version of* ***gcc, make,*** *and* ***cmake.***
- *Compute server* ***escher-03:***
  - ***OpenFOAM-v2006.***

  - *openfoam-dev.*

- *Compute server* `leonhard-01`*:*
  - *OpenFOAM-v2006.*
  - *openfoam-dev.*

  Meanwhile `openfoam-dev` *can be installed on other more powerful compute server, only* `escher-01` *has* STAR-CCM+ *(versions 11 and 13) installed. Thus, if I eliminate* STAR-CCM+ *successfully, I can run the simulation on* `escher-03` *or* `leonhard-01` *to faster runtime.*

Remark 1.2.4. *Provided these versions are already compiled properly, I can switch easily between these versions of OpenFOAM by activating/uncommenting the correct line and commenting lines for other versions in the* `~/.bashrc` *file, e.g.,*

```
. /opt/openfoam-dev/etc/bashrc
# . /opt/openfoam8/etc/bashrc
# source $HOME/OpenFOAM/OpenFOAM-2.1.1/etc/bashrc
# source $HOME/OpenFOAM/OpenFOAM-v2006/etc/bashrc
# source $HOME/OpenFOAM/OpenFOAM-dev/etc/bashrc
```

and source them in the current terminal

```
source ~/.bashrc
```

or you just need to open a new terminal and the `~/.bashrc` *will be sourced automatically by the nature of Unix/Linux Operating Systems.*

2. **References.**

   (a) For openfoam.com:
   - User Guide version v2012 (release date: Dec 22, 2020): [pdf][online].
   - Tutorial Guide version v2012 (release date: Dec 22, 2020): [pdf][online].
   - Extended Code Guide v2006: [online].
   - Programmer's Guide v2012: [pdf].
   - Tutorial Wiki: A community-driven collection of tutorials hosted at `https://wiki.openfoam.com/Main_Page`.

   (b) For openfoam.org:
   - User Guide version 8 (release: Jul 22, 2020) [pdf][online]
   - C++ Source Code Guide for OpenFOAM-dev: mainly used as a search tool/code-reference.
   - CFD Direct/OpenFOAM Documentation
     - CFD Direct/OpenFOAM Linux Guide
     - CFD Direct/Tensor Mathematics
     - CFD Direct/Energy Equation in OpenFOAM
       → added to my reading list, but I have not read yet. It's quite short though.
     - CFD Direct/Computational Fluid Dynamics
       → added to my reading list, but I have not read yet. It's quite short though.
     - CFD Direct/Productive CFD with OpenFOAM

   (c) For `foam-extended`:
   - Unofficial OpenFOAM Wiki.
   - `foam-extended` - the community edition.
   - **GitHub repository**: Unofficial-Extend-Project-Mirror.

Remark 1.2.5. *I have read almost all of these references, except C++ Source Code Guides, to see the big picture of OpenFOAM and which features are available. But I need time to consider these available features on theoretical side and dive deeper in C++ source codes to know how to practically make them fit to our software.*

3. **Solvers.** Based on the list of OpenFOAM standard solvers,

   - **Incompressible flow.**
     - `adjointOptimisationFoam, adjointShapeOptimizationFoam`: These optimization applications are based on the series of papers of Carsten Othmer and the Greek Shape/Topology Optimization community (engineering-oriented).
     - `icoFoam`: "Transient solver for incompressible, laminar flow of Newtonian fluids".
     - `pimpleFoam`: "Transient solver for incompressible, turbulent flow of Newtonian fluids on a moving mesh".
     - `pisoFoam`: "Transient solver for incompressible, turbulent flow, using the PISO algorithm".
     - `simpleFoam`: "Steady-state solver for incompressible, turbulent flows".

     Remark 1.2.6. *I am familiar with these incompressible solver (e.g., adjust parameters, set up initial and boundary conditions types, generate mesh by `blockMesh`) and run it separately on some simple cases. But I have not coupled, and unintended to at the moment, these solvers with Martin Kanitsar's `SOFTube/shapeGradientCCM` yet since his solver is more "C++-oriented" than properly "OpenFOAM-oriented". The folder structure is improper as forced in OpenFOAM rules. And the real trouble for me is OpenFOAM uses FVM, meanwhile a lot of components of Martin Kanitsar's `SOFTube/shapeGradientCCM` used FEM instead. There will be a big compatibility issue if I continue to develop Martin Kanitsar's `SOFTube`.*

   - **Compressible flow.** → I have not touched yet. But it seems unavoidable when I come to the part of solving adjoint equations numerically.

4. **Turbulence models.** A full list of available turbulence models in OpenFOAM:

   https://cfd.direct/openfoam/user-guide/v8-turbulence/.

   Remark 1.2.7. *At the moment, I only focus on k-ϵ and Smagorinsky turbulence models.*

5. **Boundary conditions.** A full list of available boundary conditions in OpenFOAM:

   https://cfd.direct/openfoam/user-guide/v8-boundaries/.

   Remark 1.2.8. *I need to work on theoretical side to derive their corresponding adjoint boundary conditions. And if the boundary layers and/or wall laws are involved, more investigations are needed.*

1.3 FEniCS

1. **Variants.** FEniCS, FEniCS-X (from 2021, renew almost all).
2. **Website.** https://fenicsproject.org/.
3. **GitHub repository.** https://github.com/FEniCS.
4. **BitBucket repository.** https://bitbucket.org/fenics-project/.
5. **Intro.**
   - "FEniCS is a popular open-source (LGPLv3) computing platform for solving PDEs.
   - FEniCS enables users to quickly translate scientific models into efficient finite element code.
   - With the high-level Python and C++ interfaces to FEniCS, it is easy to get started, but FEniCS offers also powerful capabilities for more experienced programmers.
   - FEniCS runs on a multitude of platforms ranging from laptops to high-performance clusters."
6. **Reference.** Dokken, 2020, Langtangen and Logg, 2016.
7. **FEniCS tutorials GitHub repository.** https://github.com/hplgit/fenics-tutorial.
8. **Tutorials.** I have run successfully, with some technical corrections in the Python script the following FEniCS tutorial:

(a) **Poisson's equation.**

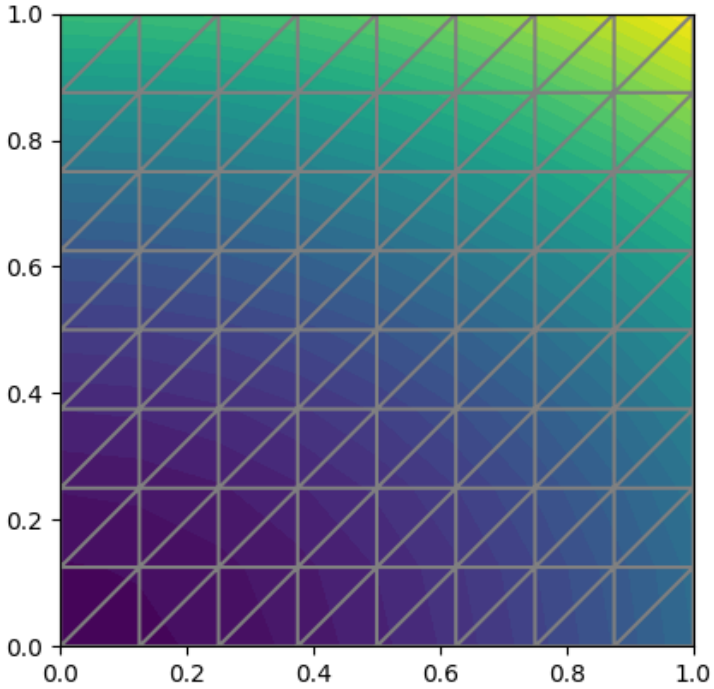

Figure 1.3: Solution of Poisson's equation.

(b) **Poisson's equation for membrane.**

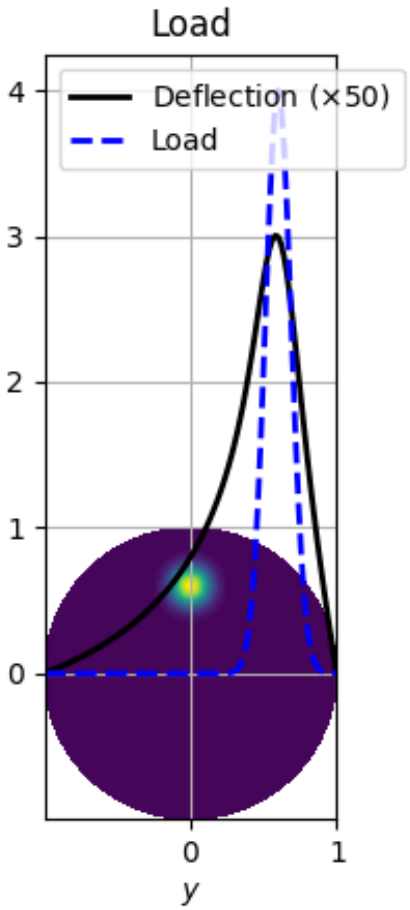

Figure 1.4: Curve for Poisson's equation for membrane.

(c) **Heat equation.**

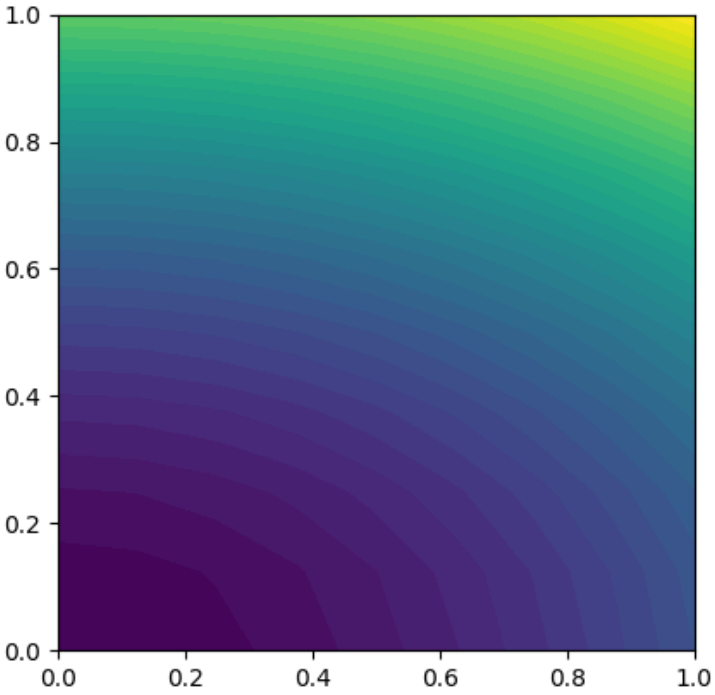

Figure 1.5: Solution of heat equation.

(d) **Heat equation with Gaussian.**

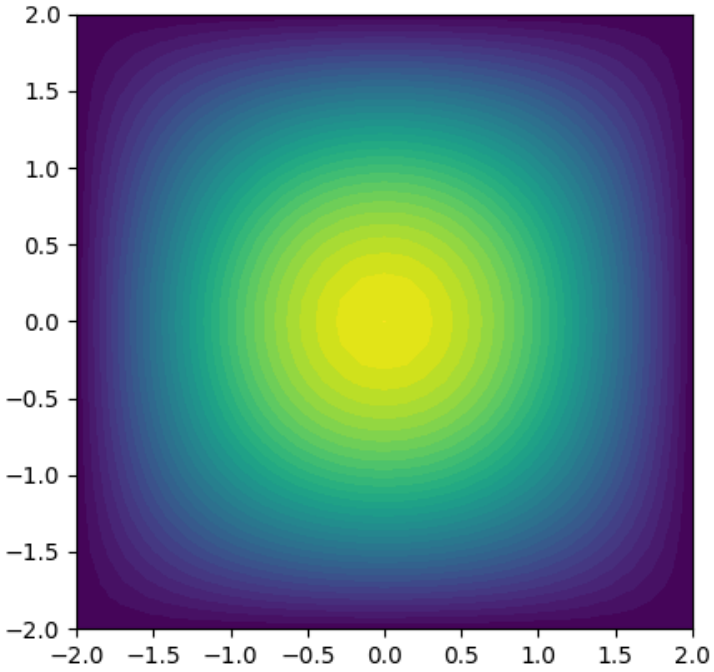

Figure 1.6: Solution of heat equation with Gaussian.

(e) **Nonlinear Poisson's equation.**

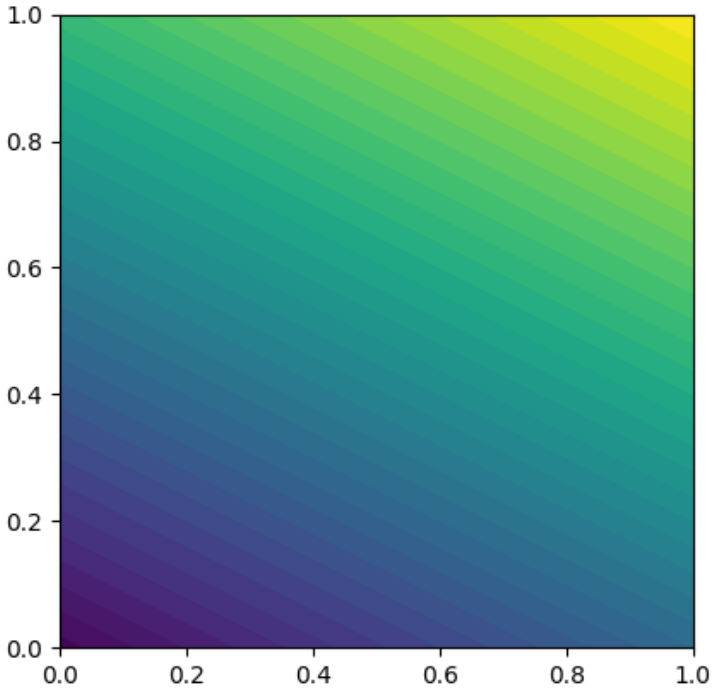

Figure 1.7: Solution of nonlinear Poisson's equation.

(f) **Elasticity.**

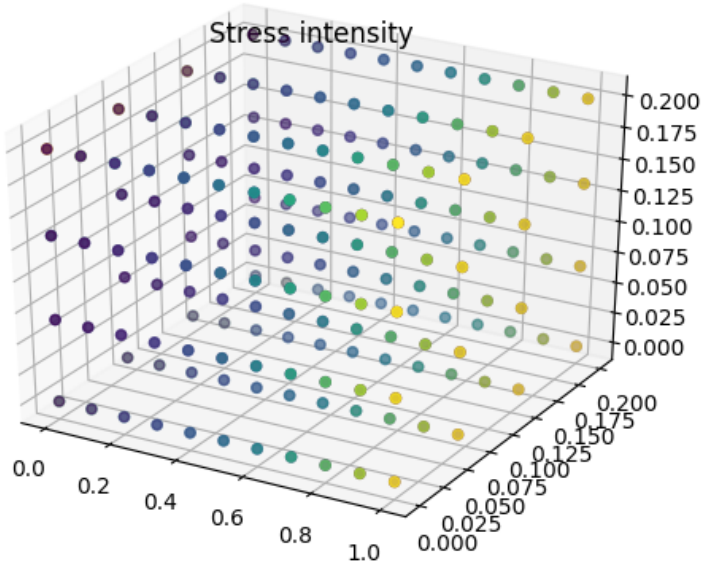

Figure 1.8: Elasticity.

(g) **Navier-Stokes flow in a channel**.

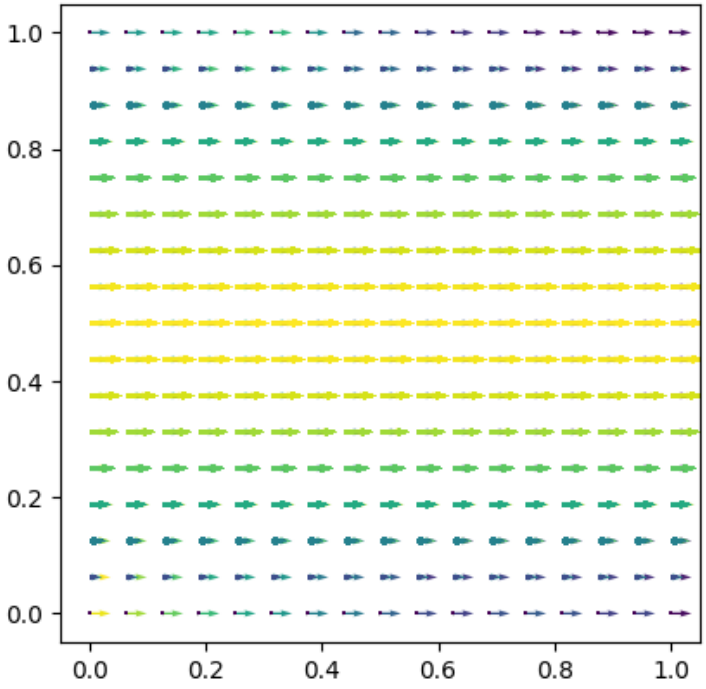

Figure 1.9: Velocity of Navier-Stokes flows in a channel.

(h) **Navier-Stokes flows in a cylinder**.

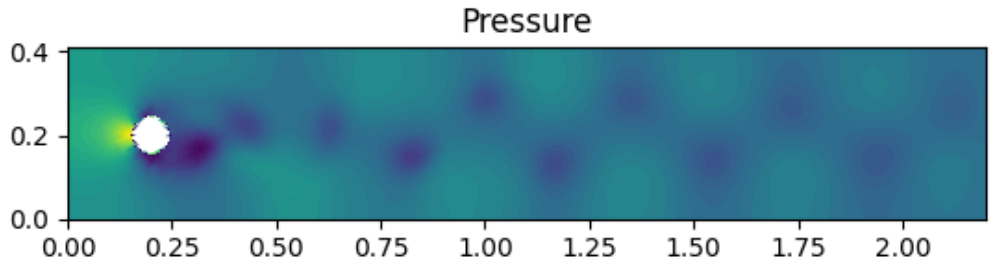

Figure 1.10: Pressure p of Navier-Stokes flows in a cylinder.

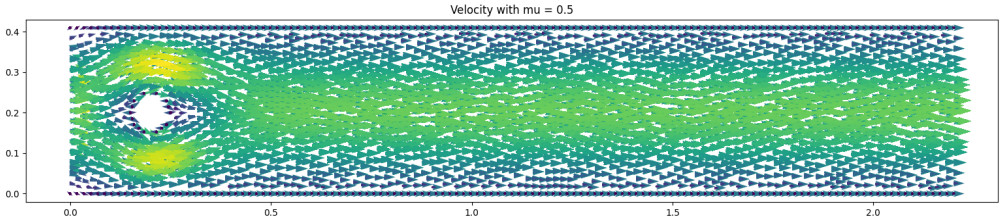

Figure 1.11: Velocity $\mathbf{u}$ of Navier-Stokes flows in a cylinder with viscosity $\mu = 0.5$.

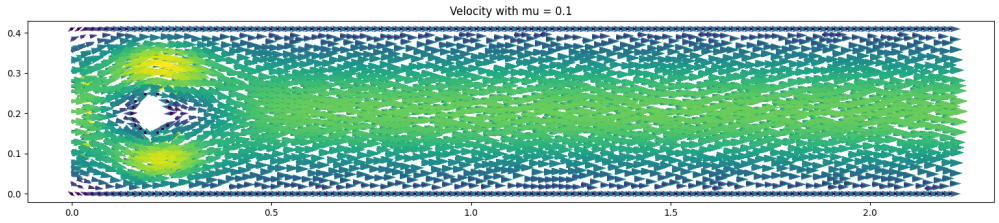

Figure 1.12: Velocity $\mathbf{u}$ of Navier-Stokes flows in a cylinder with viscosity $\mu = 0.1$.

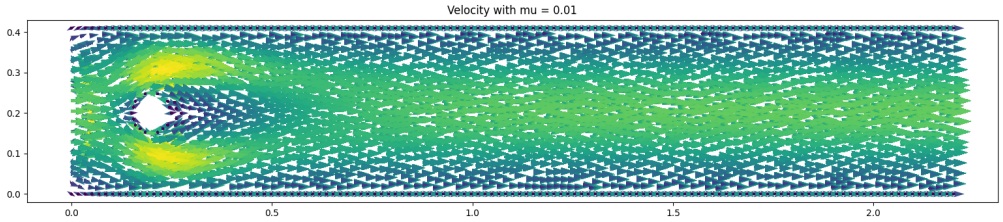

Figure 1.13: Velocity $\mathbf{u}$ of Navier-Stokes flows in a cylinder with viscosity $\mu = 0.01$.

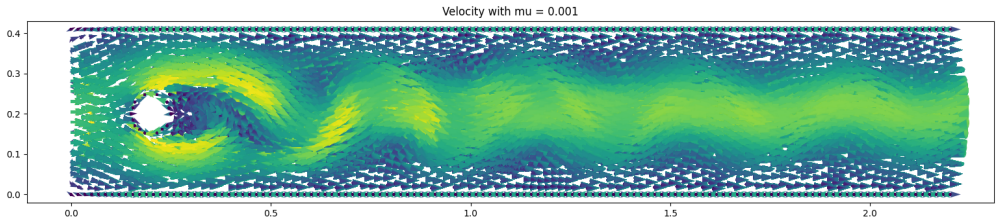

Figure 1.14: Velocity $\mathbf{u}$ of Navier-Stokes flows in a cylinder with viscosity $\mu = 0.001$.

9. **Materials.** Stephan Schmidt's course + codes on *Shape and Geometry*.

1.4 Firedrake

1. **Website.** https://www.firedrakeproject.org/
2. **GitHub repository.** https://github.com/firedrakeproject/firedrake
3. **Intro.** "*Firedrake is an automated system for the solution of PDEs using the FEM. Firedrake uses sophisticated code generation to provide mathematicians, scientists, and engineers with a very high productivity way to create sophisticated high performance simulations.*"
4. **Features.**
   - "Expressive specification of any PDE using the Unified Form Language from the FEniCS Project.
   - Sophisticated, programmable solvers through seamless coupling with PETSc.
   - Triangular, quadrilateral, and tetrahedral unstructured meshes.
   - Layered meshes of triangular wedges or hexahedra.
   - Vast range of finite element spaces.
   - Sophisticated automatic optimization, including sum factorization for high order elements, and vectorization.
   - Geometric multigrid.
   - Customizable operator preconditioners.
   - Support for static condensation, hybridization, and HDG methods."
5. **Reference.** [inserted original article(s)]
6. **Documentation.** https://www.firedrakeproject.org/documentation.html
7. **Introductory tutorials.** I have run successfully:
   - **Simple Helmholtz equation**:
   $$\begin{cases} -\Delta u + u = f & \text{in } \Omega = [0,1]^2, \\ \partial_{\mathbf{n}} u = 0 & \text{on } \Gamma. \end{cases}$$
   - **Burgers equation**: $\nu > 0$ is a constant viscosity,
   $$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} = \mathbf{0} & \text{in } \Omega = [0,1]^2, \\ (\mathbf{n} \cdot \nabla)\mathbf{u} = 0 & \text{on } \Gamma. \end{cases}$$
   - **Mixed formulation for the Poisson equation**: Introduce the negative flux $\boldsymbol{\sigma} = \nabla u$ as an auxiliary vector-valued variable,
   $$\begin{cases} \boldsymbol{\sigma} - \nabla u = 0 & \text{on } \Omega = [0,1]^2, \\ \nabla \cdot \boldsymbol{\sigma} = -f & \text{on } \Omega, \\ u = u_0 & \text{on } \Gamma_{\mathrm{D}}, \\ \boldsymbol{\sigma} \cdot \mathbf{n} = g & \text{on } \Gamma_{\mathrm{N}}. \end{cases}$$
   - **DG advection equation with upwinding**: Advection equation
   $$\begin{cases} \partial_t q + (\mathbf{u} \cdot \nabla) q = 0 & \text{in } [0,T] \times \Omega, \\ q(0, \mathbf{x}) = q_0(\mathbf{x}) & \text{in } \Omega, \\ q(t, \mathbf{x}) = q_{\text{in}}(t, \mathbf{x}) & \text{on } \Gamma_{\text{inflow}}. \end{cases}$$
   - **Steady-state continuity equation on an extruded mesh**: Use of extruded meshes, including the new regions of integration and the construction of sophisticated FE spaces.
   $$\begin{cases} \nabla \cdot (q\mathbf{u}) = 0 & \text{in } \Omega, \\ q = q_{\text{in}} & \text{on } \Gamma_{\text{inflow}}. \end{cases}$$

- **Double slit experiment**: Solve a linear wave equation using an explicit timestepping scheme. Demonstrate the use of an externally generated mesh, pointwise operations on Functions, and a time varying boundary condition:

$$\begin{cases} \partial_t^2 \phi - \Delta \phi = 0 & \text{in } [0,T] \times \Omega, \\ \partial_{\mathbf{n}} \phi = 0 & \text{on } \Gamma_{\mathrm{N}}, \\ \phi = \dfrac{1}{10\pi} \cos(10\pi t) & \text{on } \Gamma_{\mathrm{D}}. \end{cases}$$

  Make the substitution to facilitate the choice of time integrator:

$$\begin{cases} \partial_t \phi = -p & \text{in } [0,T] \times \Omega, \\ \partial_t p + \Delta \phi = 0 & \text{in } [0,T] \times \Omega, \\ \partial_{\mathbf{n}} \phi = 0 & \text{on } \Gamma_{\mathrm{N}}, \\ p = \sin(10\pi t) & \text{on } \Gamma_{\mathrm{D}}, \end{cases}$$

- **Creating Firedrake-compatible meshes in Gmsh**: Key structure of a `gmsh.geo` file that creates a Firedrake-compatible mesh.

1.5 Fireshape

1. **Website.** https://fireshape.readthedocs.io/en/latest/
2. **GitHub repository.** https://github.com/fireshape/fireshape
3. **Overview.** "Fireshape is a shape optimization toolbox for the finite element library Firedrake."

   Inputs. "To set up a shape optimization problem, all you need to provide is the mesh on an initial guess, the shape functional, and the weak-form of its PDE-contraint."

   "Fireshape computes adjoints and assembles first and second derivatives for you using pyadjoint, and it solves the optimization problem using the rapid optimization library (ROL)."
4. **Features.**
   - "Fireshape neatly distinguishes between the discretization of control and state variables.
     - To discretize the control, you can choose among finite elements, B-splines, and the free-form approach.
     - To discretize the state, you have access to all finite element spaces offered by Firedrake.
   - Fireshape relies on the mesh-deformation approach to update the geometry of the domain.
     By specifying the metric of the control space, you can decide whether meshes should be updated using Laplace or elasticity equations.
     In 2D, you can also use the elasticity equation corrected with Cauchy-Riemann terms, which generally leads to very high-quality meshes."
5. **Documentation.** Paganini and Wechsung, 2020a.
6. **Reference.** Paganini and Wechsung, 2020b.
7. **Installation.** I have installed Fireshape version 1.0 successfully in VirtualBox Ubuntu 18.04.4. I need to do this, because:
   - The version of Python Fireshape requires matches exactly the default version of Python Ubuntu 18.04.4 have.
   - After having installed Fireshape on Ubuntu, FEniCS does not work properly anymore. Because of this conflict, which seems to involve Python versions again, I use VirtualBox Ubuntu instead. But working with a virtual machine slows down considerably the whole simulation process because of, e.g.:
     - switch to VirtualBox,
     - running time of Fireshape with partially (less than half) distributed memory, then switch back to the host machine,
     - shared folder trouble between host and virtual machines.

Remark 1.5.1. *Because I will focus more on FEniCS than Fireshape, I have not tested the compatibility of Fireshape version 2.0 to Ubuntu 20.04 to protect FEniCS installed in my notebook.*

8. **Tutorials.** I have run successfully:

   (a) **Example 1: Level Set**: Minimize the shape functional

$$J(\Omega) := \int_{\Omega} f(\mathbf{x})\mathrm{d}\mathbf{x}, \text{ where } f(x,y) = \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 - \frac{1}{2}.$$

   Optimal solution:

$$\Omega_{\text{opt}} := B_{1/\sqrt{2}}\left(\left(\frac{1}{2}, \frac{1}{2}\right)\right).$$

   Numerical results.

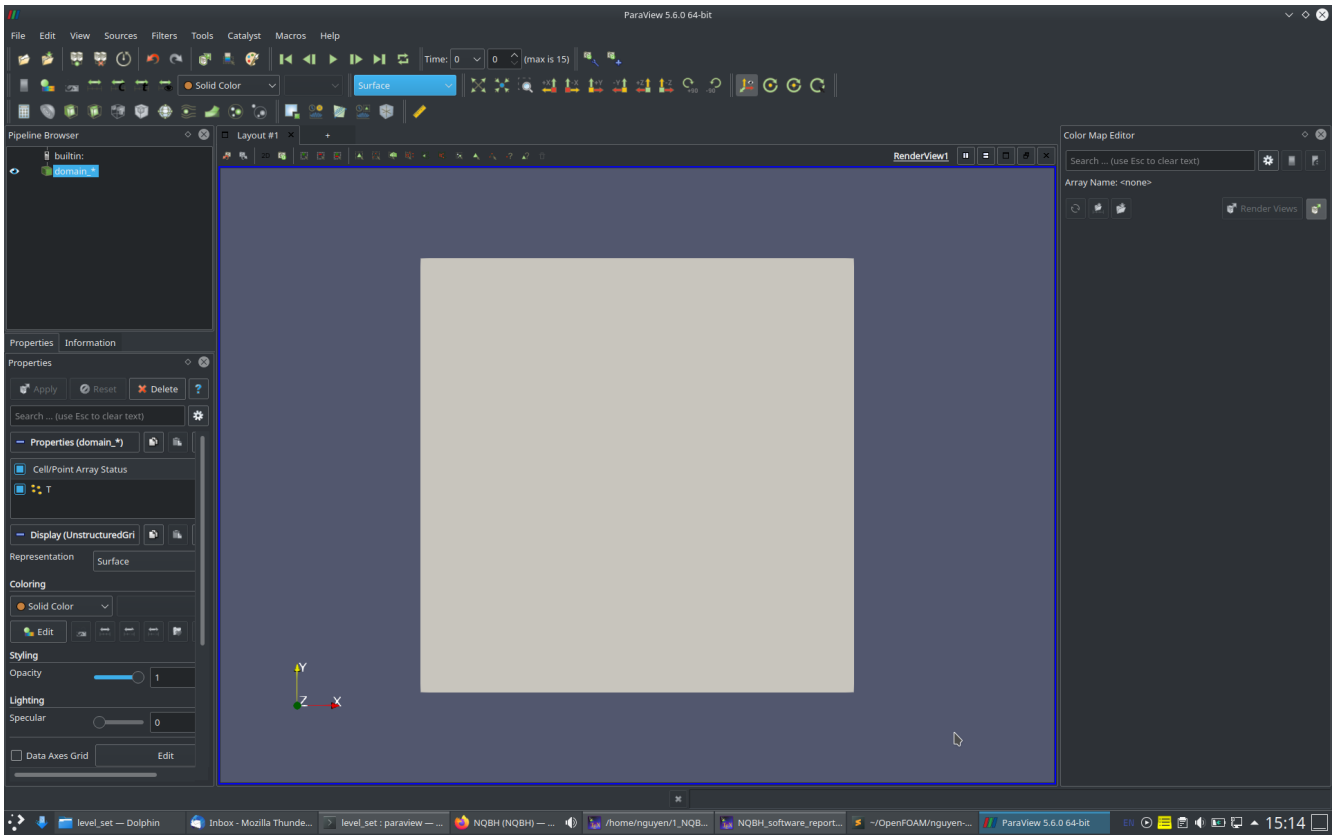

Figure 1.15: View the initial domain by ParaView with modes: `Solid Color, Surface`.

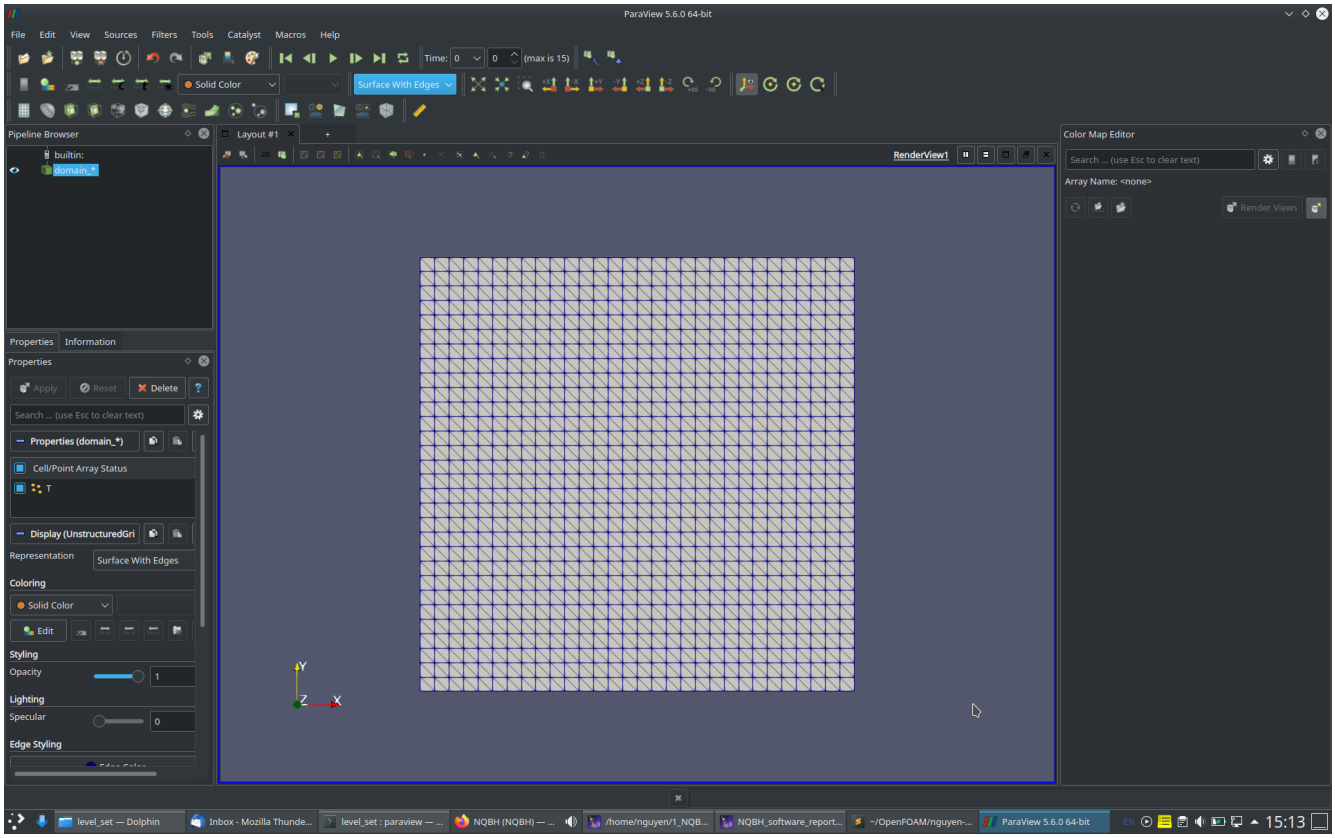

Figure 1.16: View the initial domain by ParaView with modes: `Solid Color, Surface with Edges`.

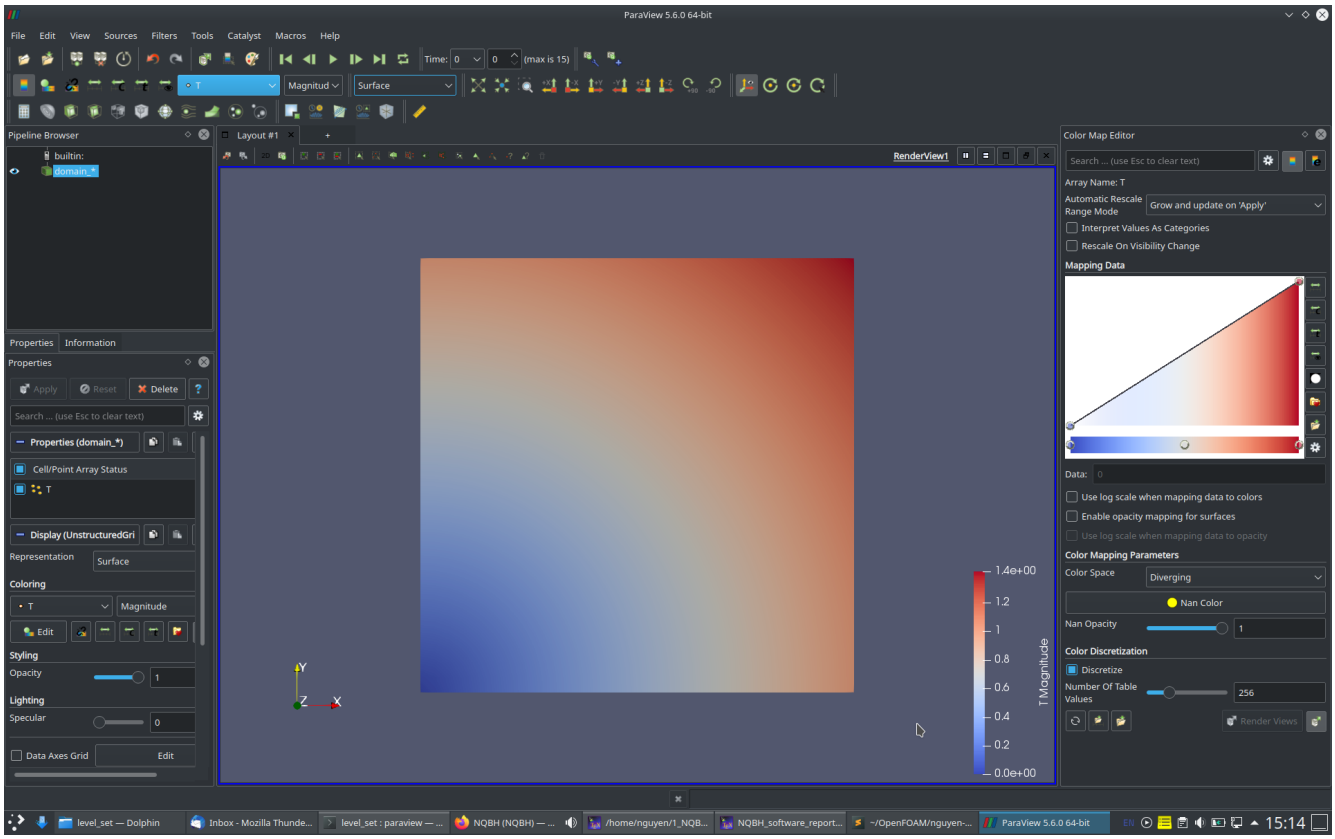

Figure 1.17: View the initial domain by ParaView with modes: T, `Surface`.

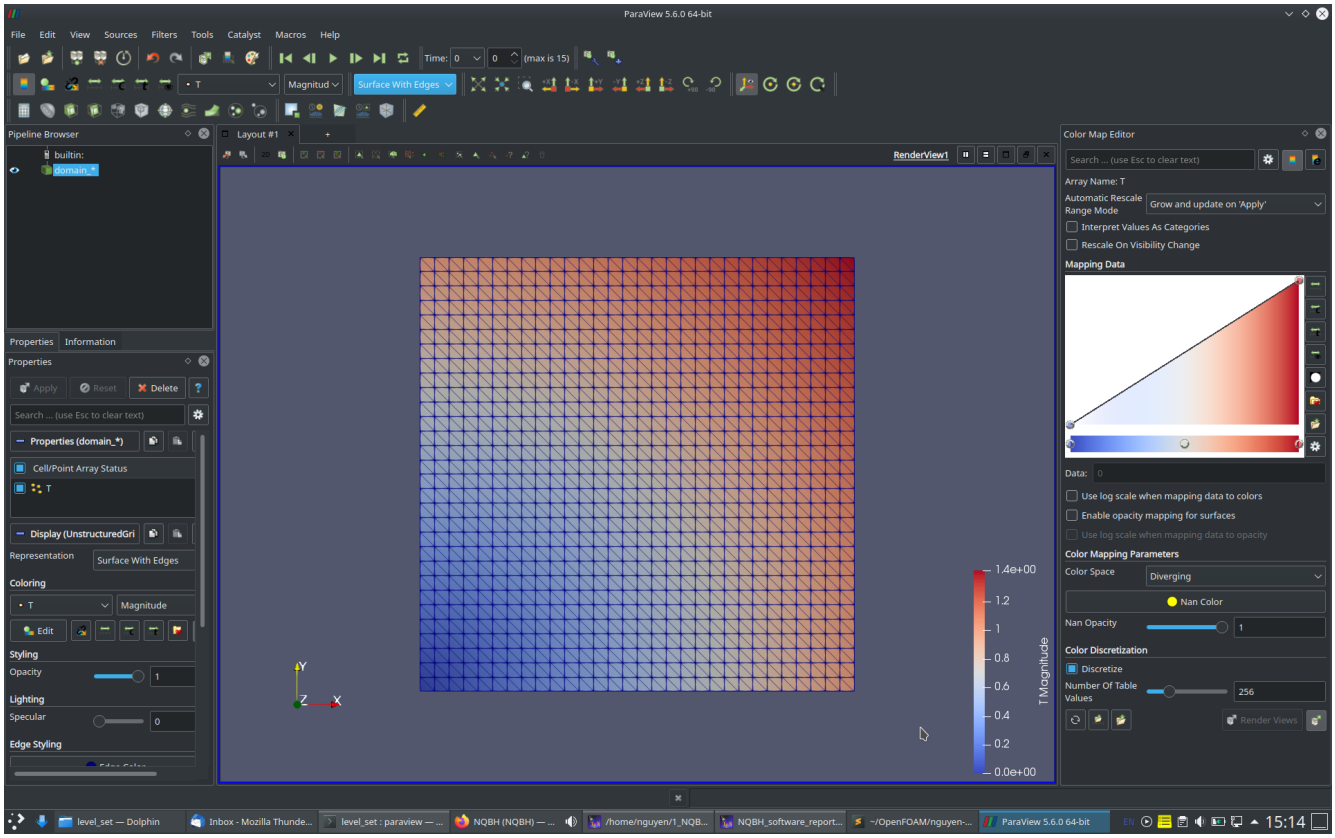

Figure 1.18: View the initial domain by ParaView with modes: T, `Surface with Edges`.

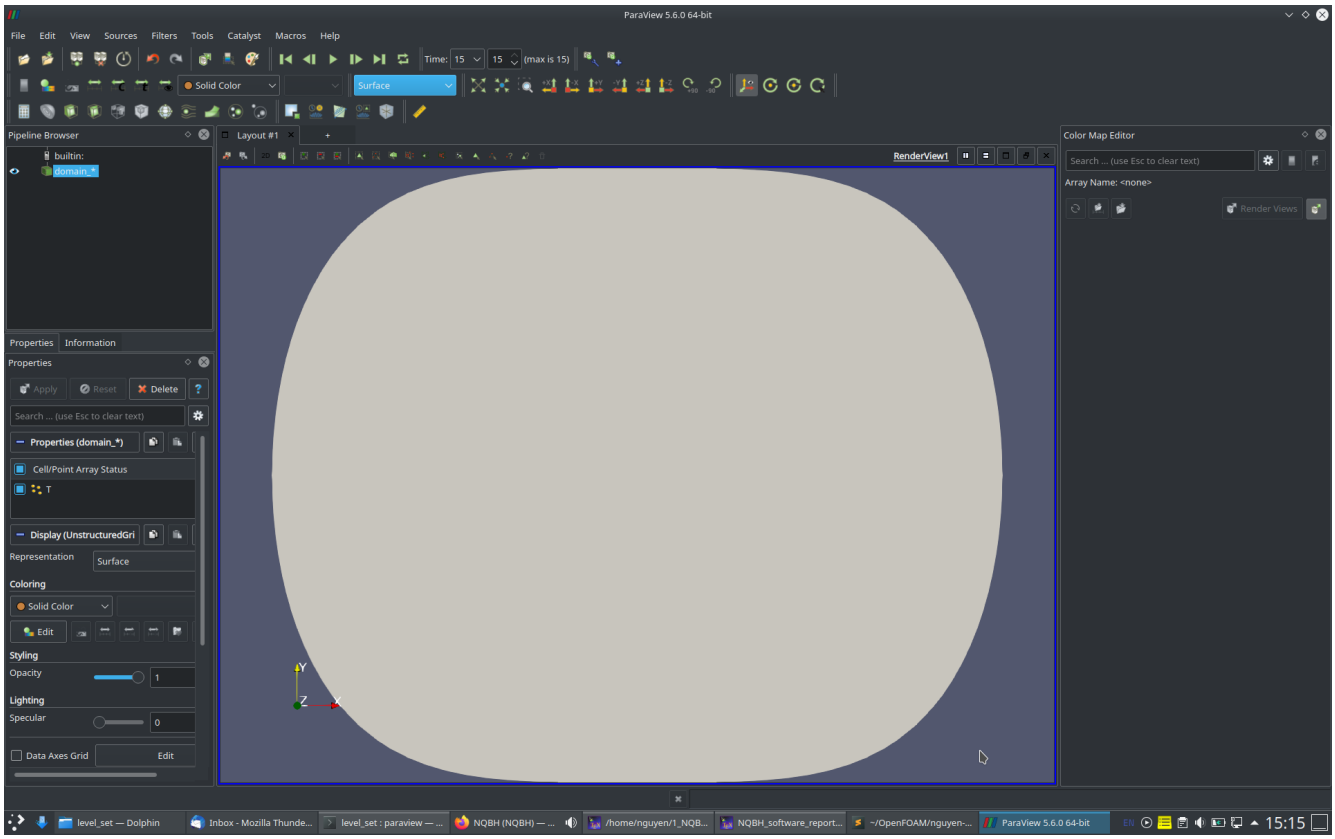

Figure 1.19: View the optimized domain after 15 optimization steps by ParaView with modes: `Solid Color, Surface`.

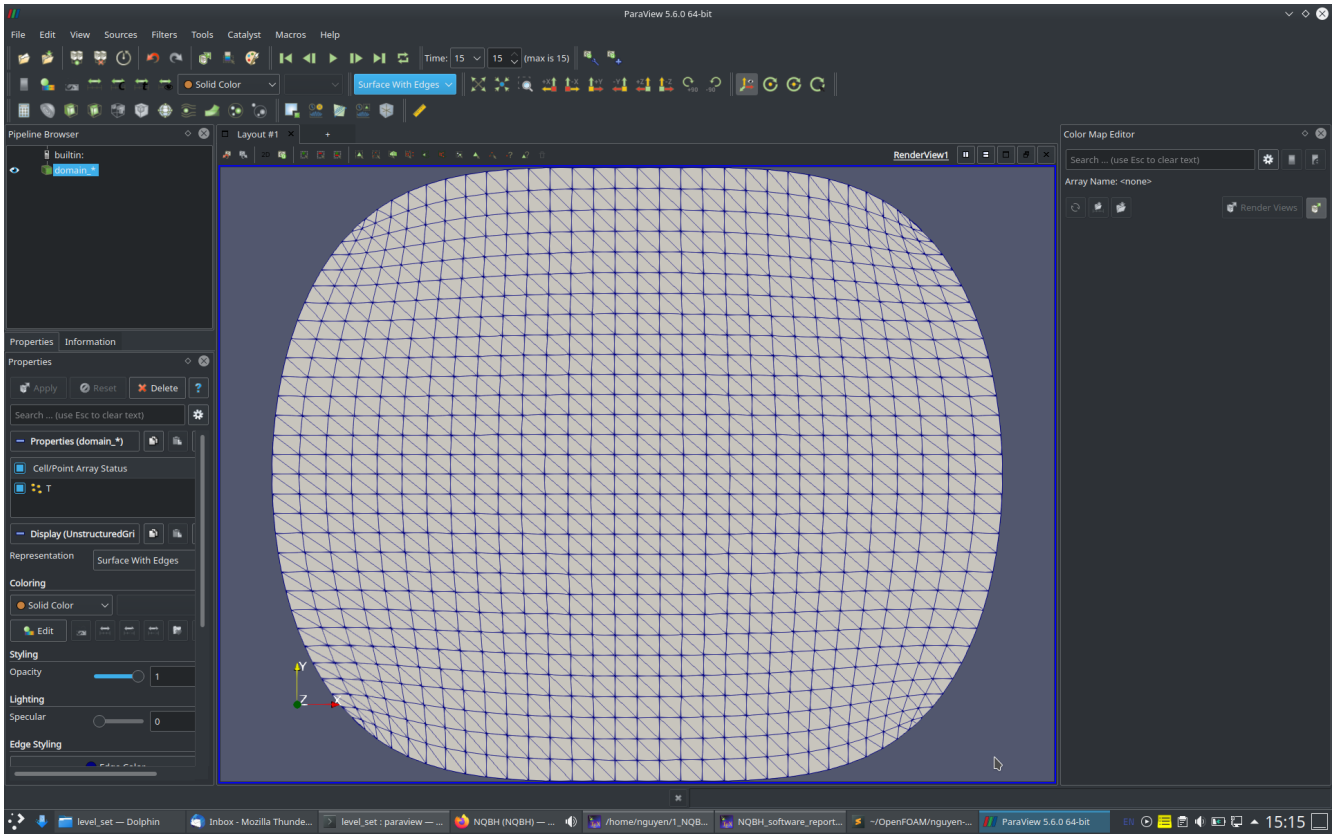

Figure 1.20: View the optimized domain after 15 optimization steps by ParaView with modes: `Solid Color, Surface with Edges`.

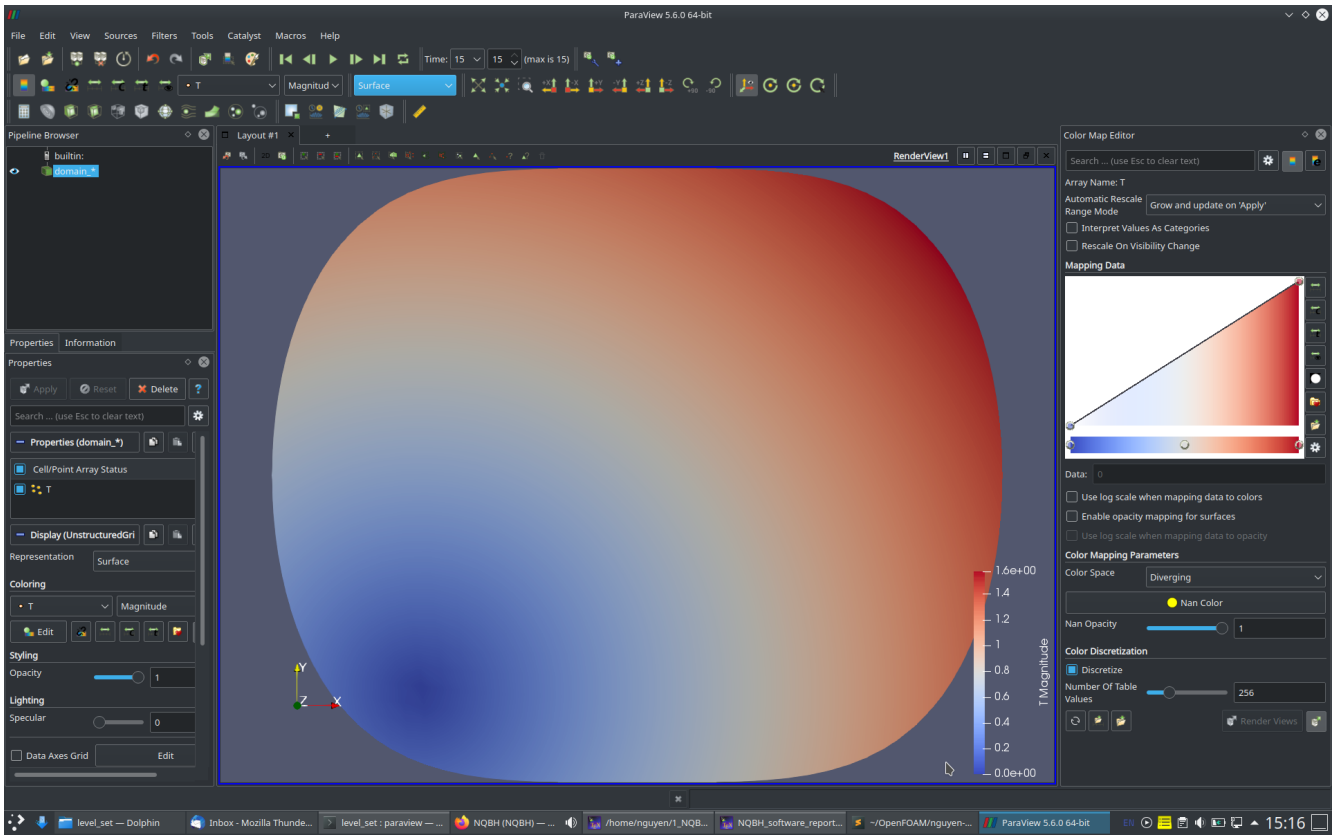

Figure 1.21: View the optimized domain after 15 optimization steps by ParaView with modes: T, `Surface`.

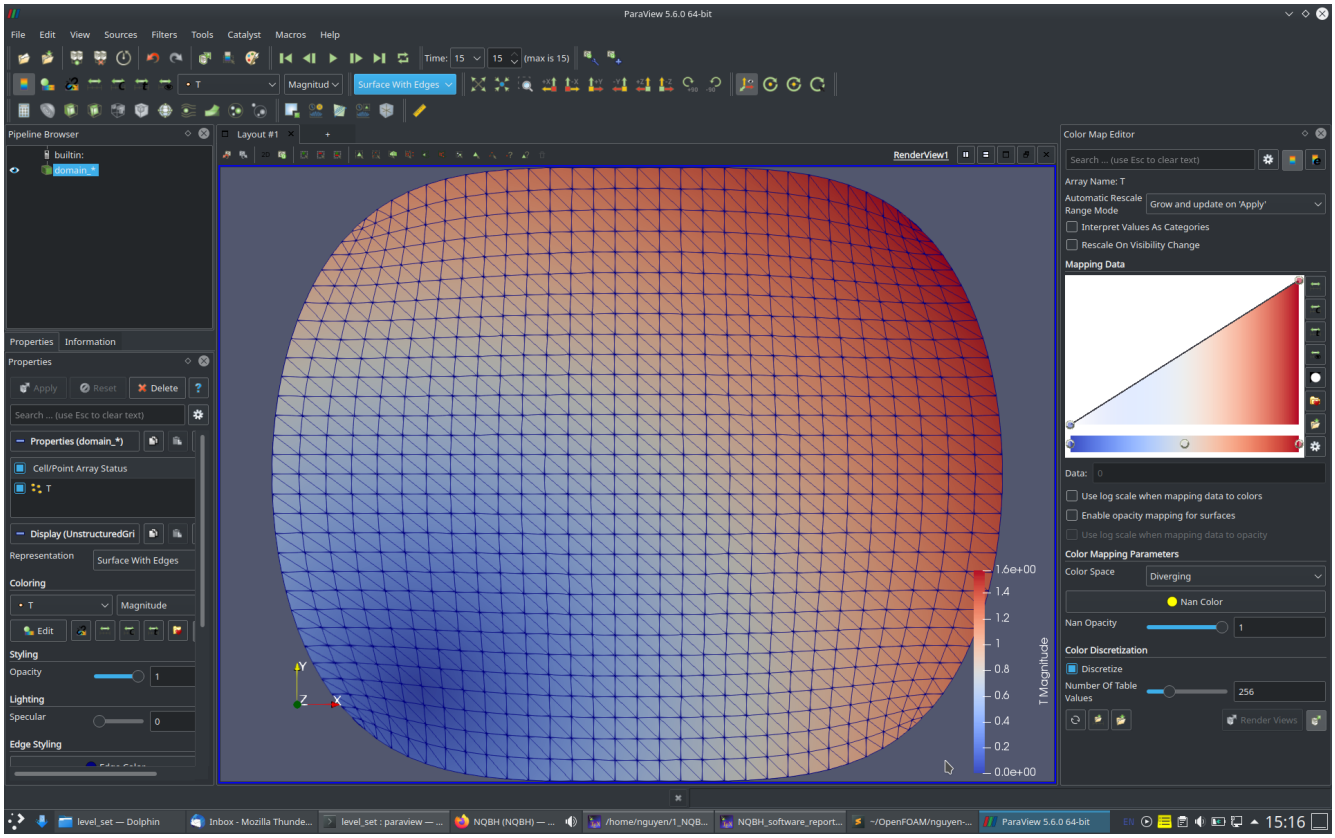

Figure 1.22: View the optimized domain after 15 optimization steps by ParaView with modes: T, `Surface with Edges`.

(b) **Example 2: L^2-tracking**: Minimize the shape functional

$$J(\Omega) := \int_\Omega \left(u_(\mathbf{x}) - u_t(\mathbf{x})\right)^2 \mathrm{d}\mathbf{x}, \text{ where } \begin{cases} -\Delta u = 4 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

and $u_t : \mathbb{R}^2 \to \mathbb{R}$ is a target function:

$$u_t(x, y) = \frac{9}{25} - \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 - \frac{1}{2}.$$

Optimal solution: $\emptyset$ and $\Omega_{\text{opt}} := B_{3/5}((\frac{1}{2}, \frac{1}{2}))$.

Numerical results.

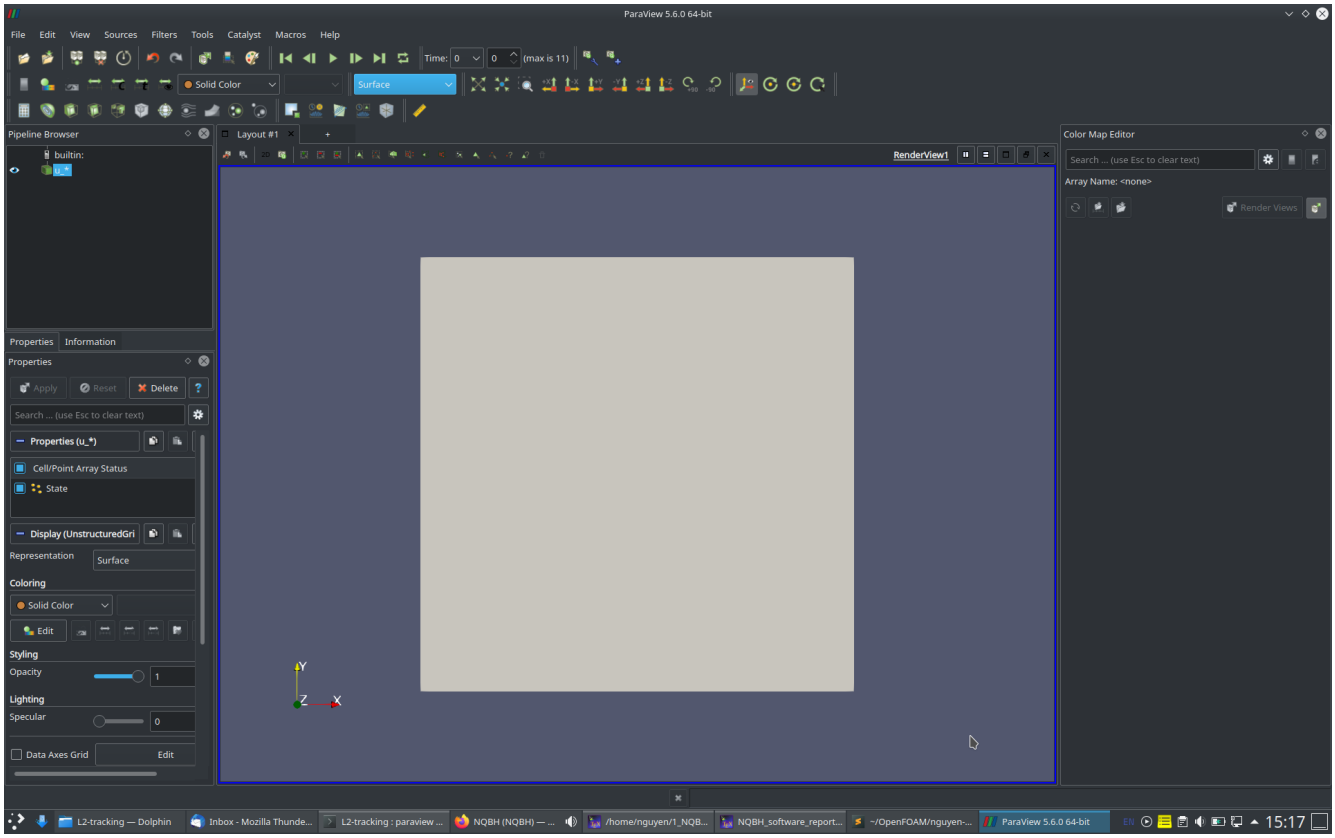

Figure 1.23: View the initial domain by ParaView with modes: `Solid Color, Surface`.

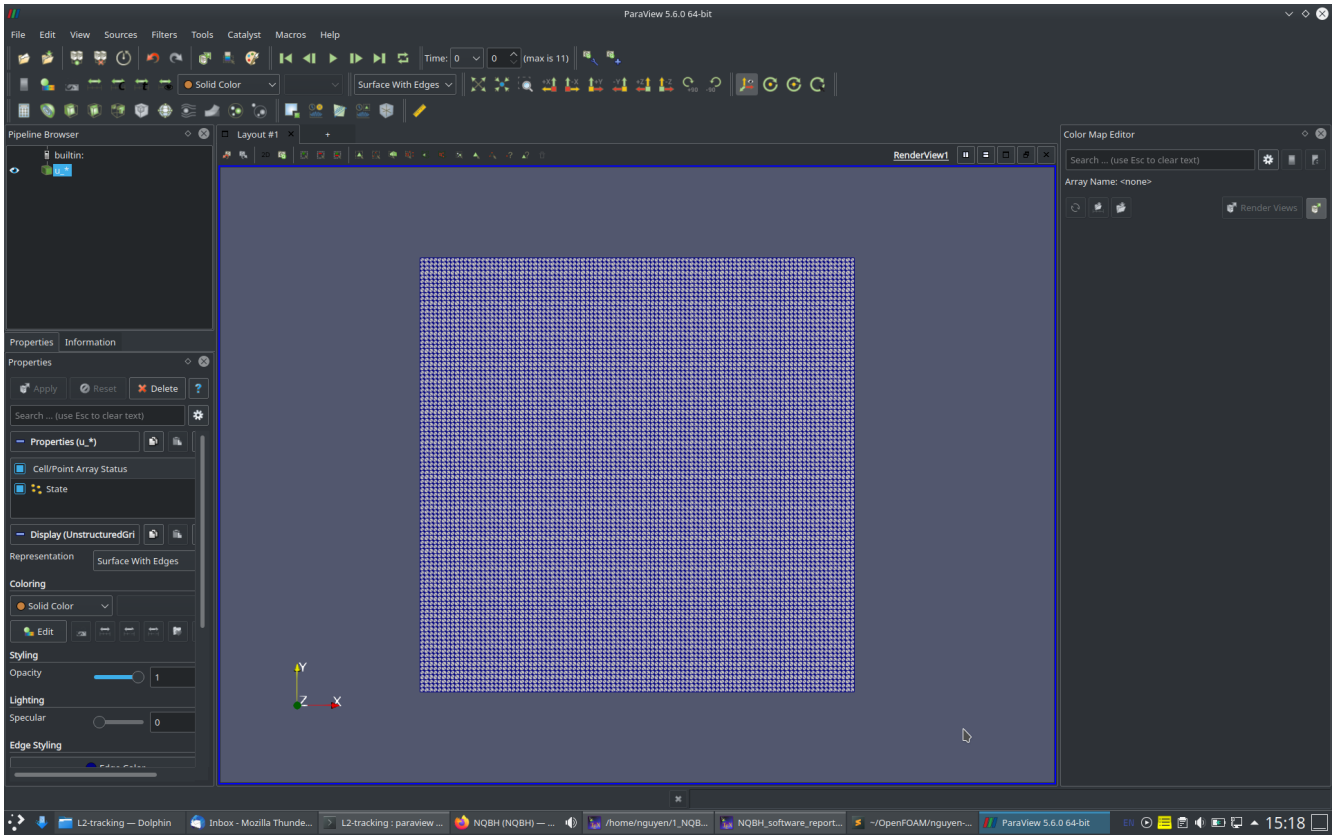

Figure 1.24: View the initial domain by ParaView with modes: `Solid Color, Surface with Edges`.

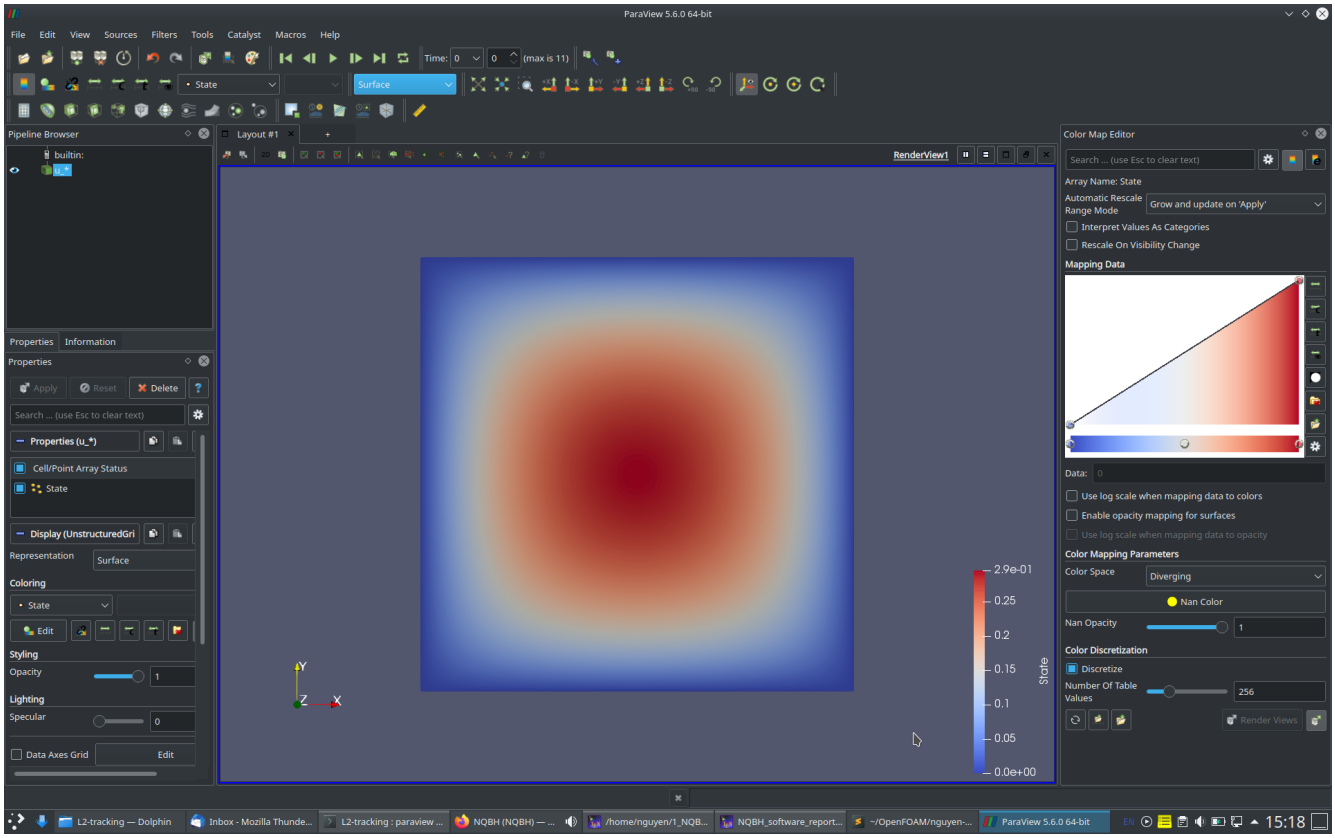

Figure 1.25: View the initial domain by ParaView with modes: `State, Surface`.

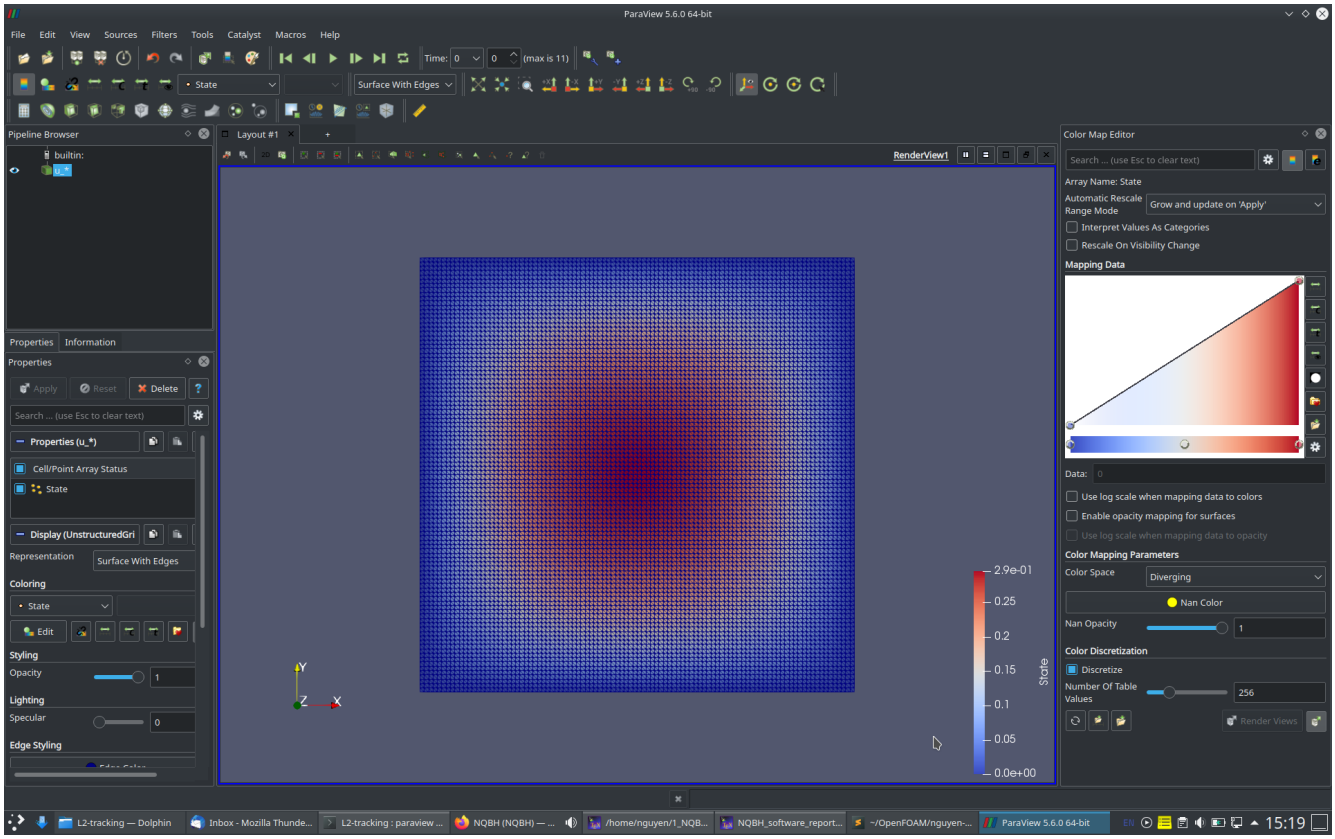

Figure 1.26: View the initial domain by ParaView with modes: `State, Surface with Edges`.

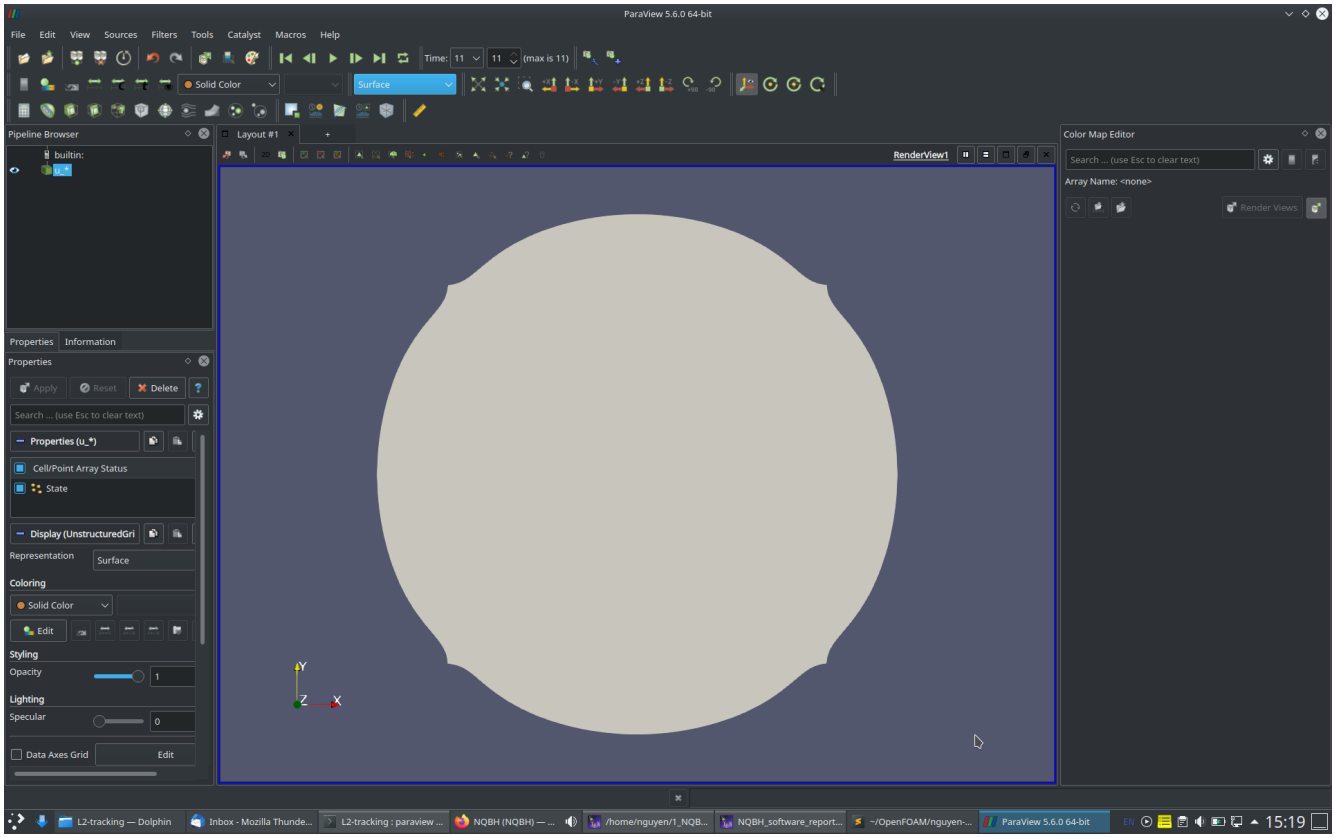

Figure 1.27: View the optimized domain after 11 optimization steps by ParaView with modes: `Solid Color, Surface`.

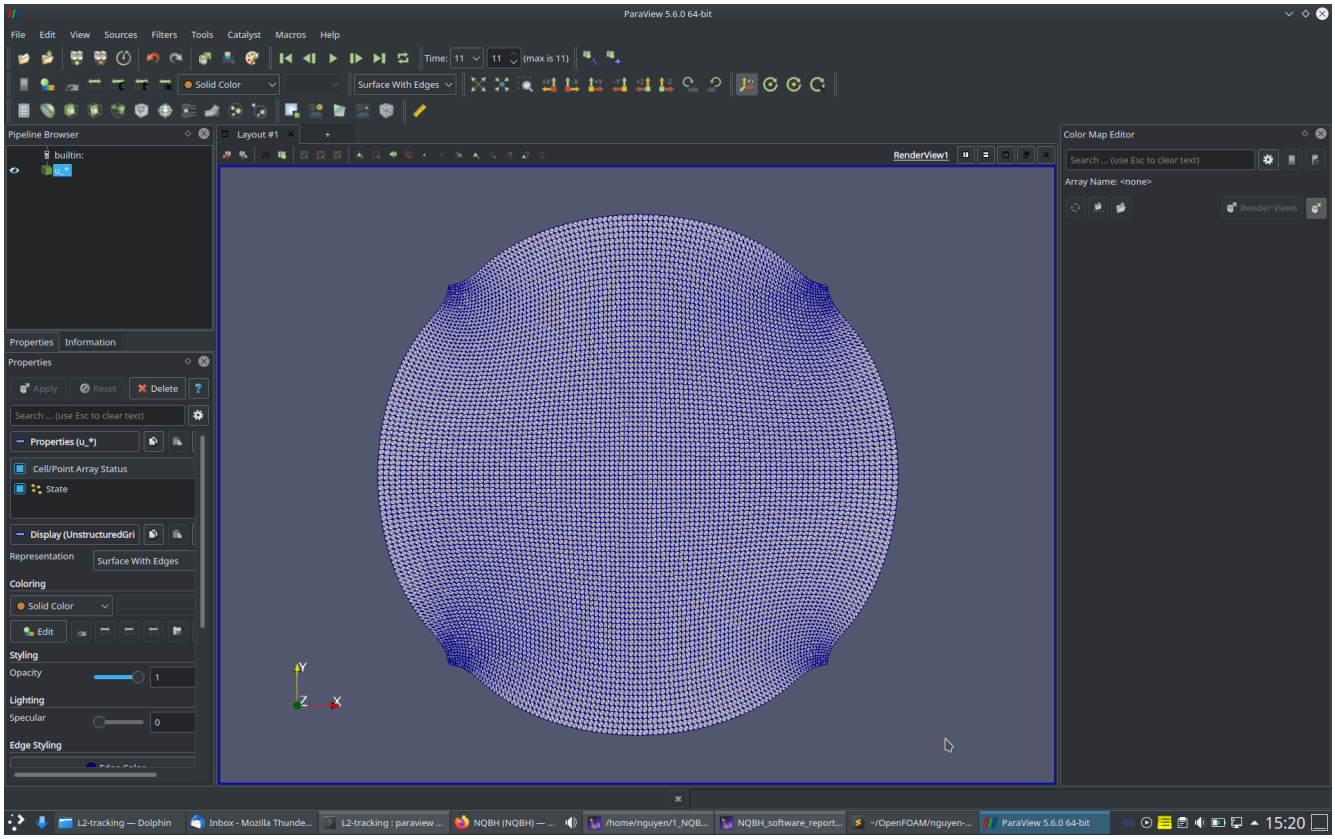

Figure 1.28: View the optimized domain after 11 optimization steps by ParaView with modes: `Solid Color, Surface with Edges`.

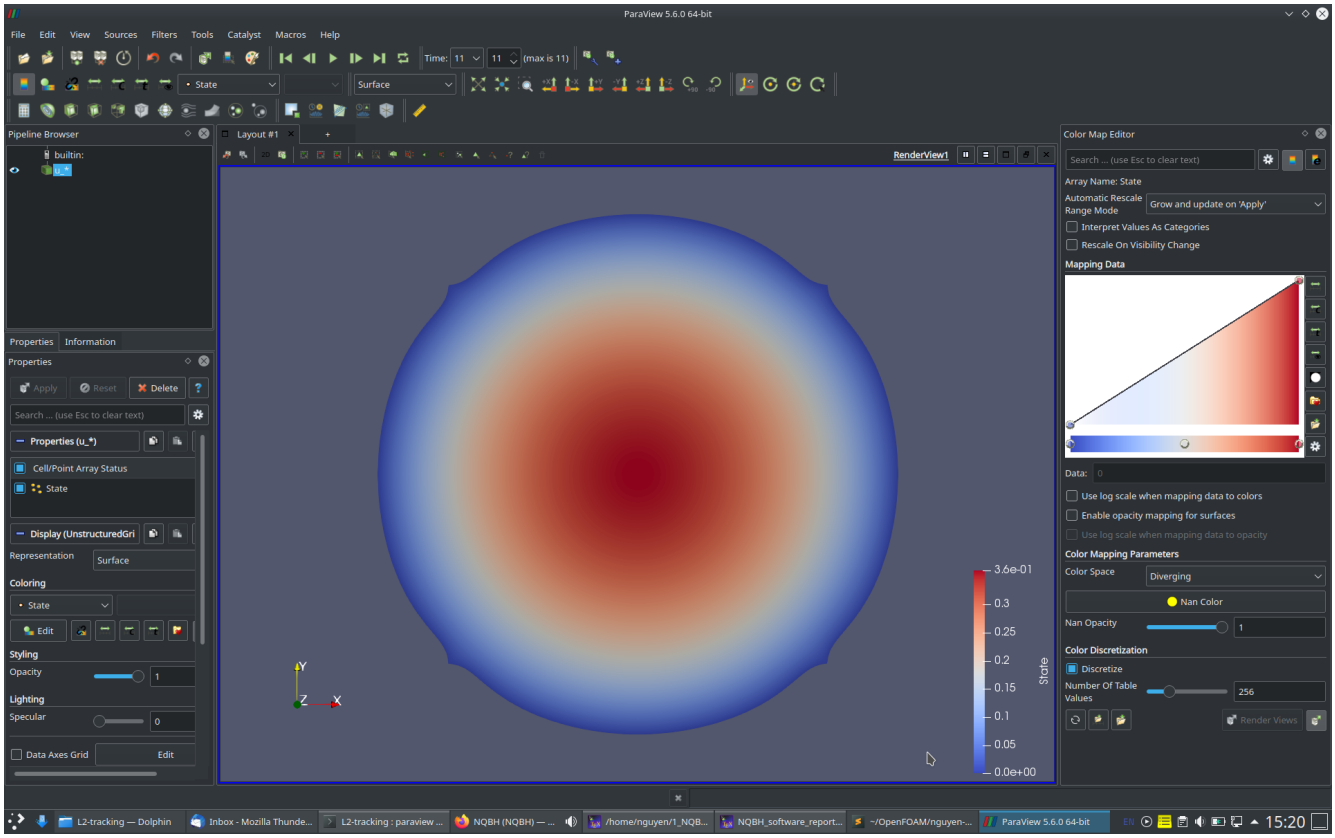

Figure 1.29: View the optimized domain after 11 optimization steps by ParaView with modes: `State, Surface`.

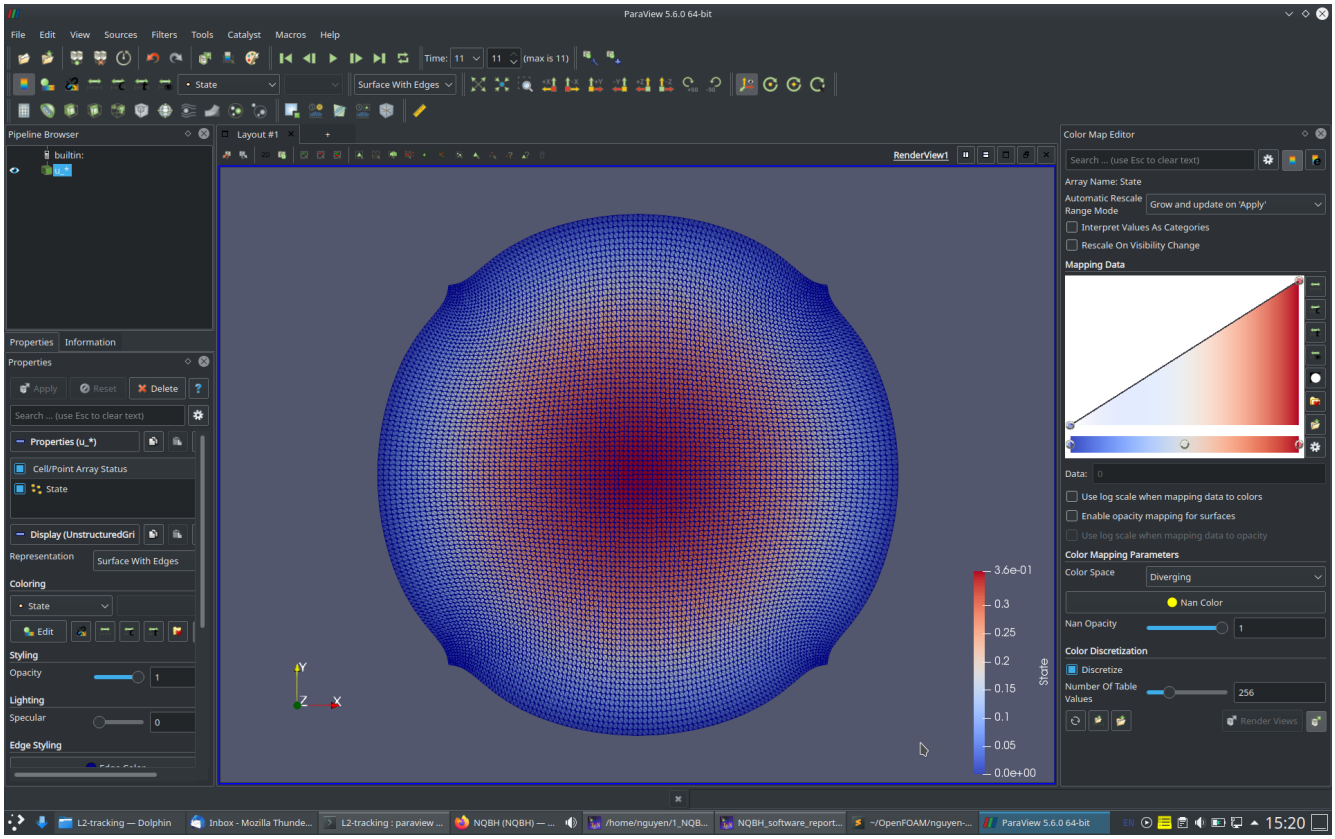

Figure 1.30: View the optimized domain after 11 optimization steps by ParaView with modes: `State, Surface with Edges`.

(c) **Example 3: Kinetic energy dissipation in a pipe**: Minimize the shape functional:

$$J(\Omega) = \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{u} \mathrm{d}\mathbf{x},$$

where the fluid velocity $\mathbf{u} : \mathbb{R}^d \to \mathbb{R}^d$ and the fluid pressure $p : \mathbb{R}^d \to \mathbb{R}$, $d \in \{2, 3\}$ satisfy the incompressible Navier-Stokes equations

$$\begin{cases} -\nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{0} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega \backslash \Gamma_{\text{out}}, \\ p\mathbf{n} - \nu \nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}}, \end{cases}$$

where $\mathbf{g}$ is given by a Poiseuille flow at the inlet and is zero on the walls of the pipe.

In addition to the PDE-constraint, enforce a *volume constraint*: the volume of the optimized domain should be equal to the volume of the initial domain: $\mathrm{Vol}(\Omega) = \mathrm{Vol}(\Omega_0) =: V_0$.

Numerical results.

- **Architecture.** Dell Intel i5, Ubuntu 18.04.4 installed on VirtualBox host machine installed on a host machine installed Ubuntu 20.04.
- **Runtime.** $\approx$ 45 minutes.

$\star$ **2D.**

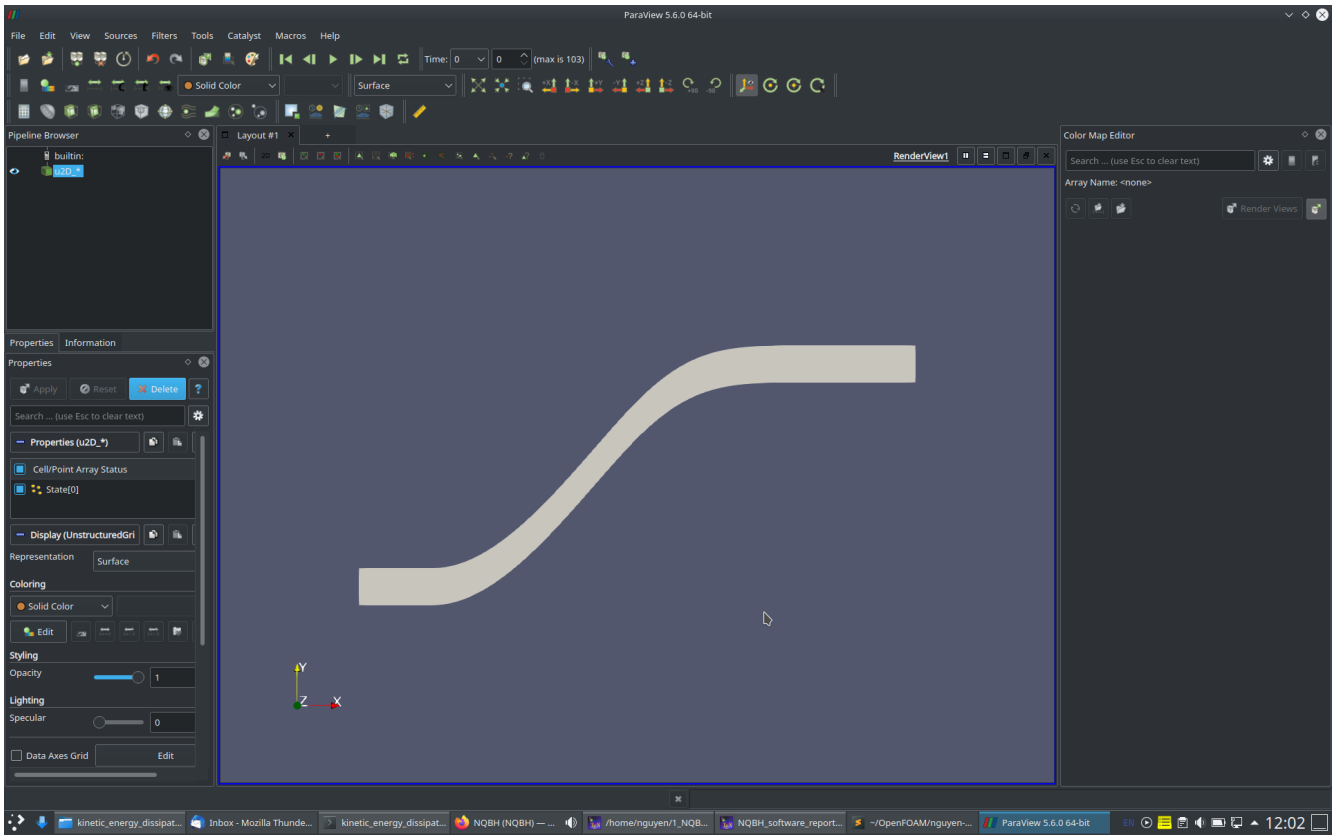

Figure 1.31: View the initial 2D domain in by ParaView with modes: `Solid Color, Surface`.

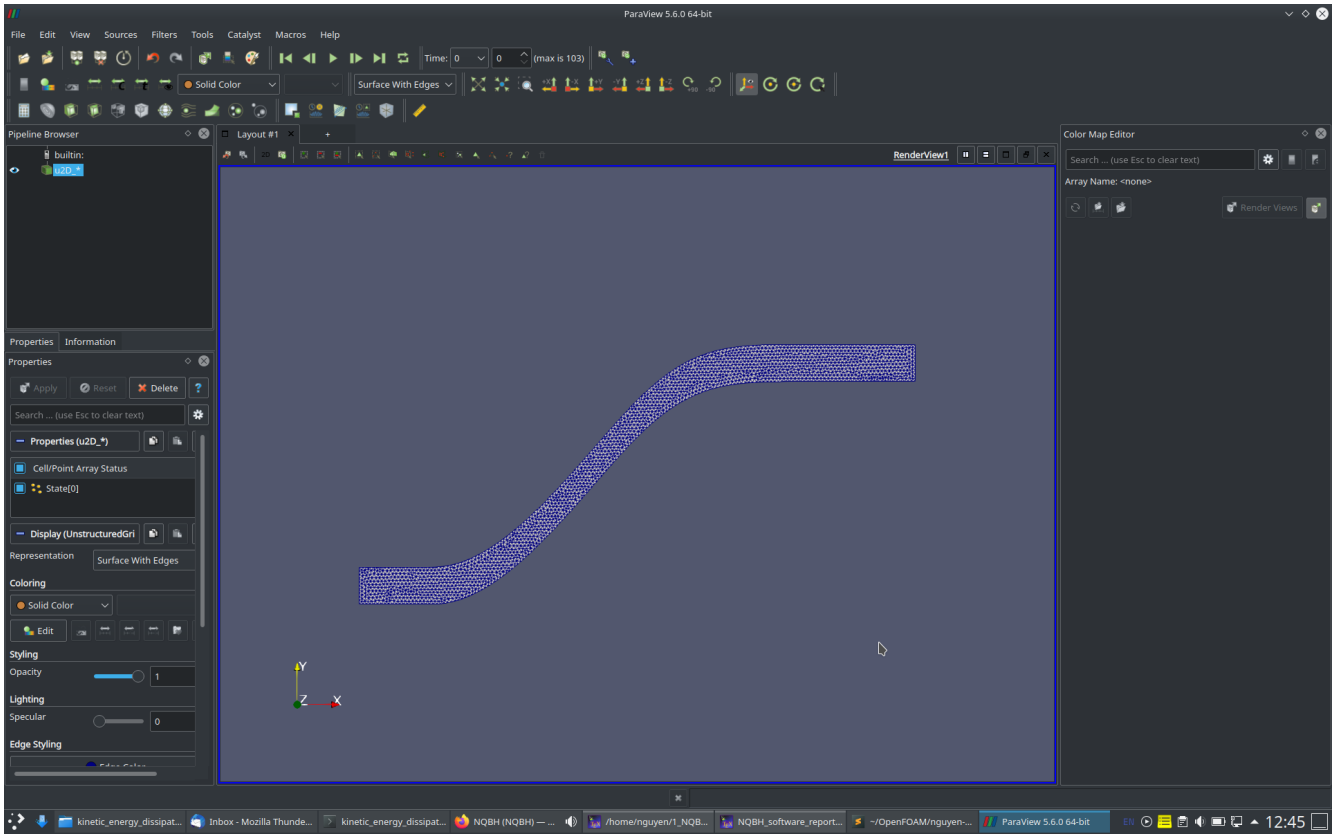

Figure 1.32: View the initial 2D domain by ParaView with modes: `Solid Color, Surface with Edges`.

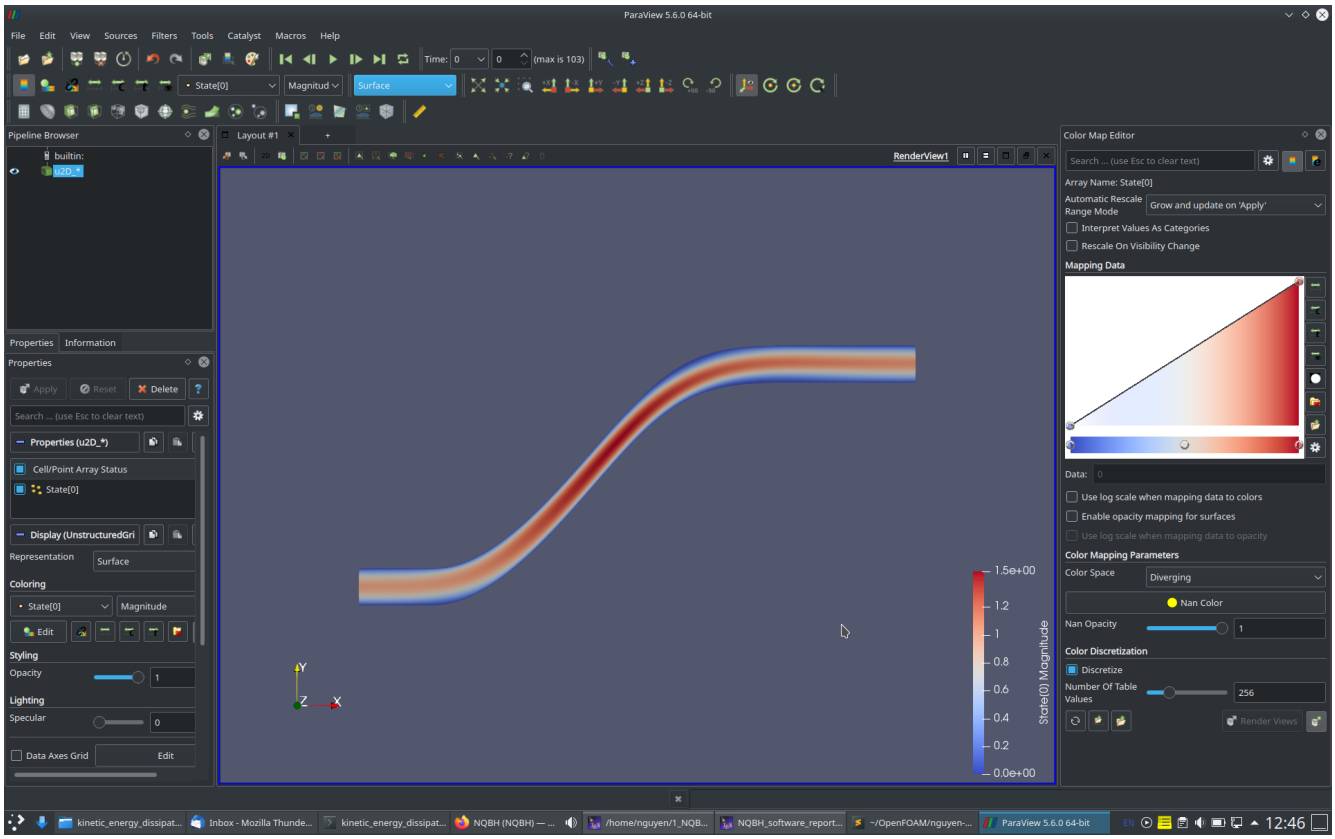

Figure 1.33: View the initial 2D domain by ParaView with modes: `State[0], Surface`.

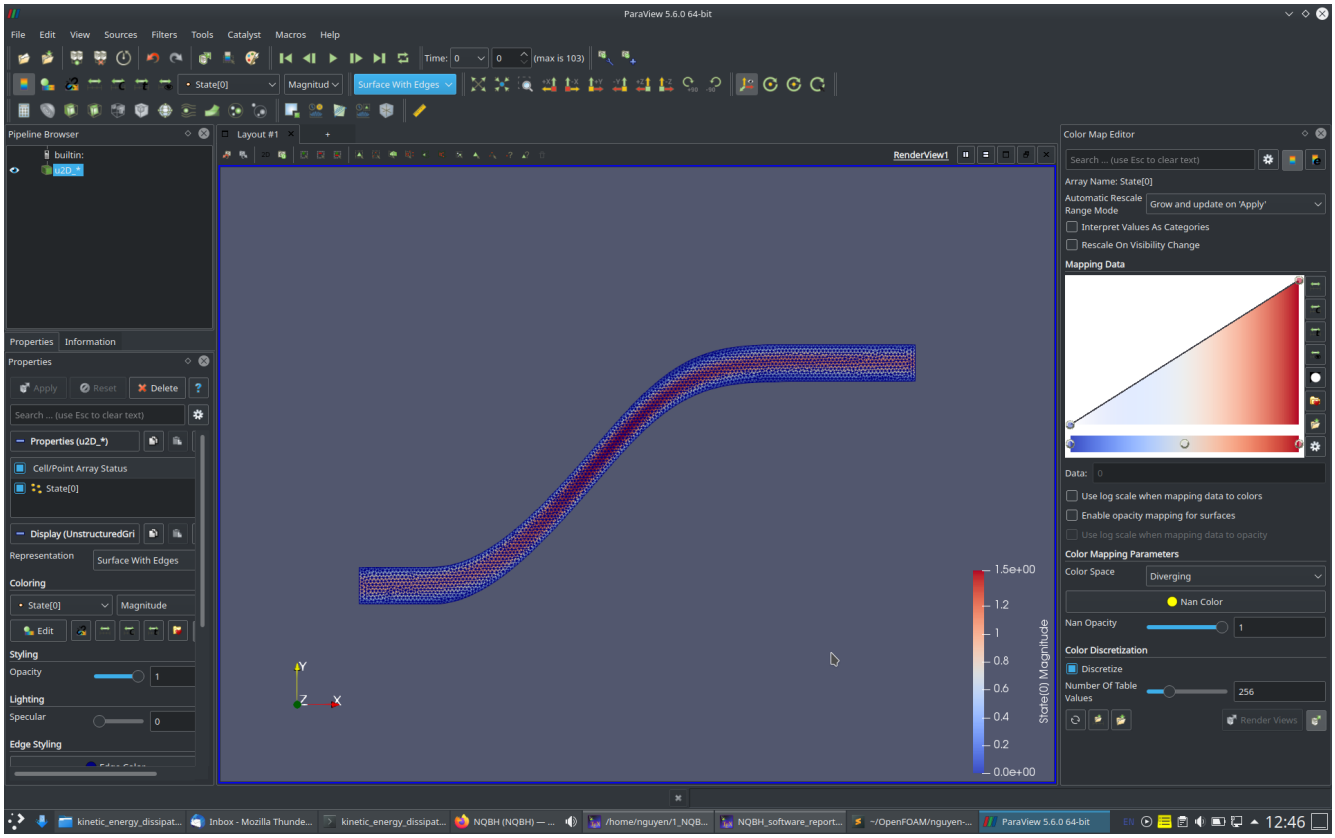

Figure 1.34: View the initial 2D domain by ParaView with modes: `State[0], Surface with Edges`.

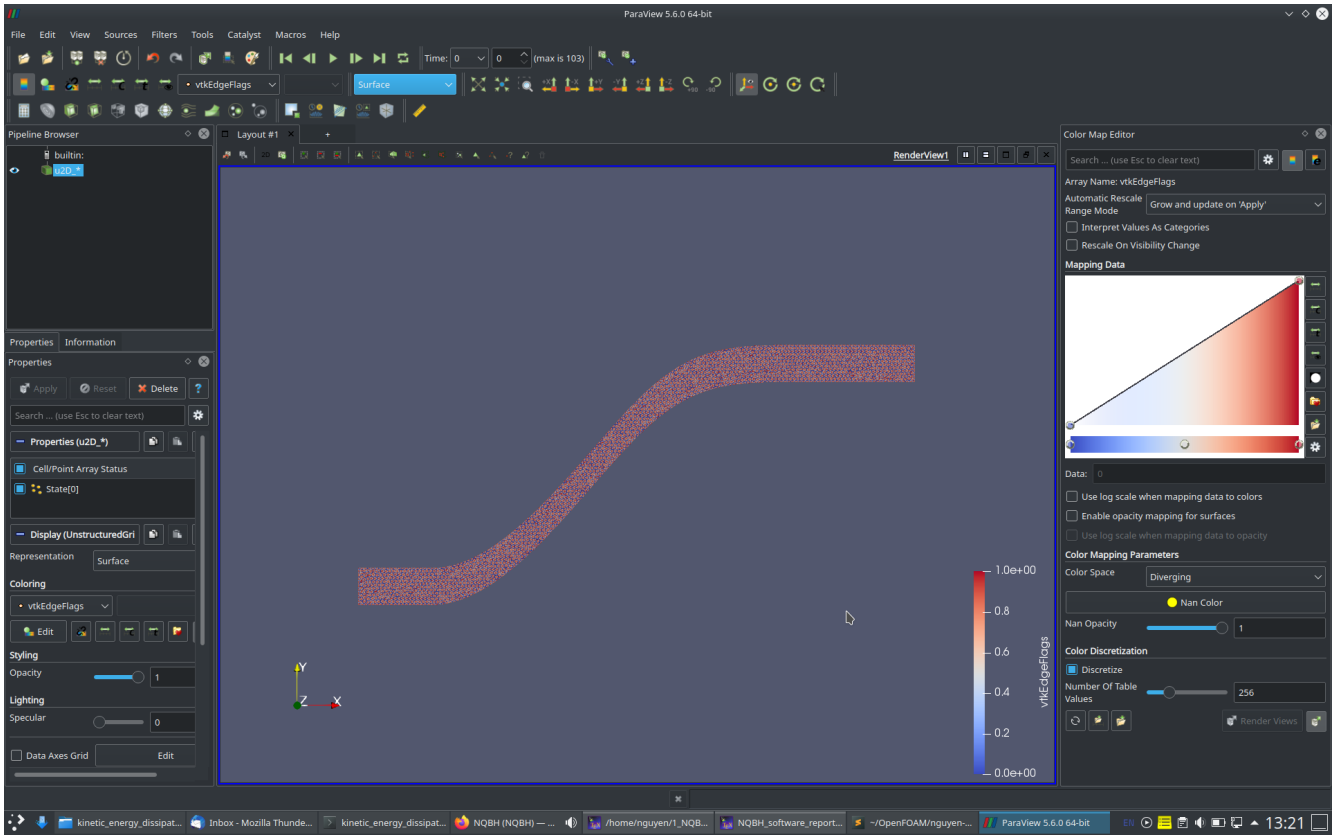

Figure 1.35: View the initial 2D domain by ParaView with modes: `vtkEdgeFlags, Surface`.

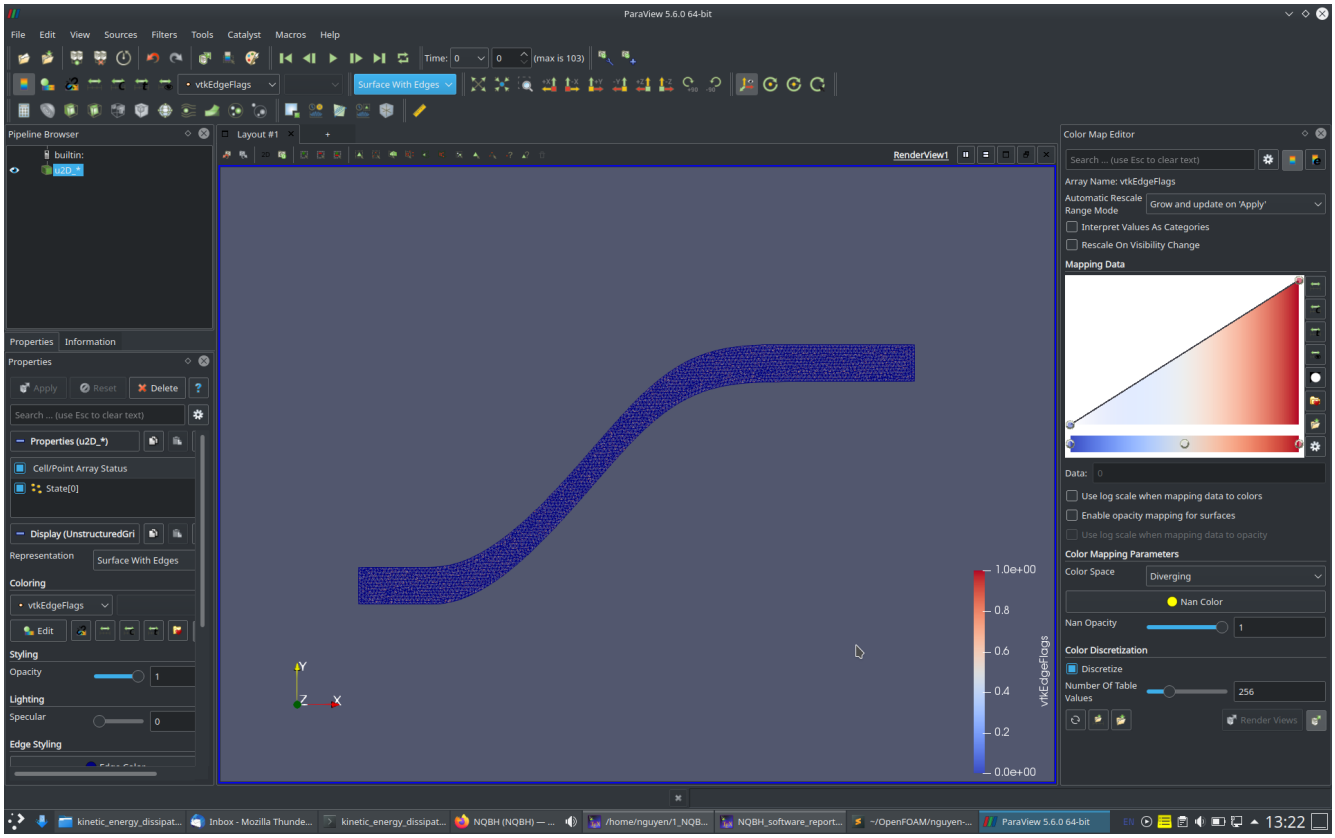

Figure 1.36: View the initial 2D domain by ParaView with modes: `vtkEdgeFlags`, `Surface with Edges`.

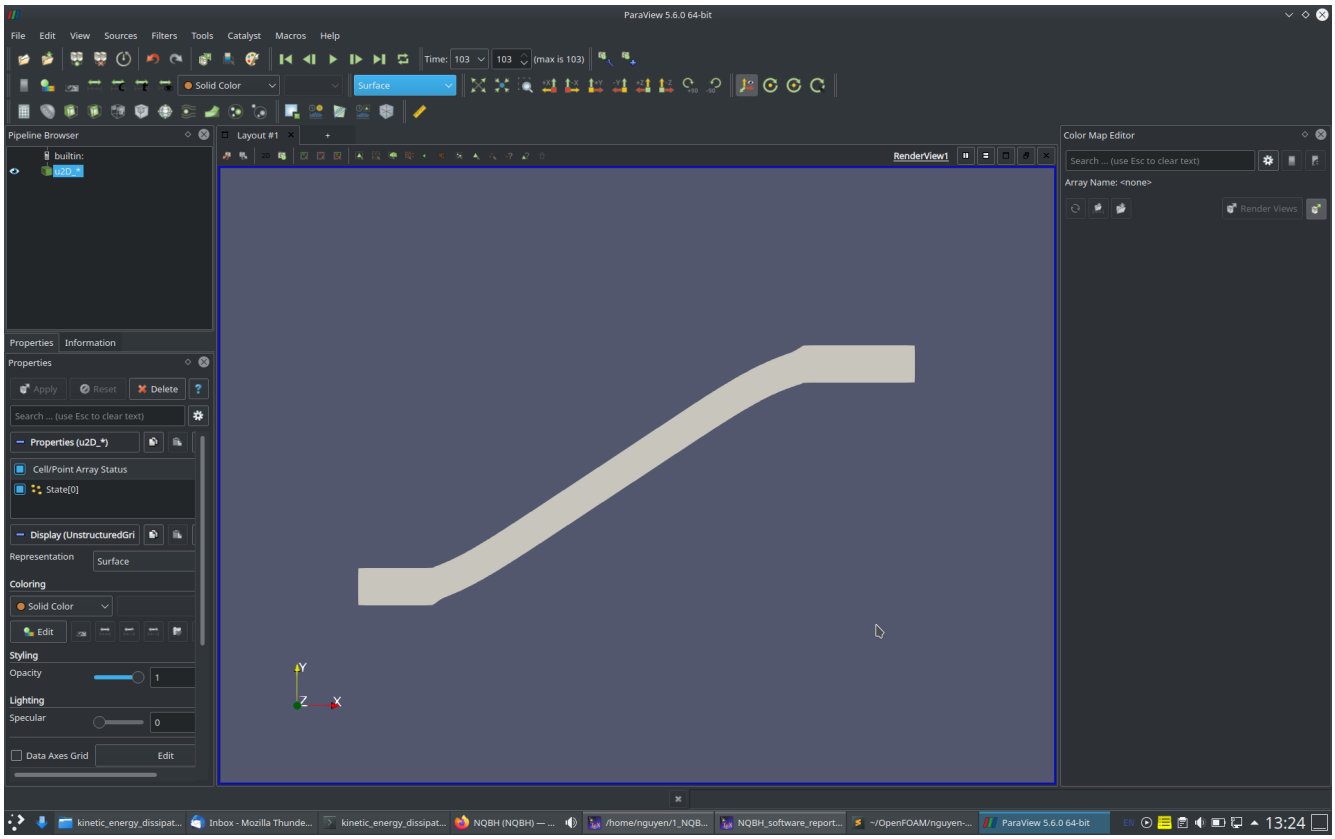

Figure 1.37: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `Solid Color`, `Surface`.

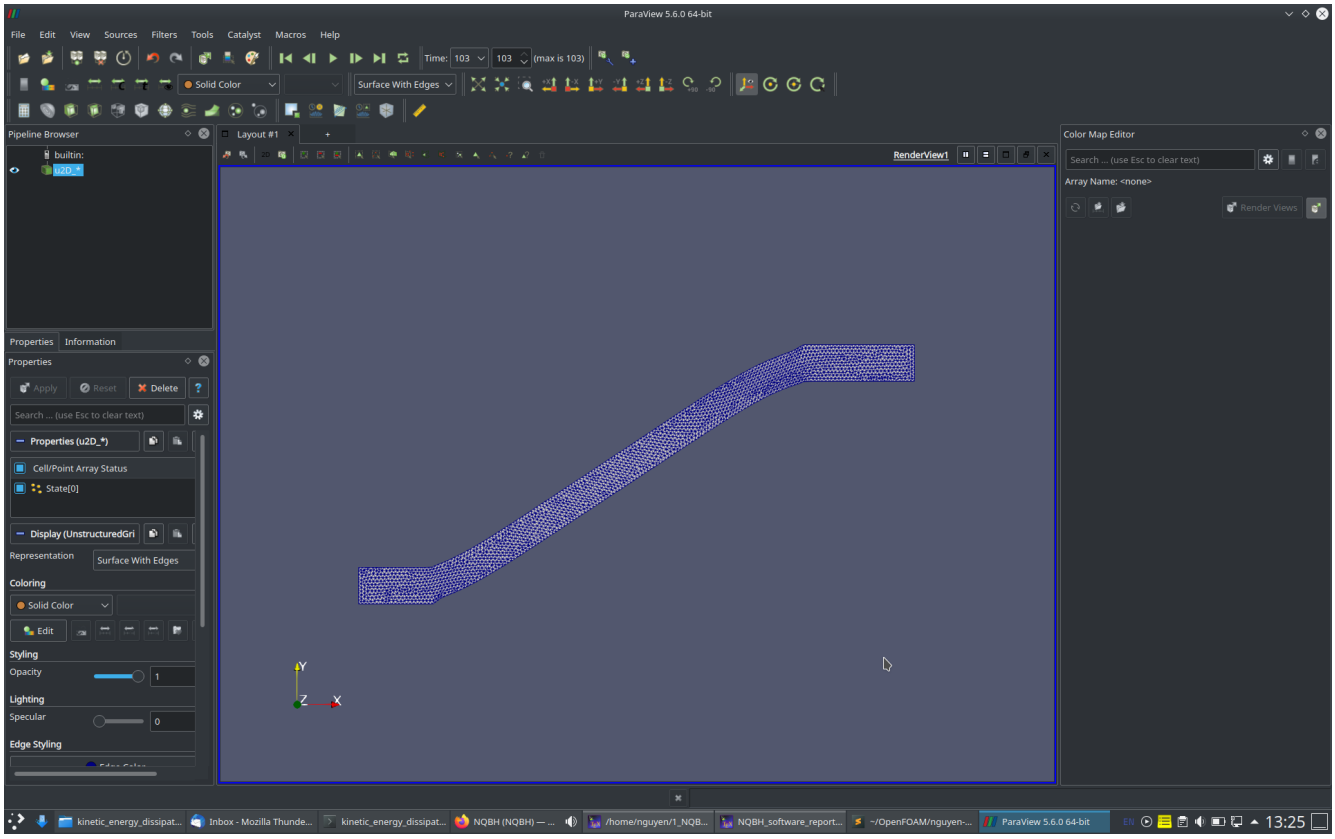

Figure 1.38: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `Solid Color`, `Surface with Edges`.

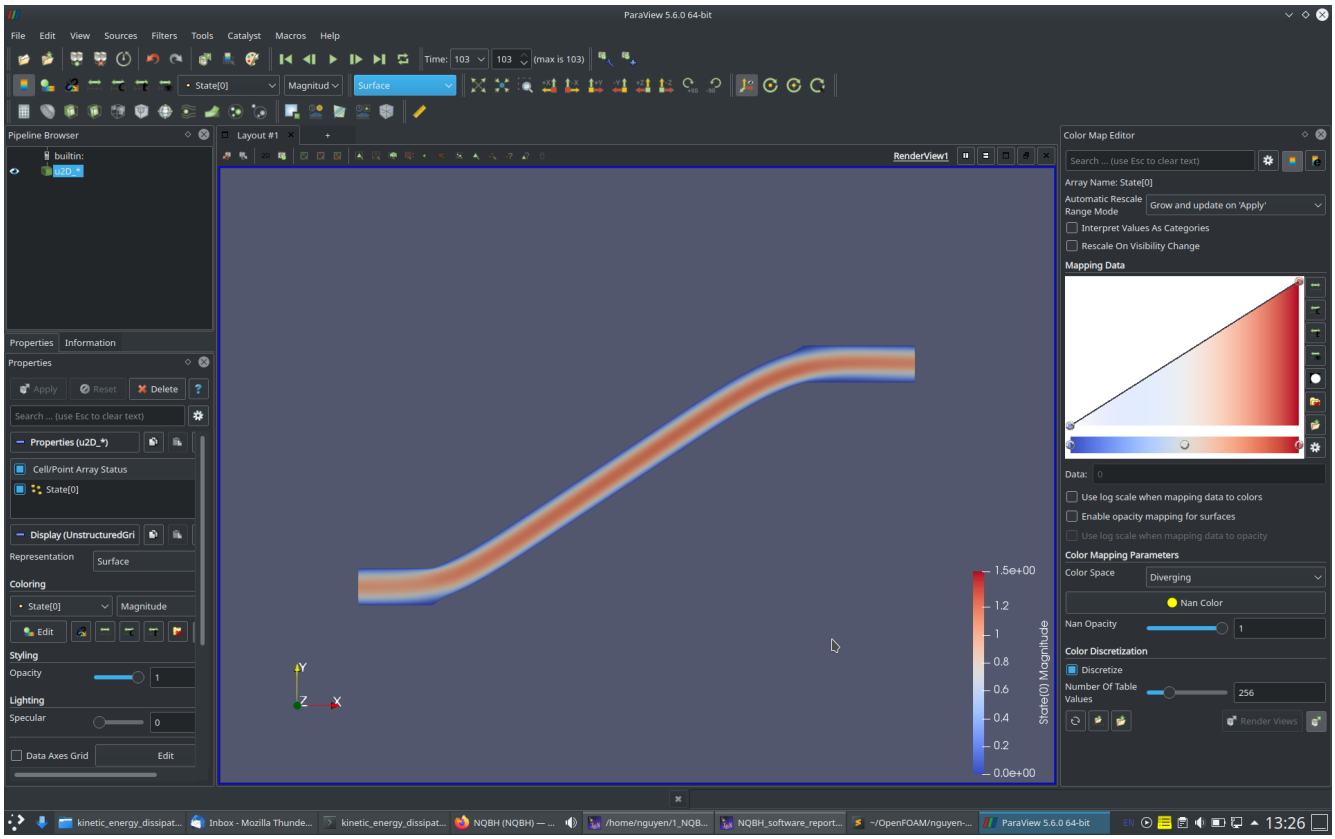

Figure 1.39: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `State[0]`, `Surface`.

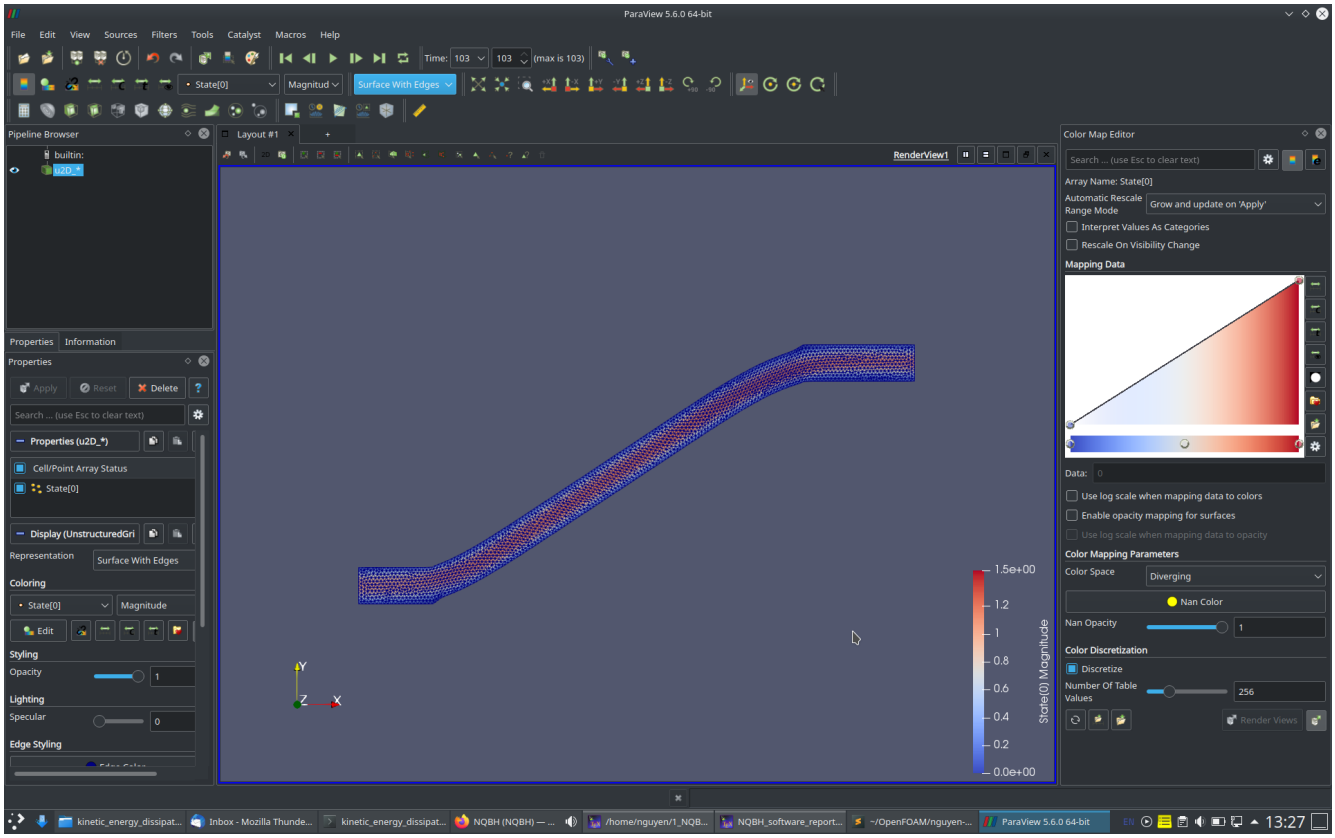

Figure 1.40: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `State[0]`, `Surface with Edges`.

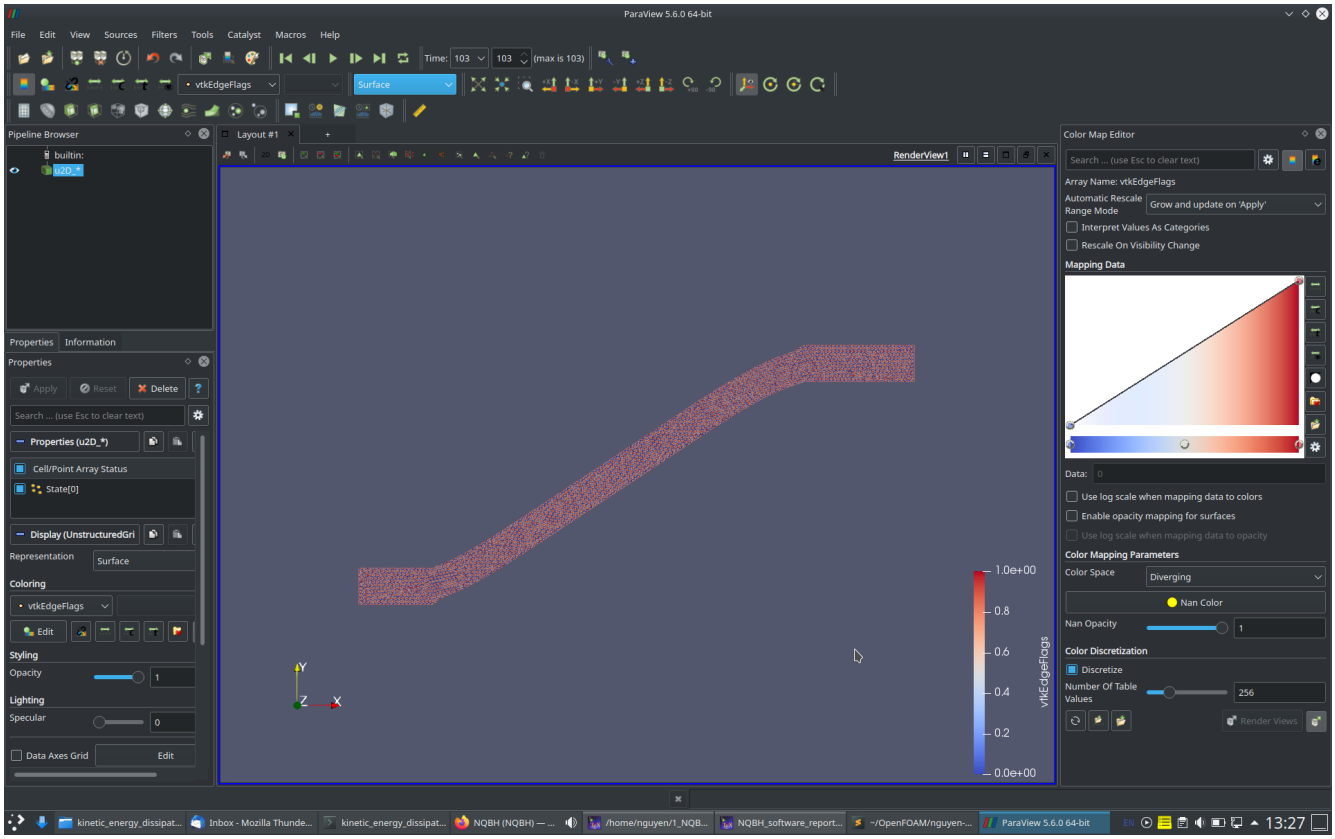

Figure 1.41: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `vtkEdgeFlags`, `Surface`.

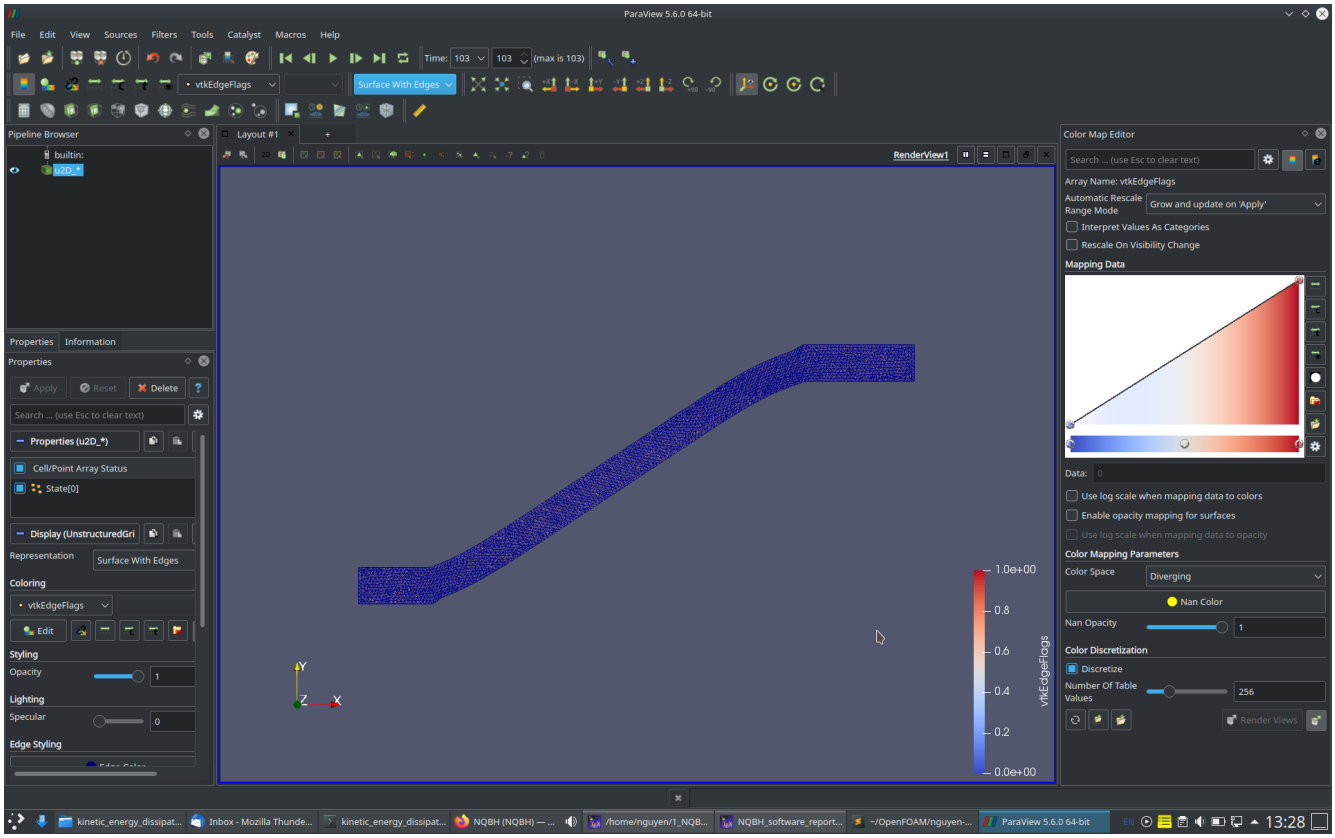

Figure 1.42: View the optimized 2D domain after 103 optimization steps by ParaView with modes: `vtkEdgeFlags, Surface with Edges`.

⋆ **3D.**

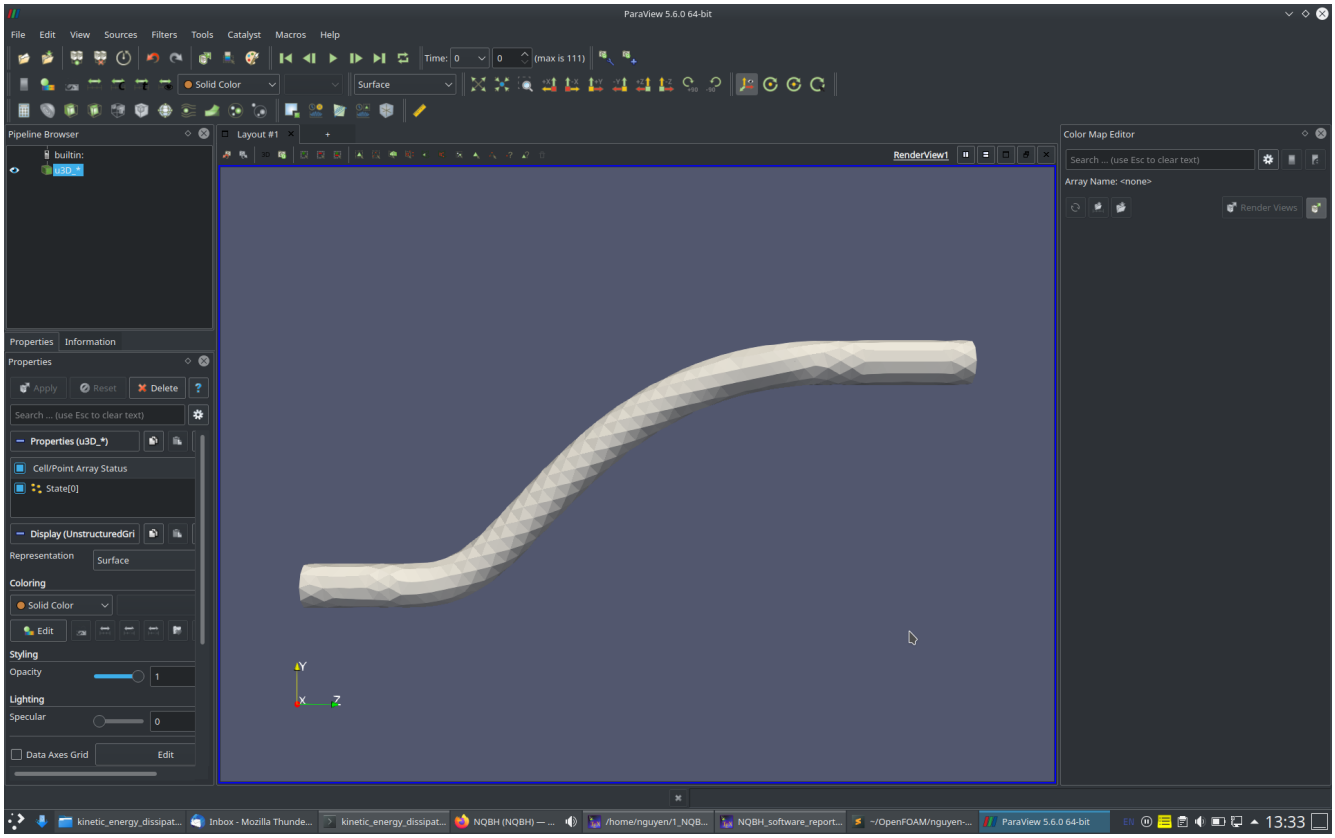

Figure 1.43: View the initial 3D domain by ParaView with modes: `Solid Color, Surface`.

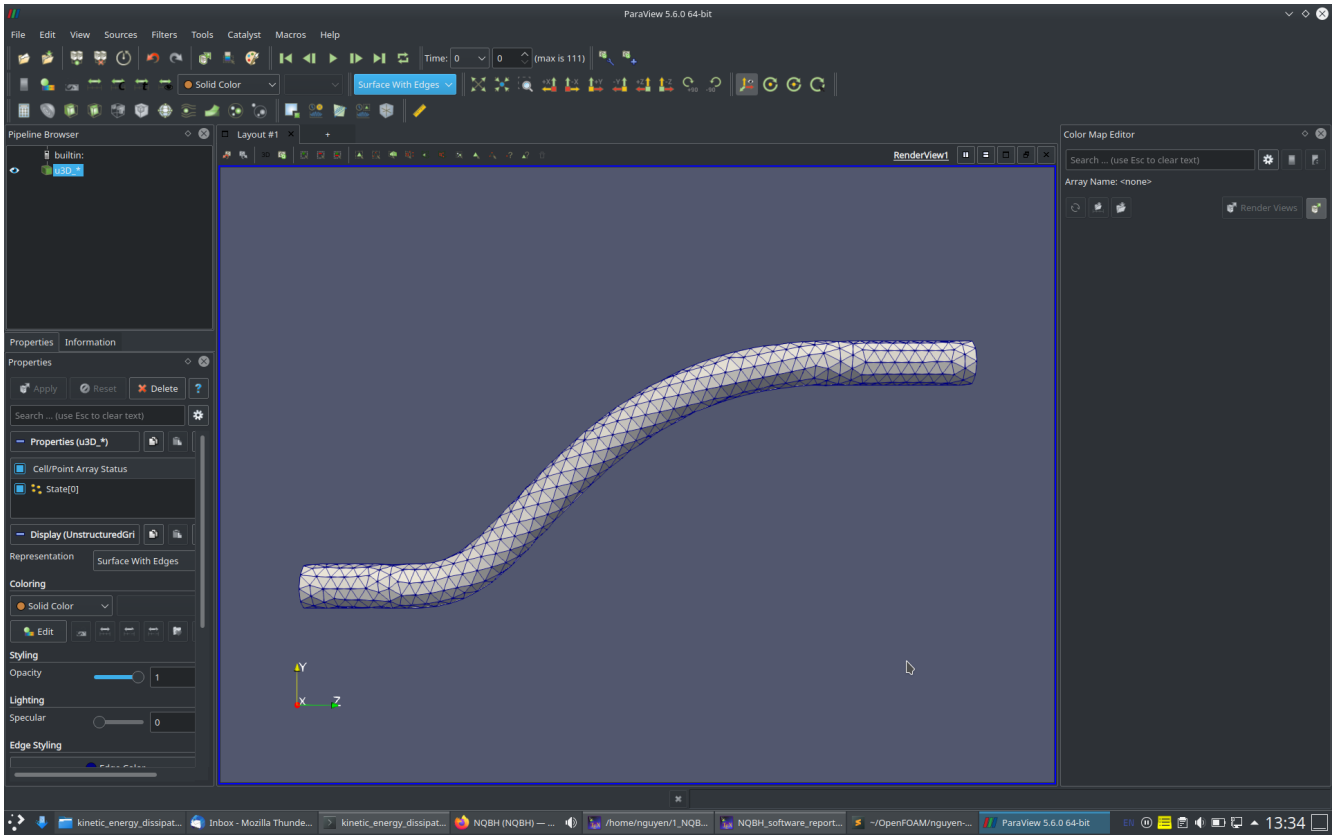

Figure 1.44: View the initial 3D domain by ParaView with modes: `Solid Color, Surface with Edges`.

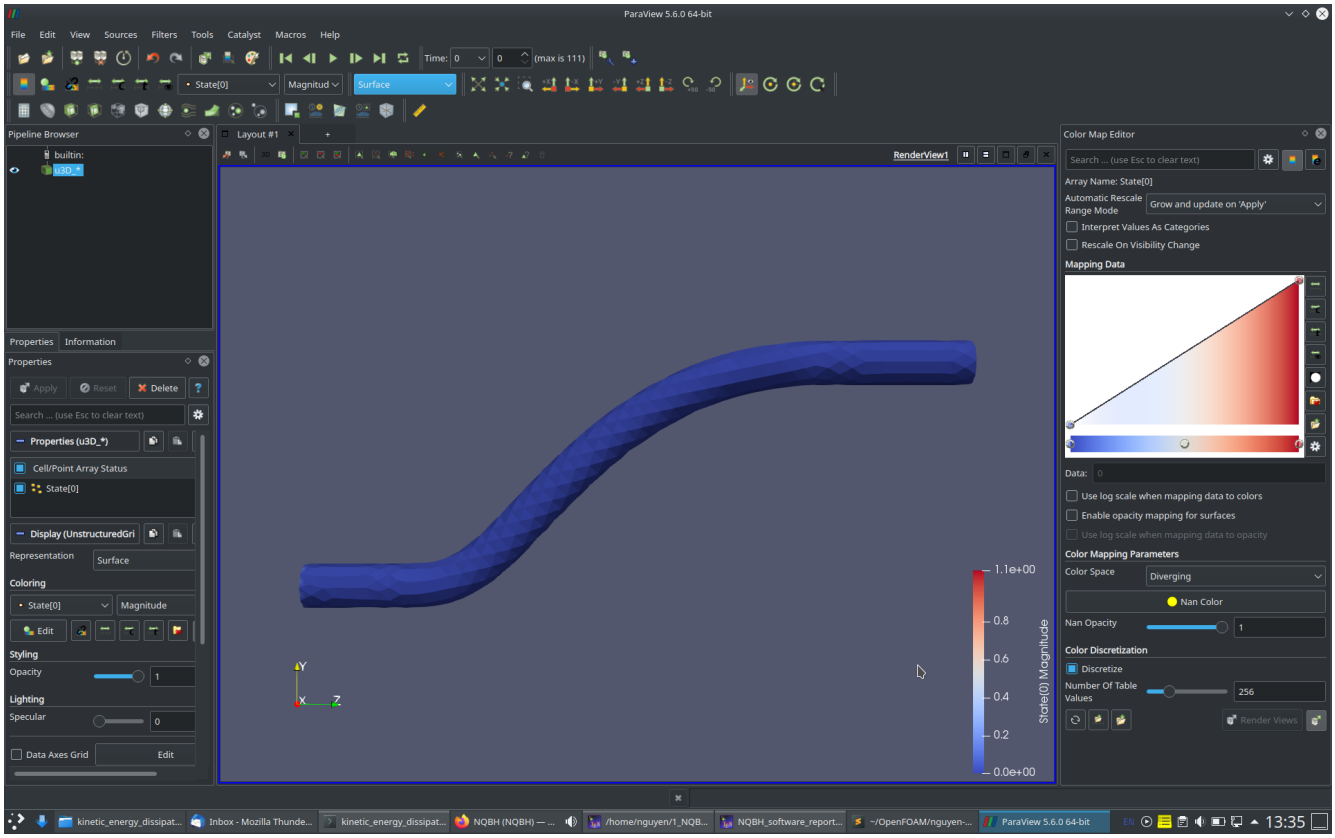

Figure 1.45: View the initial 3D domain by ParaView with modes: `State[0]`, `Surface`.

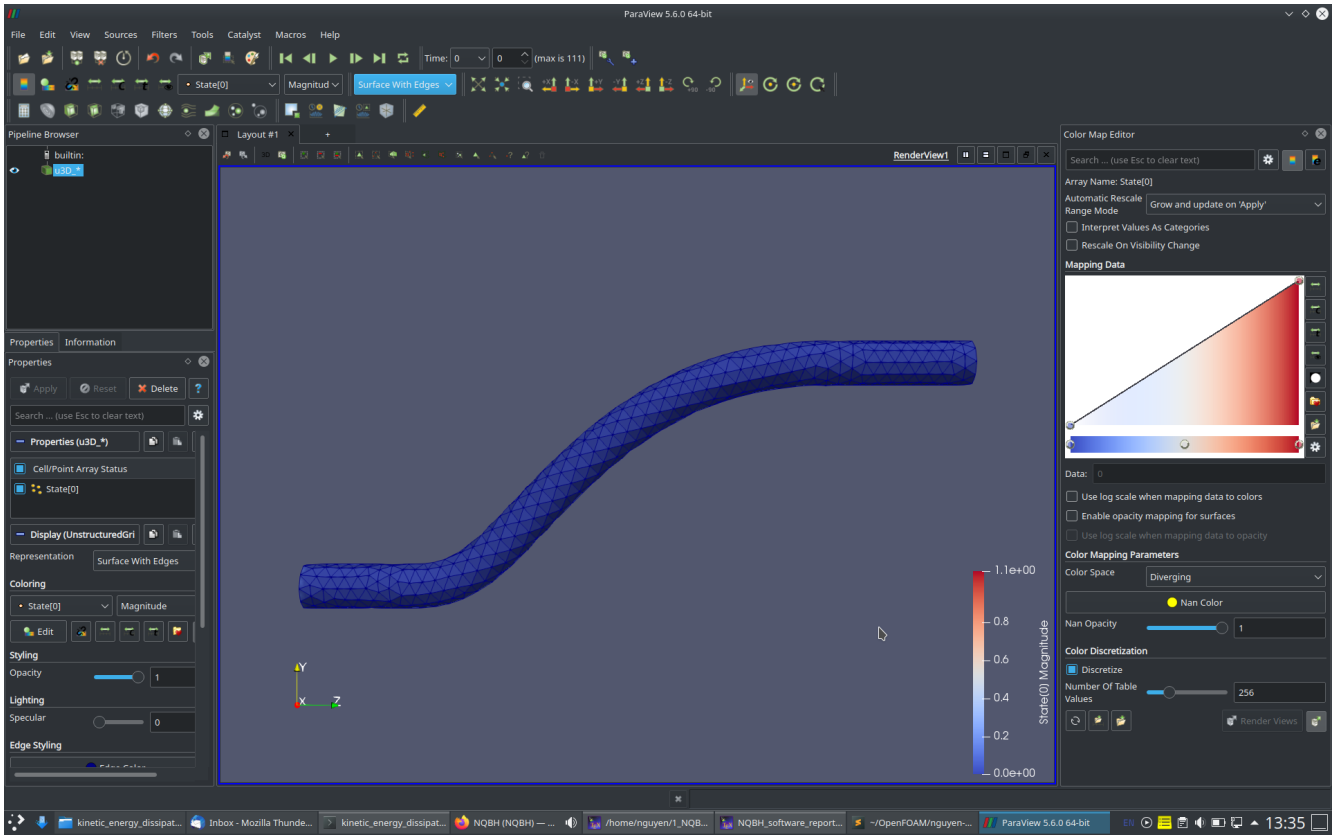

Figure 1.46: View the initial 3D domain by ParaView with modes: `State[0]`, `Surface with Edges`.

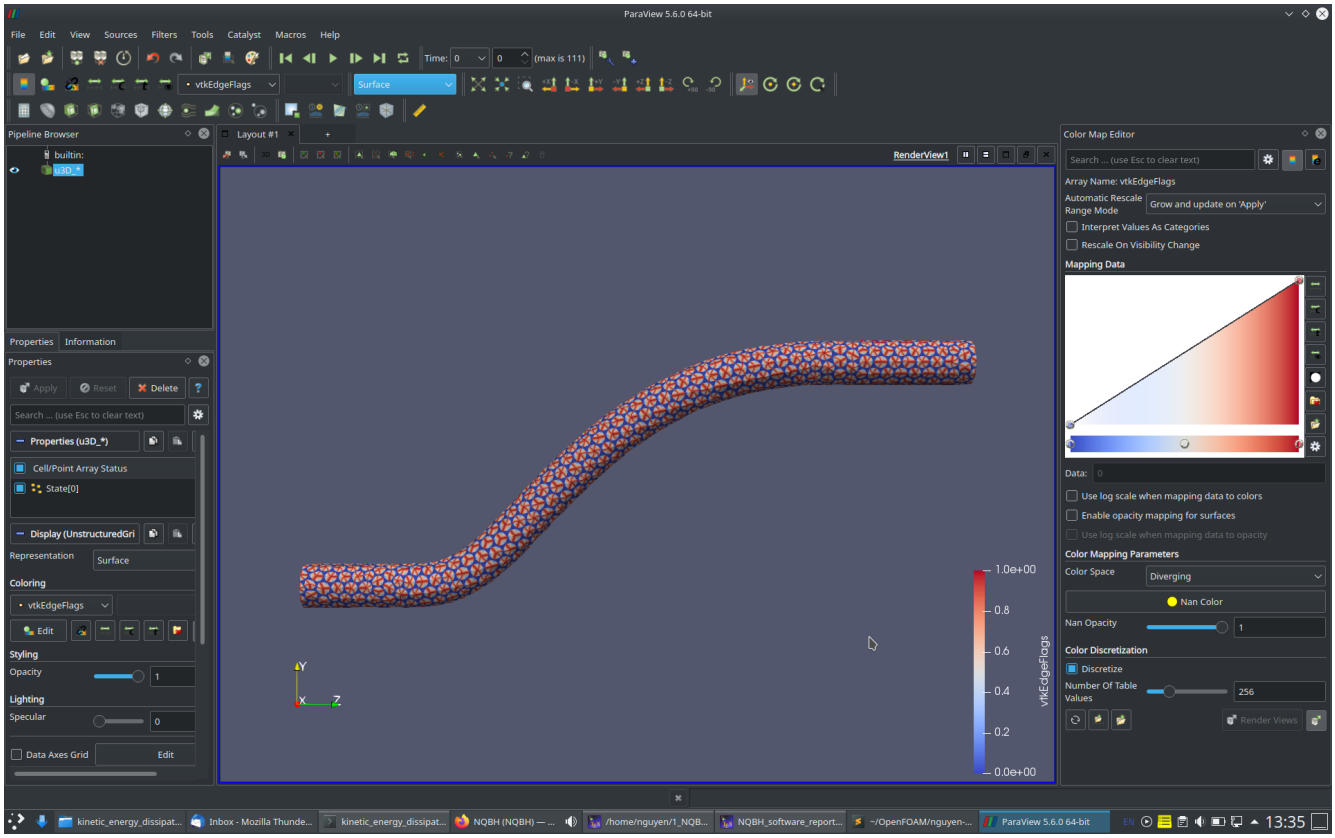

Figure 1.47: View the initial 3D domain by ParaView with modes: `vtkEdgeFlags, Surface`.

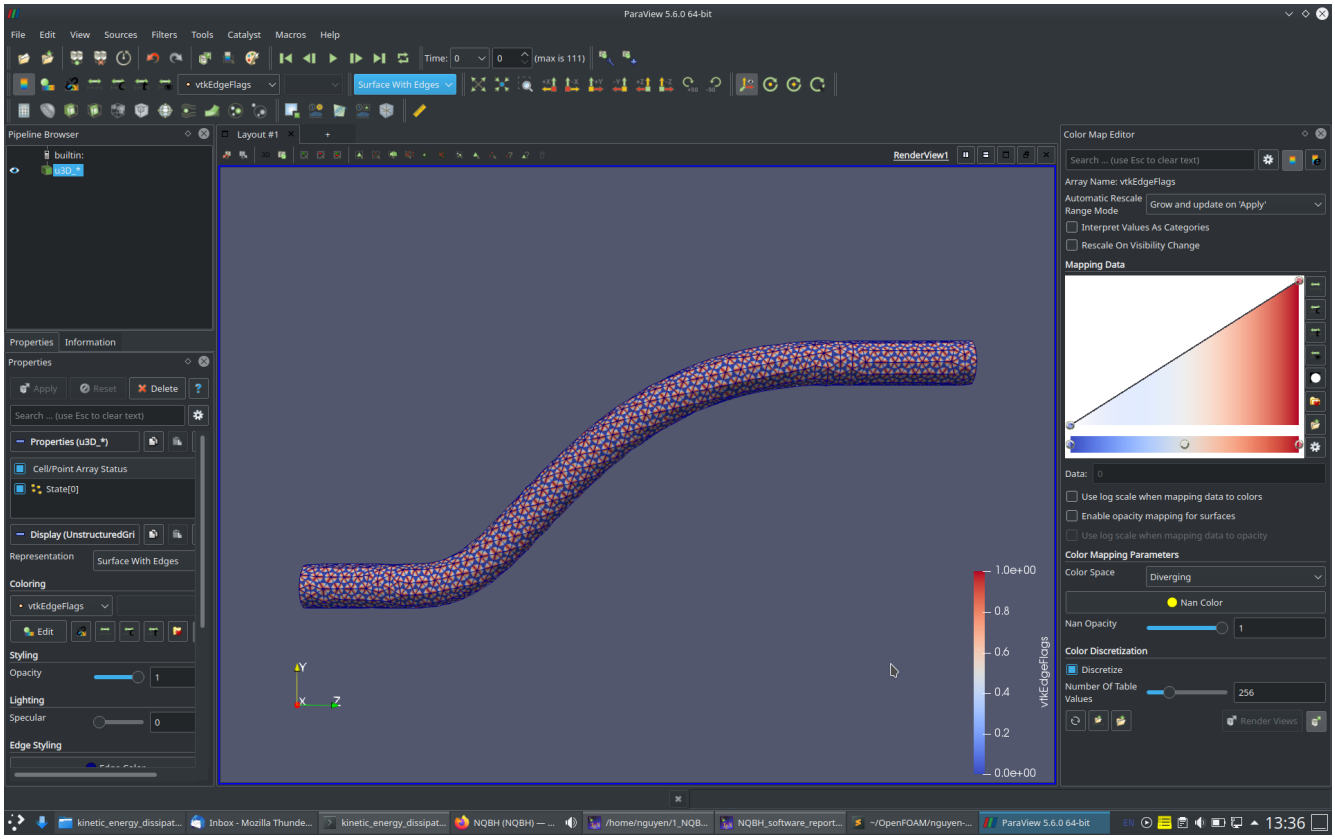

Figure 1.48: View the initial 3D domain by ParaView with modes: `vtkEdgeFlags, Surface with Edges`.

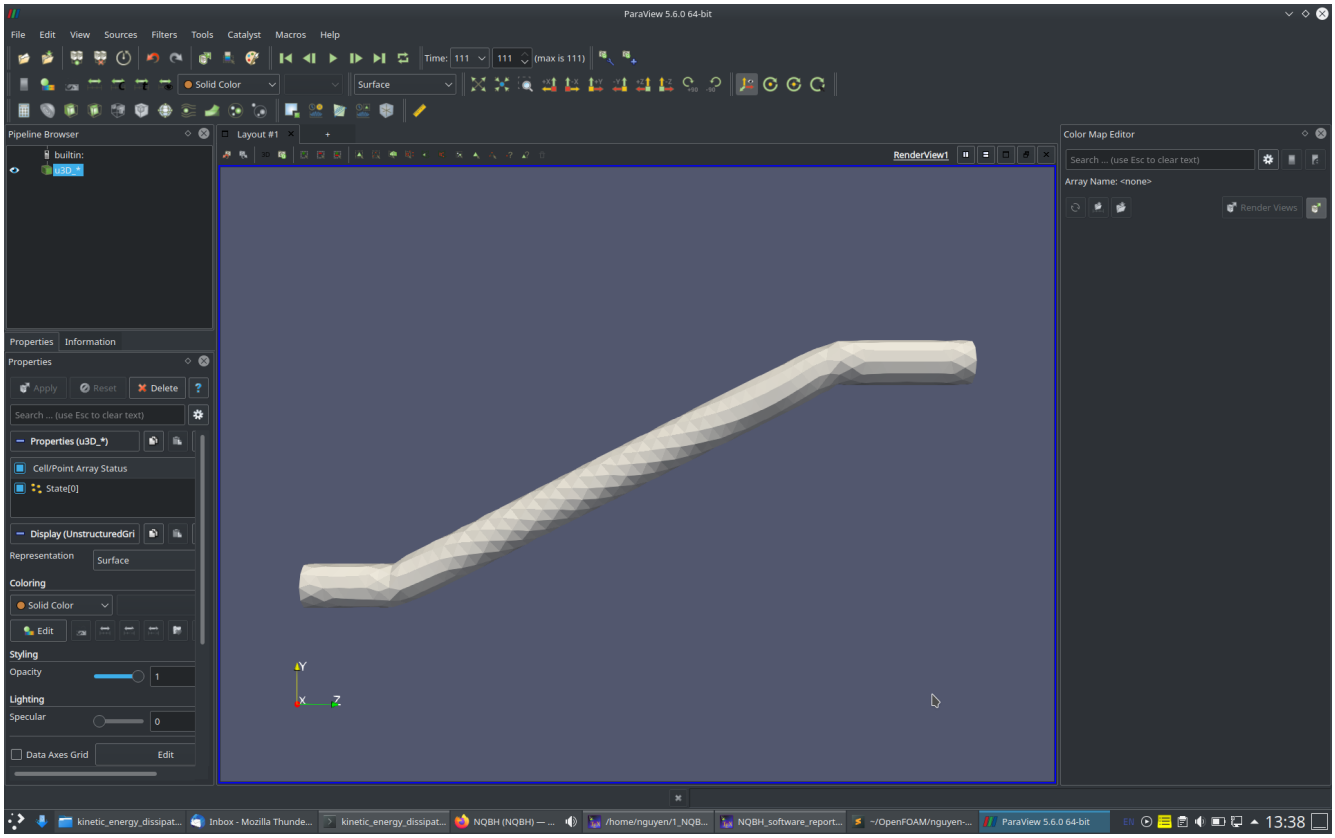

Figure 1.49: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `Solid Color, Surface`.

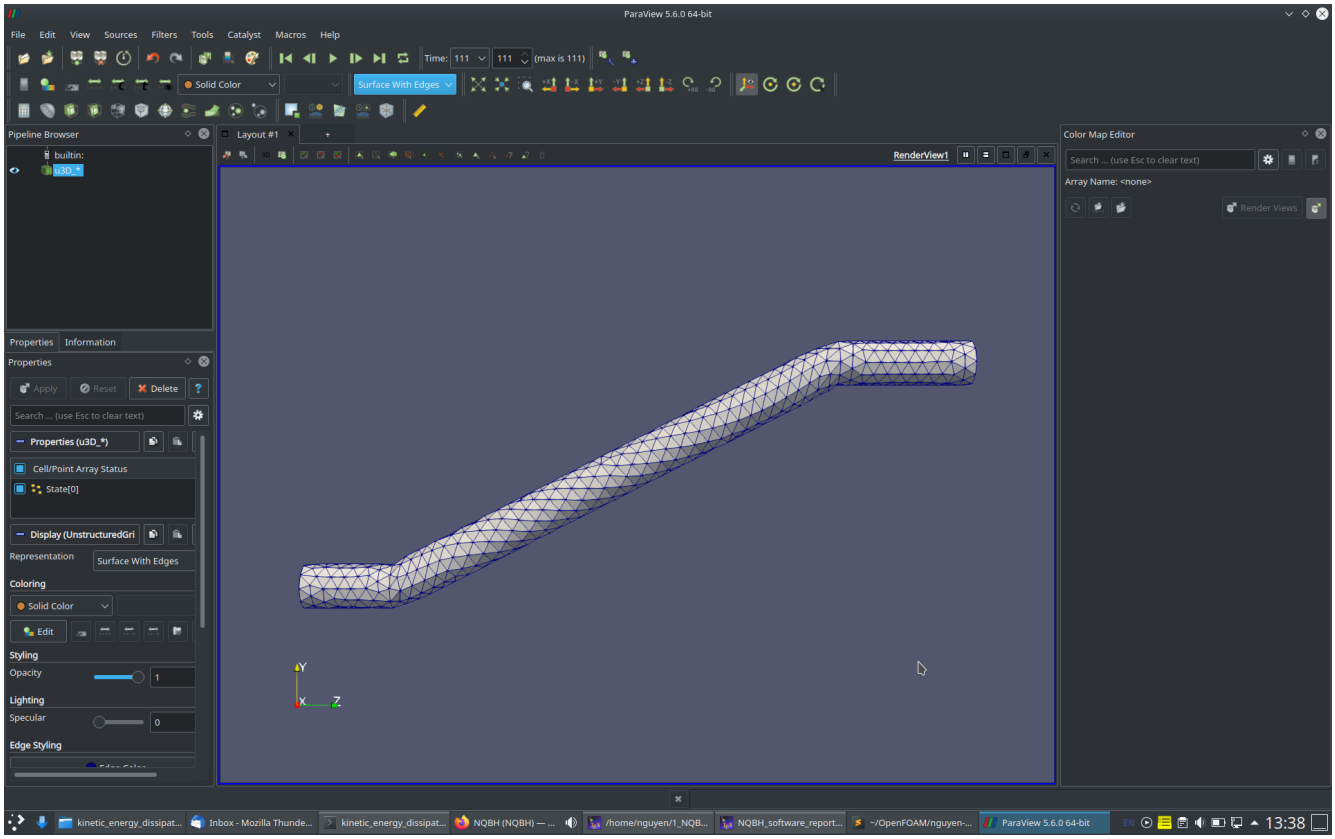

Figure 1.50: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `Solid Color, Surface with Edges`.

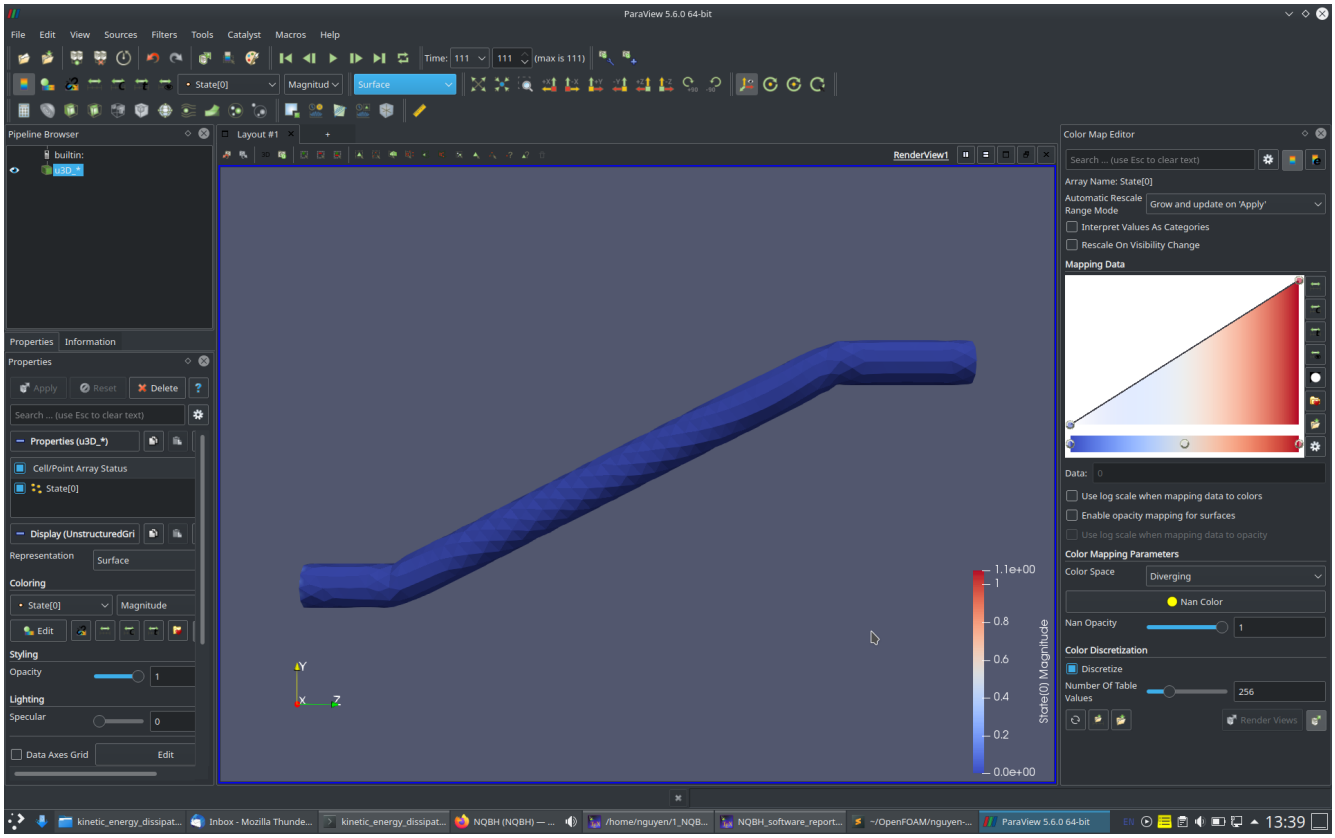

Figure 1.51: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `State[0]`, `Surface`.

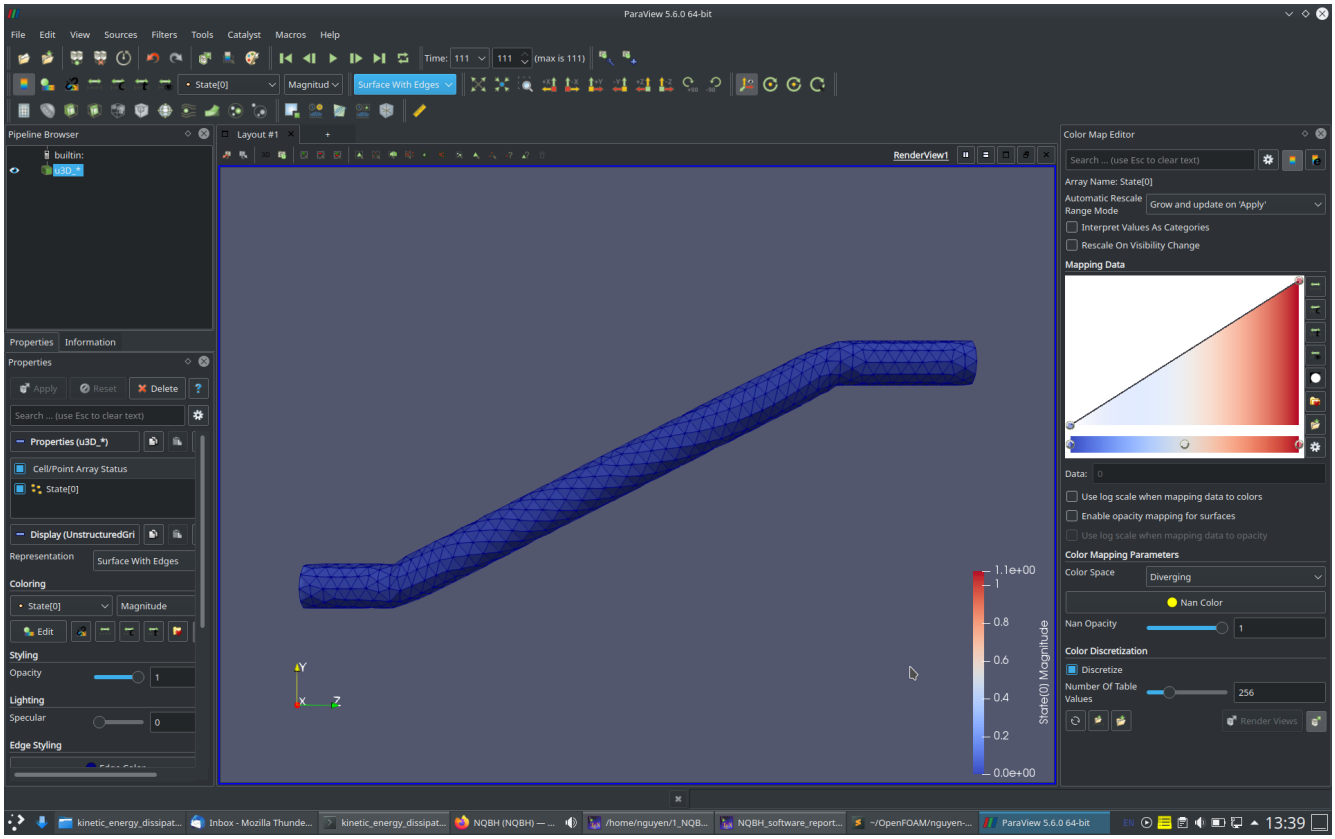

Figure 1.52: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `State[0]`, `Surface with Edges`.

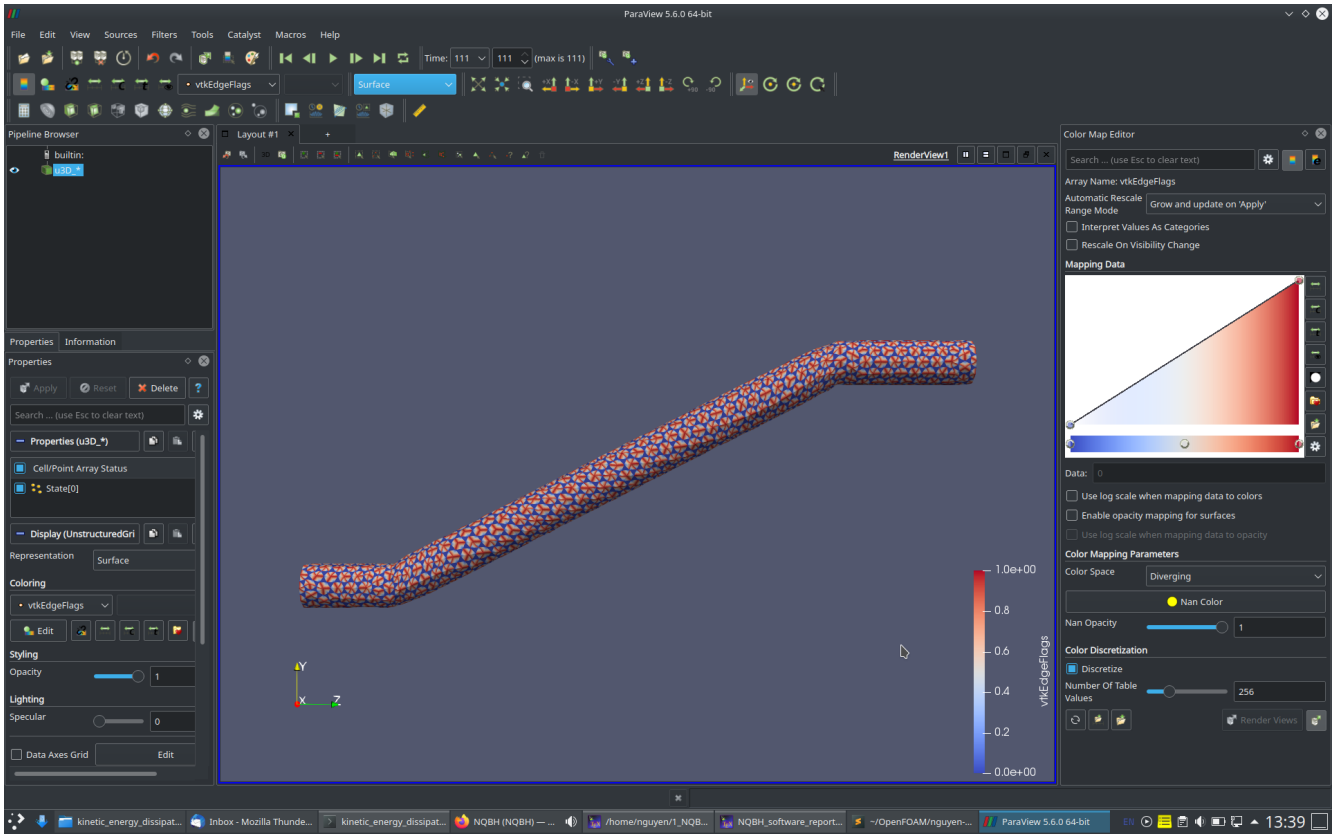

Figure 1.53: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `vtkEdgeFlags, Surface`.

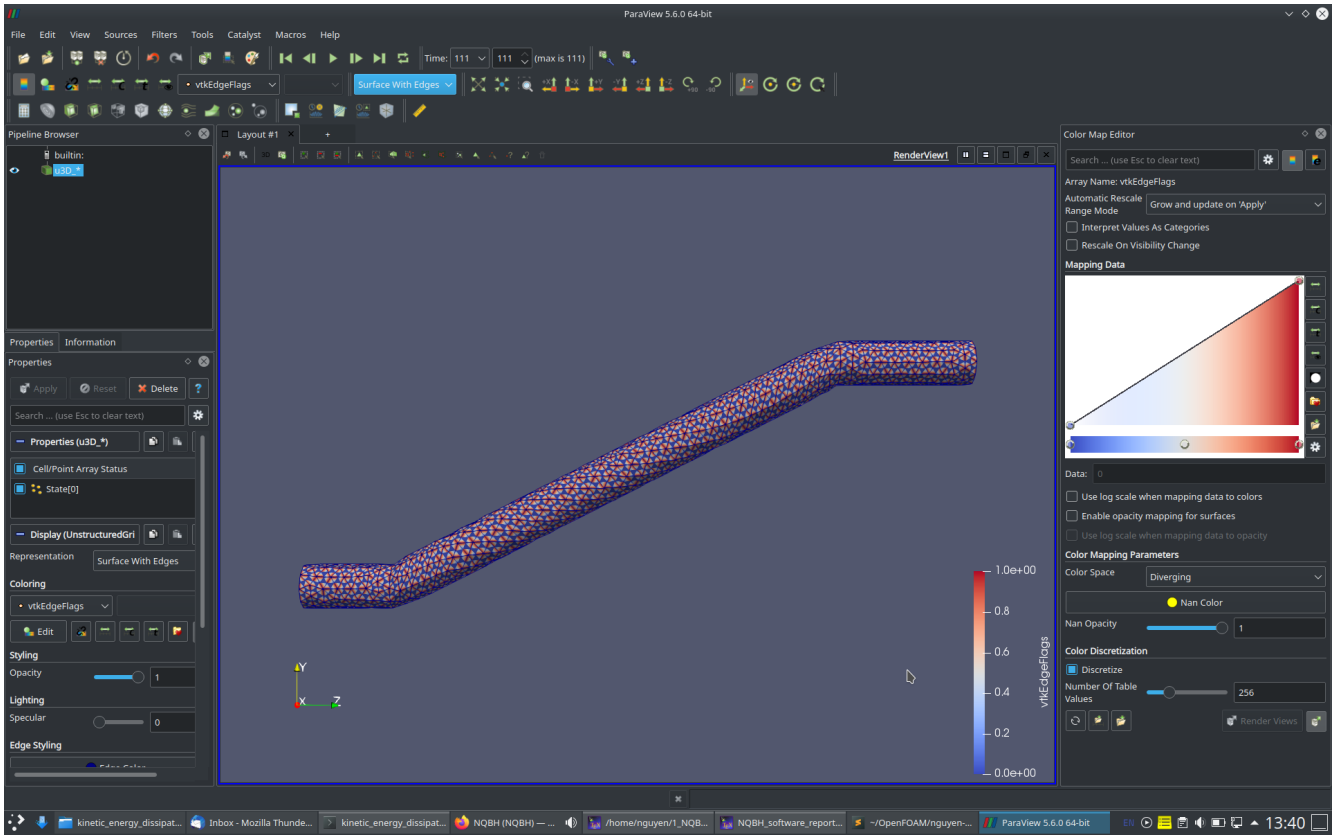

Figure 1.54: View the optimized 3D domain after 111 optimization steps by ParaView with modes: `vtkEdgeFlags, Surface with Edges`.

(d) **Using ROL in Fireshape**.

- "Fireshape allows solving shape optimization problems using the Rapid Optimization Library (ROL).
- The goal of this page is to give a very brief introduction to ROL and how to use it in Fireshape.
- Please note that this is not an official guide, but it's merely based on reverse engineering ROL's source code."

1.6 SU2

1. **Website.** https://su2code.github.io/
2. **GitHub repository.** https://github.com/su2code/SU2

   Remark 1.6.1. *Dr. Stephan Schmidt said that, in his opinion, SU2 is currently the best tool for multiphysics simulation and turbulence modeling.*

1.7 ParMooN

1. **Website.** https://cmg.cds.iisc.ac.in/parmoon/
2. **GitHub repository.** https://github.com/sassanin/ParMooN.

   Remark 1.7.1. *Prof. Volker John, Dr. Alfonso Caiazzo, and Dr. Ulrich Wilbrandt in Research Group 3, WIAS, confirmed that the version provided by this website is outdated. I contacted them and received the latest version of ParMooN already via Mercurial (hg) https://repos.wias-berlin.de/ParMooN.*

3. **References.** Wilbrandt et al., 2017.

1.8 SOFTube

1. **Author.** Martin Kanitsar martin.kanitsar@drei.at
2. **Documentation.** M. Hintermüller, K. Knall, M. Kanitsar. *Handbuck zur Software "Form- un Topologieoptimierung"*. 2016 [in German only].

   → I have translated the Handbook to English.

3. **Version.** Software-Version: 1612 (Dec, 2016).

   Remark 1.8.1 (Outdated)**.** *Only compatible with OpenFOAM 2.1.1 (release date: May 31, 2012). Cannot run on the latest version* `openfoam-dev` *(updated weekly to the cutting-edge developments) of OpenFOAM.*

4. **Components**. `SOFTube` is a group of OpenFOAM-based applications, which can be divided mainly into the following 3 subgroups:

   - `setupShapeGradientCCM`: usage: copy the default initial geometry from the folder `$FOAM_RUN/tutorials/InitialSGccm` to the current OpenFOAM case, set up the correct the folder structure for that case, and then read parameters from `parameter_sg` and then export to `$FOAM_RUN/<case directory>/system/fvSolution`

     Remark 1.8.2. *In my opinion, without considering* STAR-CCM+*, this step is redundant. OpenFOAM forces the users to set up the folder structure themselves and to change freely parameters in the dictionary* `fvSolution` *directly. Hence, if I develop my own OpenFOAM-based software, I will definitely skip this step to simplify the software elegantly and properly.*

   - **Topology Optimization subgroup.**
     - *Main executable OpenFOAM-based application*: `topoOpt`.
     - *Dependencies*: `cutSTL, generateFieldsFoam, splitSTLFoam, topoExtractSTL, topoOptCloseAll`.
     - *Usage*: For topology optimization, the largest possible initial geometry is considered, e.g., a sphere containing specified inlet(s) and outlet(s). Then `topoOpt` will optimize the shape by controlling porosity, i.e., removing counterproductive cells (see e.g., Carsten Othmer, Villiers, and Weller, 2007; C. Othmer, 2008), to obtain an acceptable initial geometry for shape optimization in the next stage.

       Remark 1.8.3. *I have not upgraded, and unintended to at the moment,* `topoOpt` *due to several doubts/concerns listed in Chap. 2.*

- **Shape Optimization subgroup.**
  - *Main executable OpenFOAM-based application*: `shapeGradientCCM`
  - *Dependencies*: `shapeGradientAddSTL, shapeGradientCloseAll, shapeGradientWall`.
  - Usage: Run in terminal (my favorite style)

    ```
    nguyen@escher-01:~/OpenFOAM/nguyen-v2006/run/Test-final> shapeGradientCCM 2>&1 | tee output
    ```

    for both the terminal output and a `output` file saving it (convenient to inspect later).

Remark 1.8.4. *I have upgraded Shape Optimization subgroup, with a lot of cleaning unused variables/headers and adjusting programming styles, to the latest version of OpenFOAM:* `OpenFOAM-v2006`*, which produces "similar" results to the old version* `OpenFOAM-2.1.1`*. "Similar" here is due to 1 of, or both, the following reasons:*

- *The new version of OpenFOAM computes more accurately than the old one (8-year difference).*
- *I upgraded based on the version of* `SOFTube` *which Martin Kanitsar sent me directly, not the current version running on* `escher-01`*.*

Chapter 2

Doubt

2.1 STAR-CCM+

- *Can the* Star-CCM+ *mesh generator and remeshing tool be replaced completely by OpenFOAM, or an open-source software?* → I need to dive in OpenFOAM mesh utilities.
- Star-CCM+ uses only discrete adjoint approach. *But its discrete adjoint approach is engineering-oriented or mathematics-oriented?*
- Each time running `shapeGradientCCM`, all available 5 Star-CCM+ licenses are used consecutively, since the option `-noexit` prevents the client-server from closing. *Is it possible to run* ***`shapeGradientCCM`*** *with checking out only 1 license?*
- *Should I drop out completely* Star-CCM+ *from now?* Should I use only Star-CCM+'s documentation for a reference to compare with OpenFOAM-based software, and stop upgrading/extending completely Star-CCM+ component in Martin Kanitsar's `SOFTube`?

2.2 Martin Kanitsar's `SOFTube`

1. `SOFTube/shapeGradientCCM` runs improperly and inaccurately.

   → I have not upgraded or touched deeply `SOFTube/topoOpt`. I have worked only with `setupShapeGradientCCM` (easy) and `shapeGradientCCM` [upgraded to the latest version of OpenFOAM].

2. `SOFTube` does not treat any boundary conditions for the primal equations and importantly, their corresponding *adjoint boundary conditions* for the adjoint equations either.

3. `SOFTube` uses discrete adjoint instead of continuous adjoint approach.

4. I have not coupled, and unintended to at the moment, OpenFOAM's standard solvers for incompressible flows with Martin Kanitsar's `SOFTube/shapeGradientCCM` yet since his solver is more "C++-oriented" than properly "OpenFOAM-oriented". And the real trouble for me is OpenFOAM uses FVM, meanwhile a lot of components of Martin Kanitsar's `SOFTube/shapeGradientCCM` used FEM instead. There will be a big compatibility issue if I continue to develop Martin Kanitsar's `SOFTube`.

5. Again, `SOFTube` is more C++-oriented than OpenFOAM-oriented. Although OpenFOAM is written by C++, but C++-orientated style has complicated `SOFTube` in unnecessary way. This can be solved by using properly build-in/default libraries of OpenFOAM.

   → I need more time to master OpenFOAM to restart the whole `SOFTube/shapeGradientCCM`.

Chapter 3

Plan

3.1 STAR-CCM+

1. `TheStevePortal/Documentation/`**Tutorials.** I have not run[1] in Sect. **Incompressible Flow**:
   (a) Porous Resistance: Isotropic Media
   (b) Porous Resistance: Orthotropic Media
   (c) Solution Recording and Playback: Vortex Shedding
   (d) Generalized Newtonian Fluid: Flow in a Static Mixer
   (e) Viscoelastic Flow: Basic Extrusion
   (f) Material Calibration: Curve Fitting
   (g) Non-Newtonian Model Parameters
   (h) Polymer Melt Extrusion: Film Casting

   → I will practice these if they seems relevant and necessary for the OpenFOAM-based counterpart.

3.2 OpenFOAM

1. Start to develop my own OpenFOAM-based software.
2. Not designed like in `SOFTube`, I can skip the whole Topology Optimization process and start directly with Shape Optimization part, by defining and generating an arbitrary initial shape by `blockMesh`[2]

   E.g., store the the following script with the right name `blockMeshDict` in the directory `$FOAM_RUN/airDuct/system/`:

```
/*--------------------------------*- C++ -*----------------------------------*\
=========                 |
\\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
 \\    /   O peration     | Website:  https://openfoam.org
  \\  /    A nd           | Version:  dev
   \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

delta   0.5;
```

[1] Cf. What I have run successfully in Sect. 1.1.
[2] An OpenFOAM built-in utility for generating OpenFOAM-usable meshes.

```
overSqrt2   0.70710678118;
minusOverSqrt2 -0.70710678118;

vertices
(
// inlet
(0 -1 4) // 0
(1  0 4) // 1
(0  1 4) // 2
(-1 0 4) // 3

// cut-off inlet cross surface
(0 -1 3.5) // 4
(1  0 3.5) // 5
(0  1 3.5) // 6
(-1 0 3.5) // 7

// fixed inlet cross surface
(0 -1 3) // 8
(1  0 3) // 9
(0  1 3) // 10
(-1 0 3) // 11

// before bending cross surface
(0 -1 2) // 12
(1  0 2) // 13
(0  1 2) // 14
(-1 0 2) // 15

// after bending cross surface
(0  2 -1) // 16
(1  2  0) // 17
(0  2  1) // 18
(-1 2  0) // 19

// fixed outlet  cross surface
(0  3 -1) // 20
(1  3  0) // 21
(0  3  1) // 22
(-1 3  0) // 23

// cut-off outlet cross surface
(0  3.5 -1) // 24
(1  3.5  0) // 25
(0  3.5  1) // 26
(-1 3.5  0) // 27

// outlet
(0  4 -1) // 28
(1  4  0) // 29
(0  4  1) // 30
(-1 4  0) // 31
);

blocks
(
// cut-off inlet block
hex (4 5 6 7 0 1 2 3) (10 10 10) simpleGrading (1 1 1)
```

```
// fixed inlet\cut-off inlet block
hex (8 9 10 11 4 5 6 7) (10 10 10) simpleGrading (1 1 1)

// before bending\fixed inlet block
hex (12 13 14 15 8 9 10 11) (10 10 10) simpleGrading (1 1 1)

// bending
hex (16 17 18 19 12 13 14 15) (10 10 10) simpleGrading (1 1 1)

// after bending\fixed outlet block
hex (19 18 17 16 23 22 21 20) (10 10 10) simpleGrading (1 1 1)

// fixed outlet\cut-off outlet block
hex (23 22 21 20 27 26 25 24) (10 10 10) simpleGrading (1 1 1)

// cut-off outlet block
hex (27 26 25 24 31 30 29 28) (10 10 10) simpleGrading (1 1 1)
);

edges
(
// inlet
arc 0 1 (0.70710678118  -0.70710678118 4)
arc 1 2 (0.70710678118   0.70710678118 4)
arc 2 3 (-0.70710678118  0.70710678118 4)
arc 3 0 (-0.70710678118 -0.70710678118 4)

// cut-off inlet cross surface
arc 4 5 (0.70710678118  -0.70710678118 3.5)
arc 5 6 (0.70710678118   0.70710678118 3.5)
arc 6 7 (-0.70710678118  0.70710678118 3.5)
arc 7 4 (-0.70710678118 -0.70710678118 3.5)

// fixed inlet cross surface
arc 8 9   (0.70710678118   -0.70710678118 3)
arc 9 10  (0.70710678118    0.70710678118 3)
arc 10 11 (-0.70710678118   0.70710678118 3)
arc 11 8  (-0.70710678118  -0.70710678118 3)

// before bending cross surface
arc 12 13 (0.70710678118  -0.70710678118 2)
arc 13 14 (0.70710678118   0.70710678118 2)
arc 14 15 (-0.70710678118  0.70710678118 2)
arc 15 12 (-0.70710678118 -0.70710678118 2)

// after bending cross surface
arc 19 18 (-0.70710678118  2   0.70710678118)
arc 18 17 (0.70710678118   2   0.70710678118)
arc 17 16 (0.70710678118   2  -0.70710678118)
arc 16 19 (-0.70710678118  2  -0.70710678118)

// fixed outlet cross surface
arc 23 22 (-0.70710678118  3  0.70710678118)
arc 22 21 (0.70710678118   3  0.70710678118)
arc 21 20 (0.70710678118   3 -0.70710678118)
arc 20 23 (-0.70710678118  3 -0.70710678118)

// cut-off outlet cross surface
arc 27 26 (-0.70710678118  3.5  0.70710678118)
```

```
arc 26 25 (0.70710678118   3.5  0.70710678118)
arc 25 24 (0.70710678118   3.5 -0.70710678118)
arc 24 27 (-0.70710678118  3.5 -0.70710678118)

// outlet
arc 31 30 (-0.70710678118  4  0.70710678118)
arc 30 29 (0.70710678118   4  0.70710678118)
arc 29 28 (0.70710678118   4 -0.70710678118)
arc 28 31 (-0.70710678118  4 -0.70710678118)

// cut-off inlet block
line 0 4
line 1 5
line 2 6
line 3 7

// fixed inlet\cut-off inlet block
line 4 8
line 5 9
line 6 10
line 7 11

// before bending\fixed inlet block
line 8 12
line 9 13
line 10 14
line 11 15

// bending

// - Use straight lines for bending
/*
line 12 16
line 13 17
line 14 18
line 15 19
*/
// - Use curves for bending
arc 12 16 (0 -0.12132034356 -0.12132034356) // 2 - 3/sqrt(2)
arc 13 17 (1  0.58578643762  0.58578643762) // 2 - 2/sqrt(2)
arc 14 18 (0  1.29289321881  1.29289321881) // 2 - 1/sqrt(2)
arc 15 19 (-1 0.58578643762  0.58578643762) // 2 - 2/sqrt(2)

// after bending\fixed outlet block
line 16 20
line 17 21
line 18 22
line 19 23

// fixed outlet\cut-off outlet block
line 20 24
line 21 25
line 22 26
line 23 27

// cut-off outlet block
line 24 28
line 25 29
line 26 30
```

```
line 27 31
);

boundary
(
inlet
{
    type    patch;
    faces
    (
    (0 1 2 3)
    );
}

cutOffInlet
{
    type wall;
    faces
    (
    (4 5 1 0)
    (5 6 2 1)
    (6 7 3 2)
    (7 4 0 3)
    );
}

fixedWallInletExcludeCutOffInlet
{
    type wall;
    faces
    (
    (8 9 5 4)
    (9 10 6 5)
    (10 11 7 6)
    (11 8 4 7)
    );
}

movingWall
{
    type wall;
    faces
    (
    // before bending\fixed inlet block
    (12 13 9 8)
    (13 14 10 9)
    (14 15 11 10)
    (15 12 8 11)

    // bending
    (16 17 13 12)
    (17 18 14 13)
    (18 19 15 14)
    (19 16 12 15)

    // after bending\fixed outlet block
    (17 16 20 21)
    (18 17 21 22)
    (19 18 22 23)
```

```
    (16 19 23 20)
    );
}

fixedWallOutletExcludeCutOffOutlet
{
    type wall;
    faces
    (
    (21 20 24 25)
    (22 21 25 26)
    (23 22 26 27)
    (20 23 27 24)
    );
}

cutOffOutlet
{
    type wall;
    faces
    (
    (25 24 28 29)
    (26 25 29 30)
    (27 26 30 31)
    (24 27 31 28)
    );
}

outlet
{
    type    patch;
    faces
    (
    (31 30 29 28)
    );
}
);

mergePatchPairs
(
);


// ************************************************************************* //
```

Execute in the directory `$FOAM_RUN/airDuct/` (1 level up instead) the OpenFOAM built-in utility `blockMesh` to generate the mesh for the initial duct defined by the script `blockMeshDict`:

```
nguyen@fg8nb1:~/OpenFOAM/nguyen-dev/run/airDuct$ blockMesh
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  dev
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
Build  : dev-d35103fe2221
```

```
Exec   : blockMesh
Date   : Jan 31 2021
Time   : 14:25:04
Host   : "fg8nb1"
PID    : 24214
I/O    : uncollated
Case   : /home/nguyen/OpenFOAM/nguyen-dev/run/airDuct
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Create time

Deleting polyMesh directory
    "/home/nguyen/OpenFOAM/nguyen-dev/run/airDuct/constant/polyMesh"
Creating block mesh from
    "system/blockMeshDict"
Creating block edges
No non-planar block faces defined
Creating topology blocks
Creating topology patches

Creating block mesh topology

Check topology

        Basic statistics
                Number of internal faces : 6
                Number of boundary faces : 30
                Number of defined boundary faces : 30
                Number of undefined boundary faces : 0
        Checking patch -> block consistency

Creating block offsets
Creating merge list .

Creating polyMesh from blockMesh
Creating patches
Creating cells
Creating points with scale 1
    Block 0 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.05
    Block 1 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.05 .. 0.05
    Block 2 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.1 .. 0.1
    Block 3 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.157839 .. 0.151568 0.163291 .. 0.156568
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.157839 .. 0.151568 0.163291 .. 0.156568
        k : 0.470755 .. 0.470755 0.464562 .. 0.464562 0.464562 .. 0.464562 0.45533 .. 0.45533
    Block 4 cell size :
```

```
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.1 .. 0.1
    Block 5 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.05 .. 0.05
    Block 6 cell size :
        i : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        j : 0.156918 .. 0.156918 0.161753 .. 0.161753 0.156918 .. 0.156918 0.161753 .. 0.161753
        k : 0.05

Writing polyMesh
----------------
Mesh Information
----------------
  boundingBox: (-1 -1 -1) (1 4 4)
  nPoints: 8591
  nCells: 7000
  nFaces: 22500
  nInternalFaces: 19500
----------------
Patches
----------------
  patch 0 (start: 19500 size: 100) name: inlet
  patch 1 (start: 19600 size: 400) name: cutOffInlet
  patch 2 (start: 20000 size: 400) name: fixedWallInletExcludeCutOffInlet
  patch 3 (start: 20400 size: 1200) name: movingWall
  patch 4 (start: 21600 size: 400) name: fixedWallOutletExcludeCutOffOutlet
  patch 5 (start: 22000 size: 400) name: cutOffOutlet
  patch 6 (start: 22400 size: 100) name: outlet

End
```

then check the quality of the mesh generated by the OpenFOAM built-in utility `checkMesh`:

```
nguyen@fg8nb1:~/OpenFOAM/nguyen-dev/run/airDuct$ checkMesh
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  dev
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
Build  : dev-d35103fe2221
Exec   : checkMesh
Date   : Jan 31 2021
Time   : 14:27:19
Host   : "fg8nb1"
PID    : 24265
I/O    : uncollated
Case   : /home/nguyen/OpenFOAM/nguyen-dev/run/airDuct
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
Create time

Create polyMesh for time = 0

Time = 0

Mesh stats
    points:           8591
    faces:            22500
    internal faces:   19500
    cells:            7000
    faces per cell:   6
    boundary patches: 7
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     7000
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
    Patch               Faces    Points   Surface topology
    inlet               100      121      ok (non-closed singly connected)
    cutOffInlet         400      440      ok (non-closed singly connected)
    fixedWallInletExcludeCutOffInlet400      440      ok (non-closed singly connected)
    movingWall          1200     1240     ok (non-closed singly connected)
    fixedWallOutletExcludeCutOffOutlet400      440      ok (non-closed singly connected)
    cutOffOutlet        400      440      ok (non-closed singly connected)
    outlet              100      121      ok (non-closed singly connected)

Checking geometry...
    Overall domain bounding box (-1 -1 -1) (1 4 4)
    Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
    Mesh has 3 solution (non-empty) directions (1 1 1)
    Boundary openness (-2.12757e-17 3.35212e-17 2.83562e-17) OK.
    Max cell openness = 2.15006e-16 OK.
    Max aspect ratio = 10.1397 OK.
    Minimum face area = 0.00747764. Maximum face area = 0.0794287.  Face area magnitudes OK.
    Min volume = 0.000418055. Max volume = 0.0155085.  Total volume = 21.9584.  Cell volumes OK.
    Mesh non-orthogonality Max: 61.503 average: 18.6781
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.97808 OK.
    Coupled point location match (average 0) OK.
```

```
Mesh OK.

End
```

View the generated mesh of the initial duct geometry via ParaView (with different modes):

```
nguyen@fg8nb1:~/OpenFOAM/nguyen-dev/run/airDuct$ paraview
```

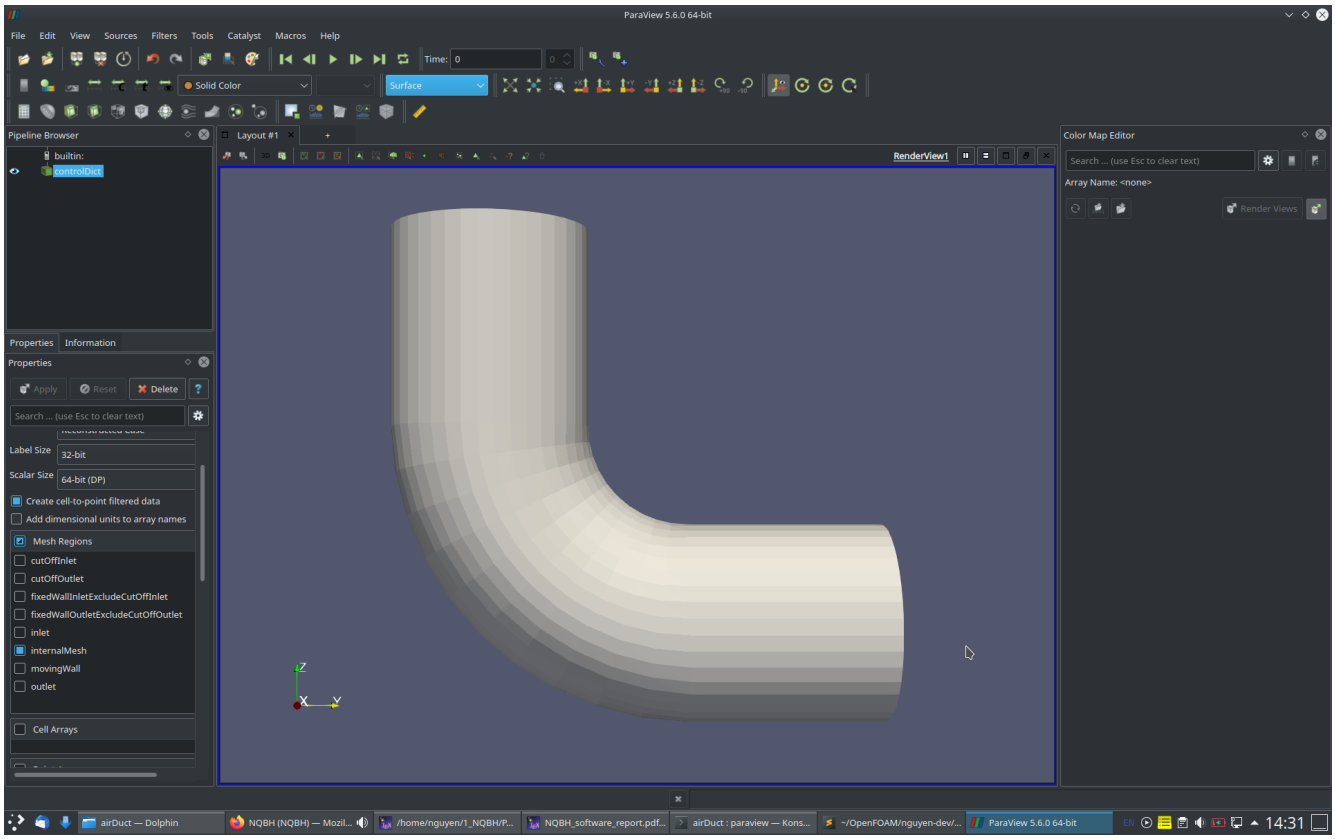

Figure 3.1: Optimized S-bend shape with the deformation field illustrated.

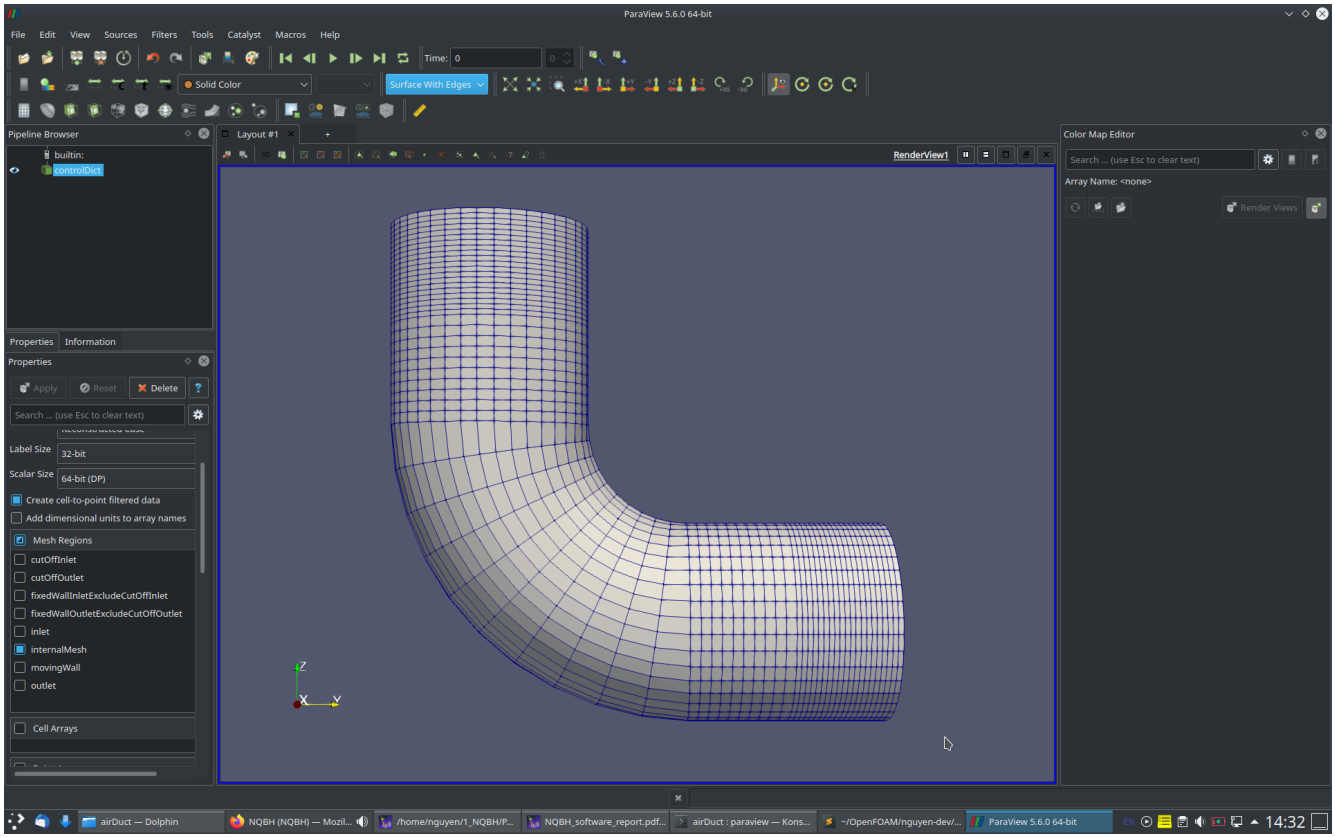

Figure 3.2: Optimized S-bend shape with the deformation field illustrated.

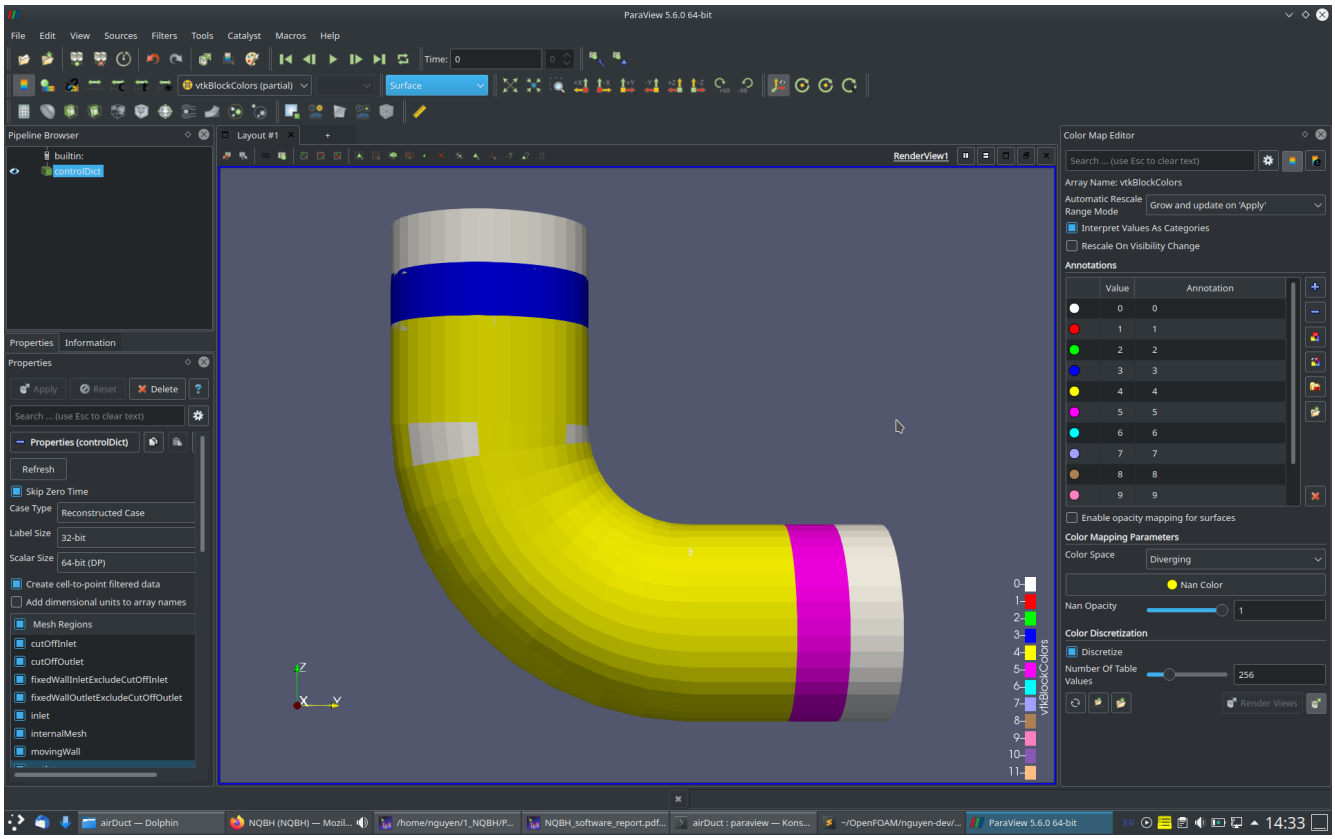

Figure 3.3: Optimized S-bend shape with the deformation field illustrated.

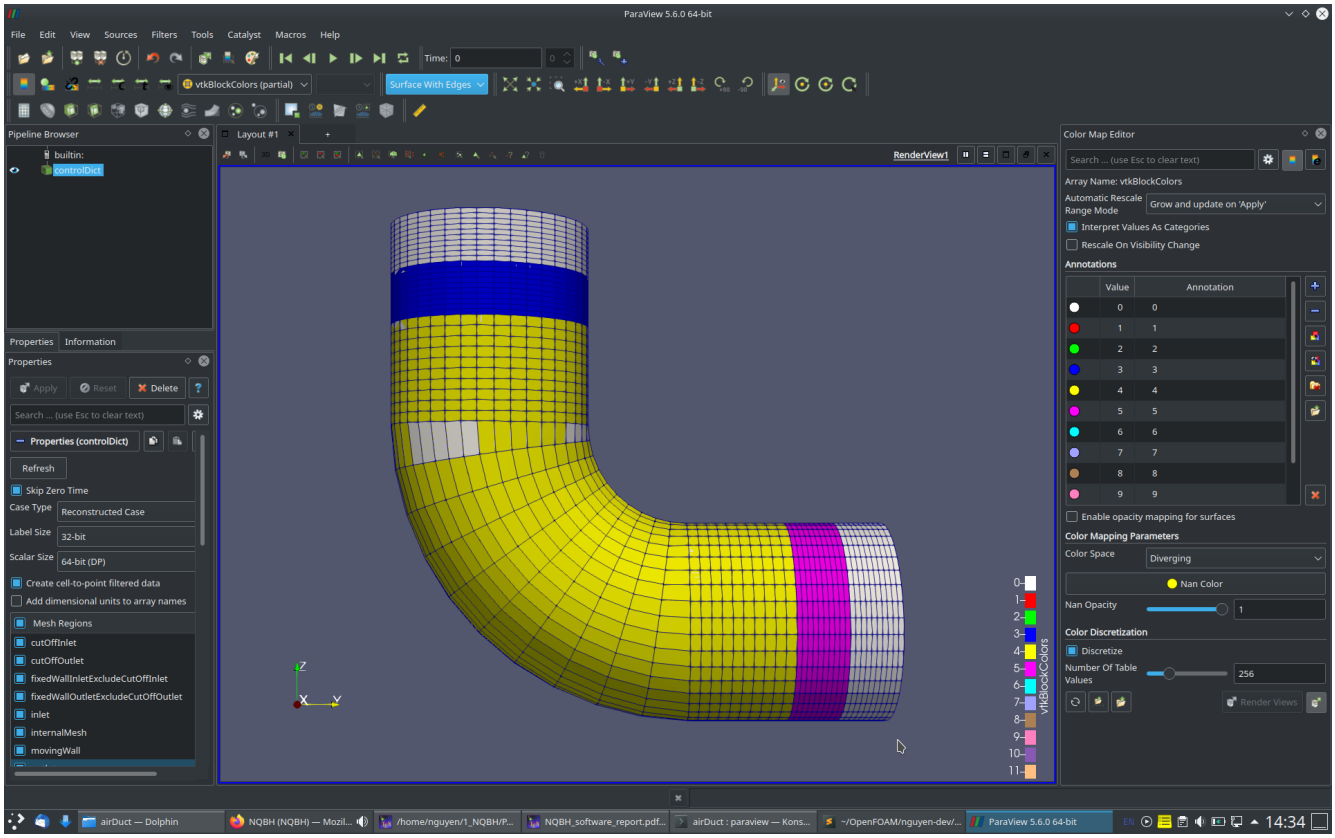

Figure 3.4: Optimized S-bend shape with the deformation field illustrated.

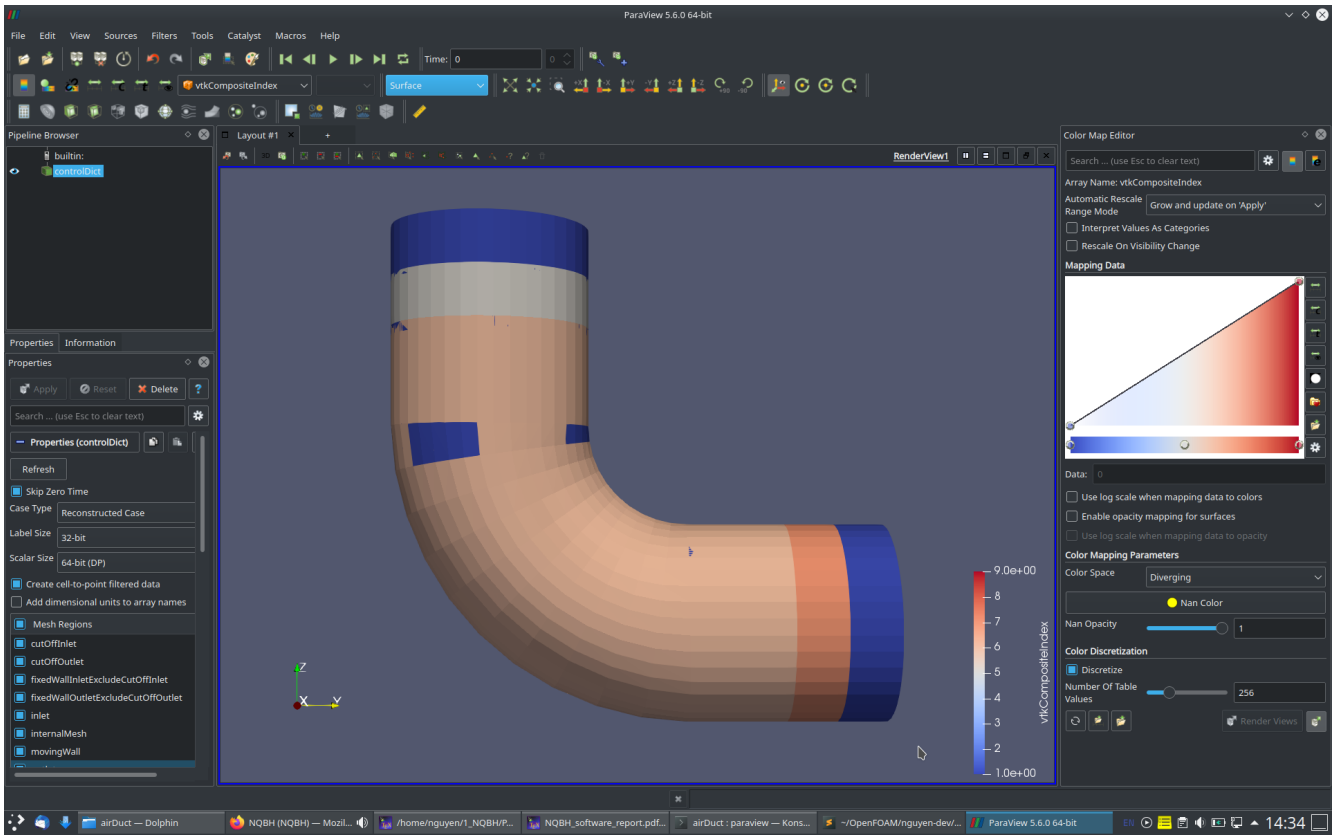

Figure 3.5: Optimized S-bend shape with the deformation field illustrated.

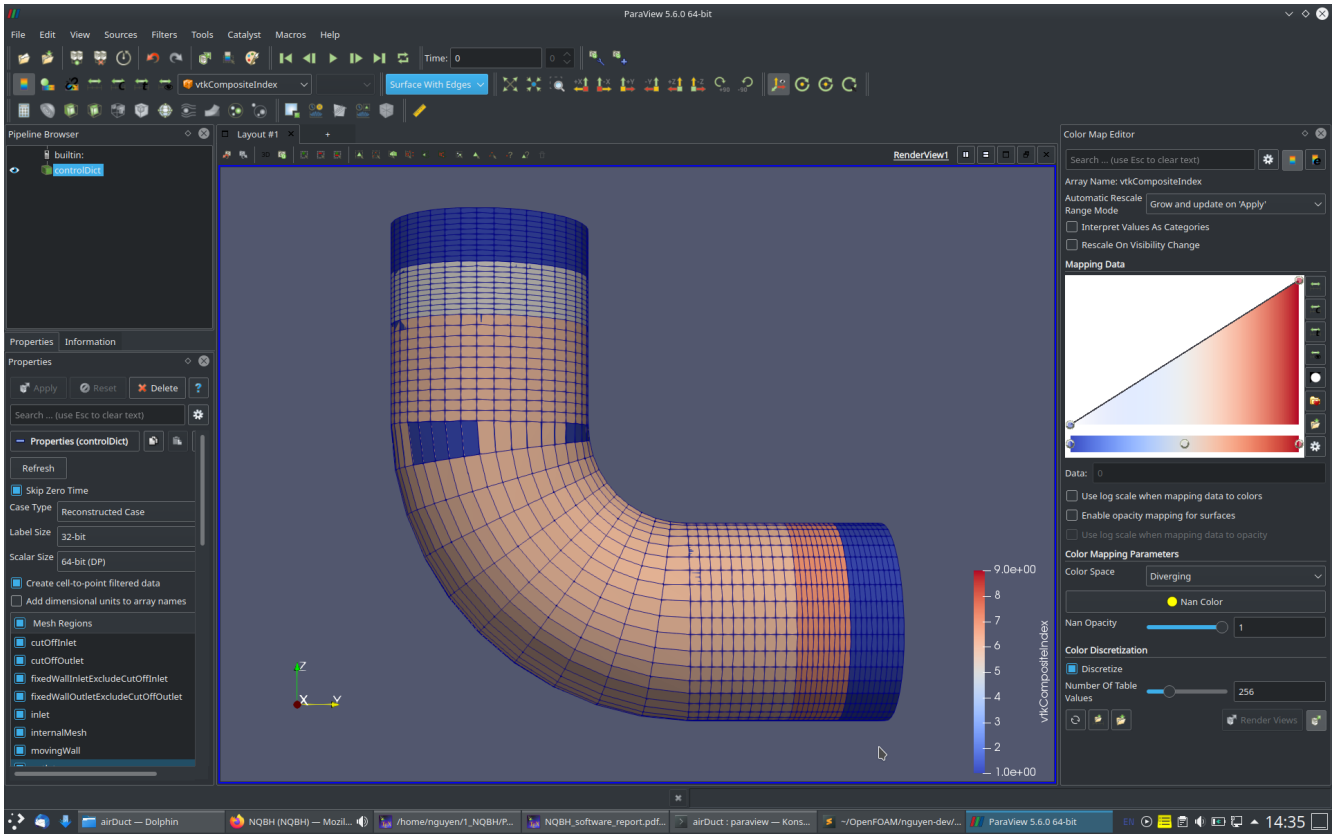

Figure 3.6: Optimized S-bend shape with the deformation field illustrated.

Remark 3.2.1 (Issue)**.** *I really do not understand why there is some strange elements colored inconsistently in the middle part of the duct geometry although I have carefully written and checked my* `blockMeshDict` *script. I hope this is just a viewing issue with ParaView. I will keep investigating on this.*

3. **Next step.** I will dive into OpenFOAM solvers in the group of `incompressible` solvers, with working parallelly on the *adjoint boundary conditions*, *boundary layers*, and *wall laws* on the theoretical side.

3.3 FEniCS

1. I have not run the following FEniCS tutorial examples:
   (a) **Reaction system.**
   (b) **Extended Poisson's equation.**
   (c) **Magnetostatics.**
   (d) **Poisson's solver.**
   (e) **Boxfield.**

   → I will practice these if they seems relevant and necessary for the FEM part. At this moment, I want to focus on FVM for OpenFOAM instead.

Bibliography

Dokken, Jørgen S. (2020). "Automatic shape derivatives for transient PDEs in FEniCS and Firedrake". In: URL: https://arxiv.org/abs/2001.10058.

Langtangen, Hans Petter and Anders Logg (2016). *Solving PDEs in Python*. Vol. 3. Simula SpringerBriefs on Computing. The FEniCS tutorial I. Springer, Cham, pp. ix+148. ISBN: 978-3-319-52461-0; 978-3-319-52462-7. DOI: 10.1007/978-3-319-52462-7. URL: https://doi.org/10.1007/978-3-319-52462-7.

Othmer, C. (2008). "A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows". In: *Internat. J. Numer. Methods Fluids* 58.8, pp. 861–877. ISSN: 0271-2091. DOI: 10.1002/fld.1770. URL: https://doi.org/10.1002/fld.1770.

Othmer, Carsten, Eugene de Villiers, and Henry Weller (2007). "Implementation of a Continuous Adjoint for Topology Optimization of Ducted Flows". In: *18th AIAA Computational Fluid Dynamics Conference*, pp. 1–7. DOI: 10.2514/6.2007-3947. URL: https://doi.org/10.2514/6.2007-3947.

Paganini, Alberto and Florian Wechsung (2020a). "Fireshape Documentation, Release 0.0.1". In: pp. ii+31. URL: https://fireshape.readthedocs.io/en/latest/index.html.

— (2020b). "Fireshape: a shape optimization toolbox for Firedrake". In: *arXiv*. DOI: 10.17028/RD.LBORO.12102531.V1. URL: https://arxiv.org/abs/2005.07264.

Wilbrandt, Ulrich et al. (2017). "ParMooN—a modernized program package based on mapped finite elements". In: *Comput. Math. Appl.* 74.1, pp. 74–88. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2016.12.020. URL: https://doi.org/10.1016/j.camwa.2016.12.020.