**Solving PDEs with Python and Fenics Libraries.**

**Part 2**

**Linear and Nonlinear Poisson PDE solutions**

**Poisson with neumann boundary conditions.**

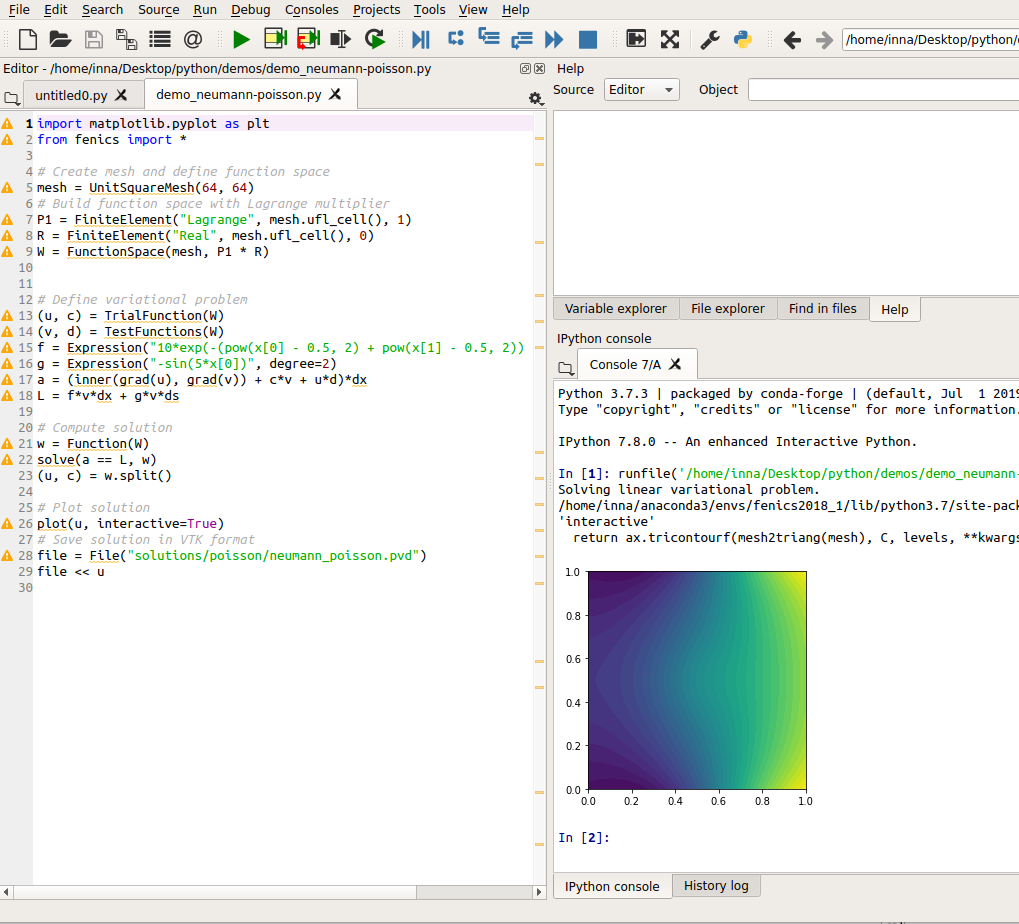
div grad u(x, y) = f(x, y)

du/dn(x, y) = -sin(5\*x)

f(x, y) = 10\*exp(-((x - 0.5)^2 + (y - 0.5)^2) / 0.02)

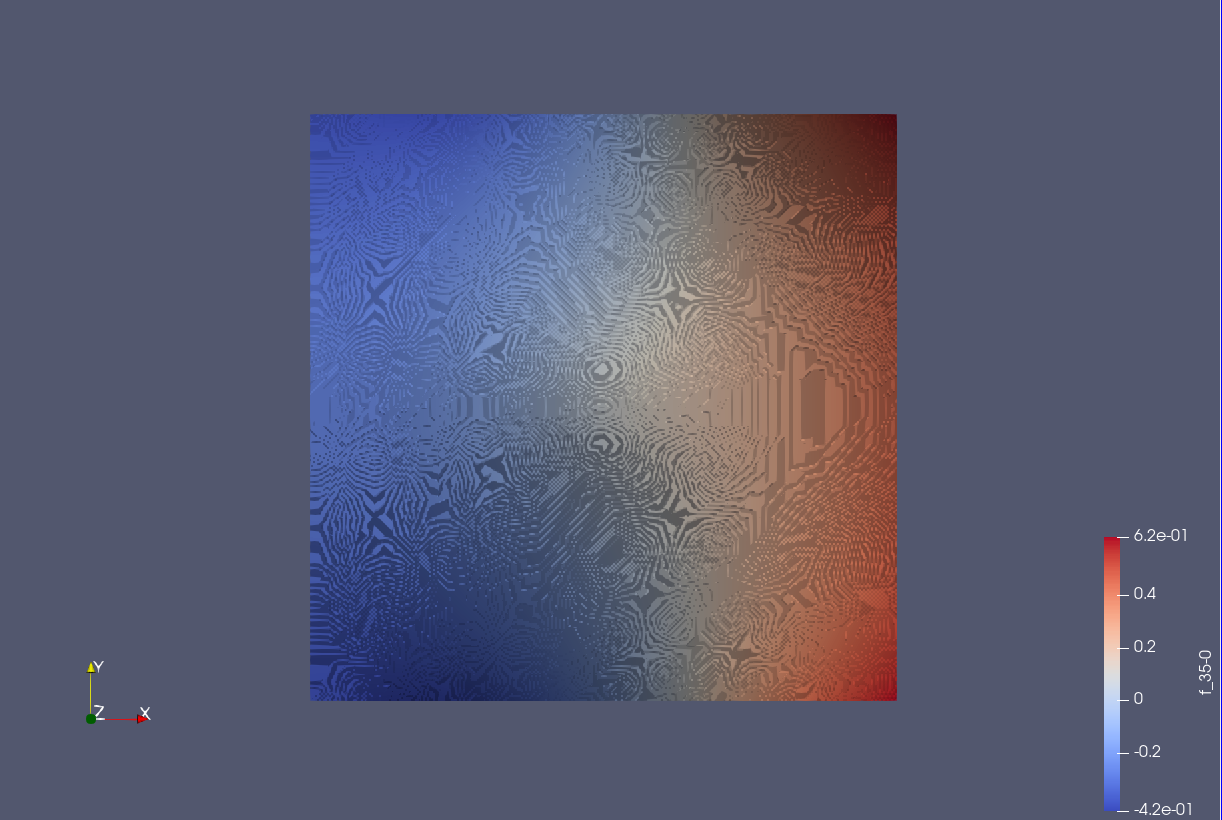
An addition constraint is thus required, for instance int u = 0

As additional unknown constant c is used

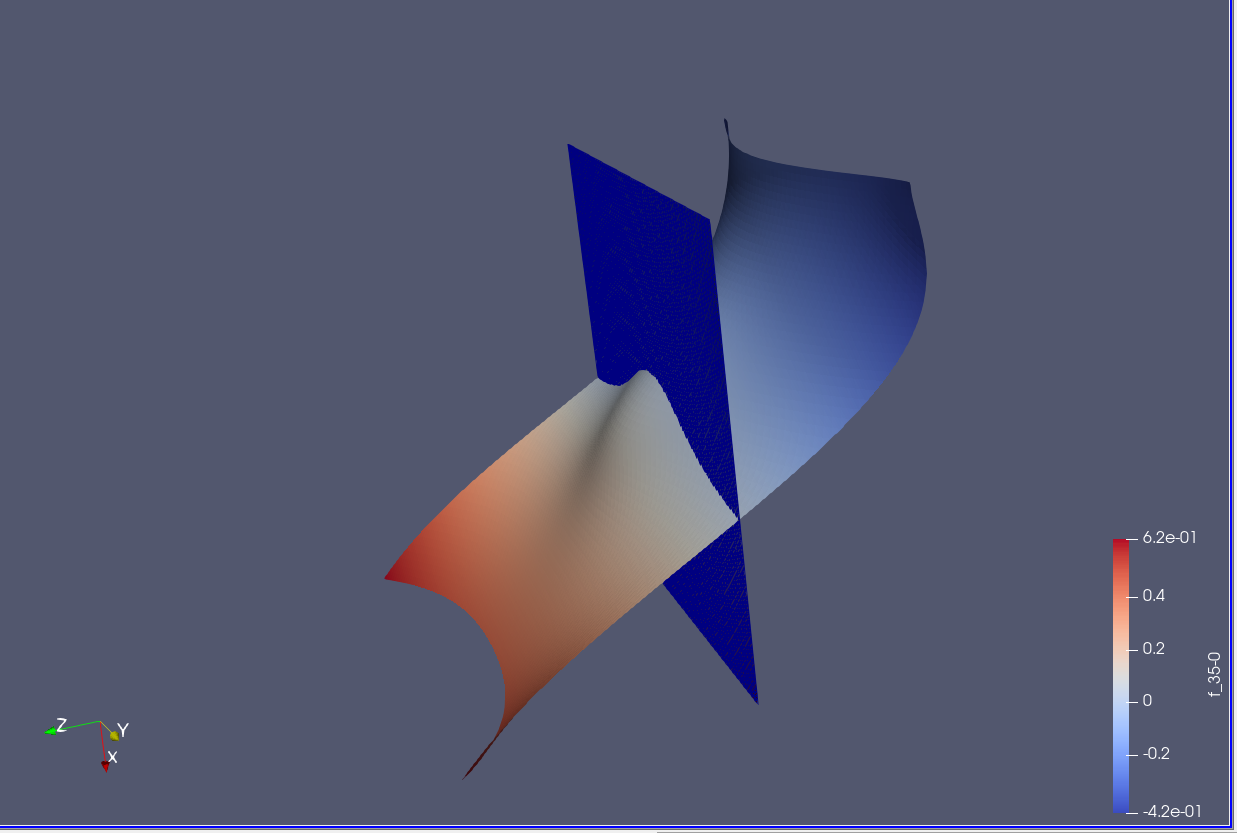


Paraview images

Paraview files



Parview 3D



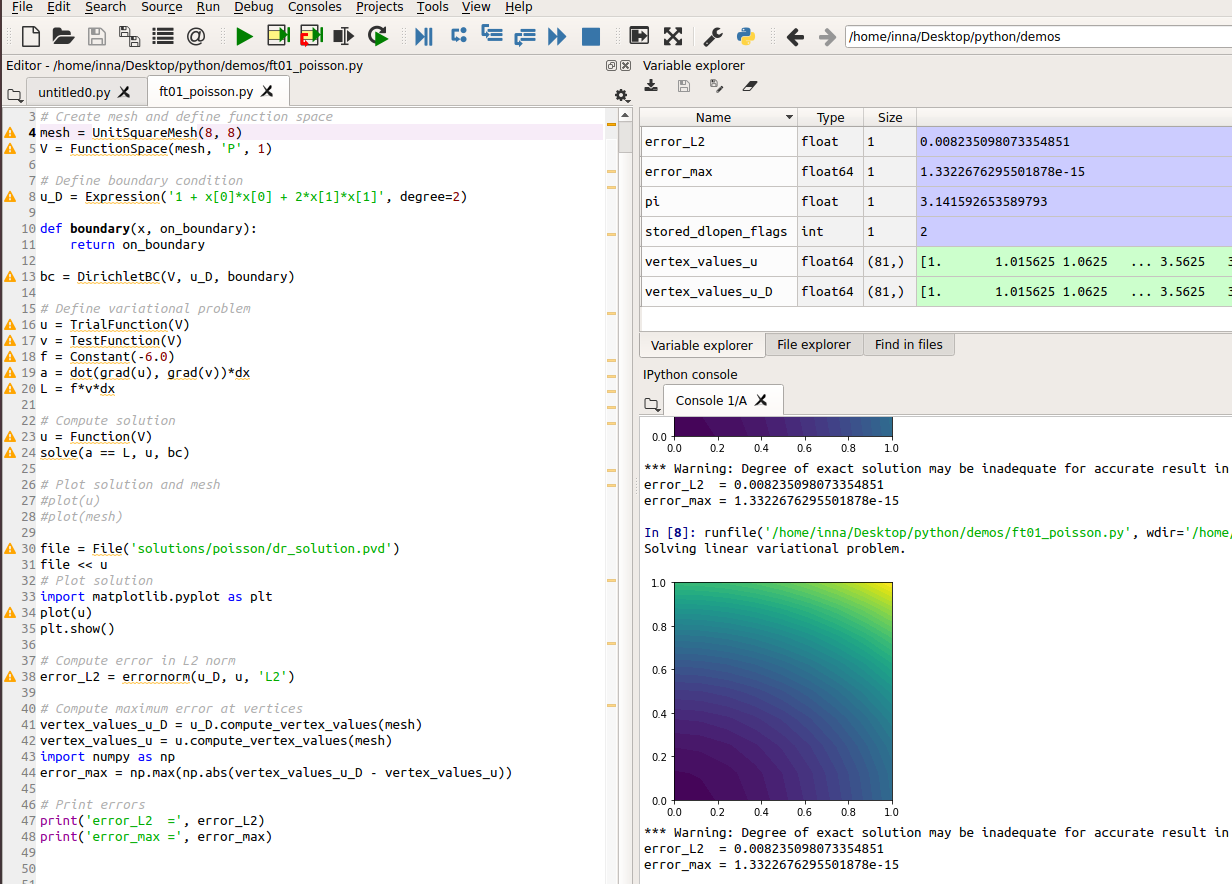
**Poisson equation with Dirichlet conditions.**

-Laplace(u) = f in the unit square

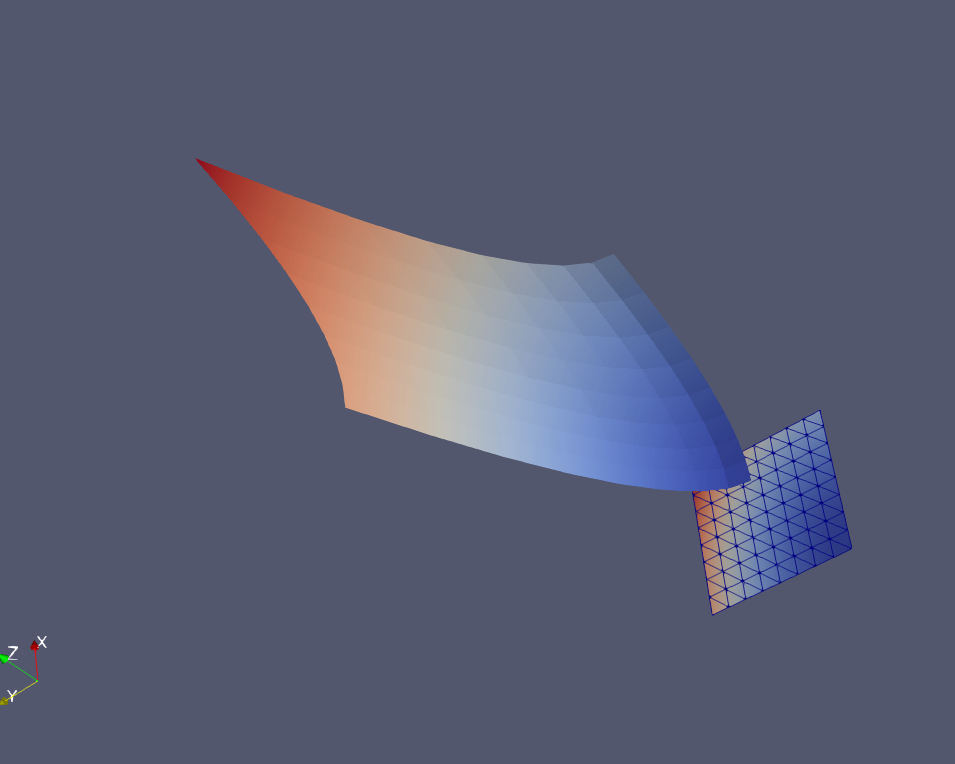
u = u\_D on the boundary

u\_D = 1 + x^2 + 2y^2

f = -6



Paraview files



**Mixed Poisson equation**

sigma - grad(u) = 0

div(sigma) = f

The corresponding weak (variational problem)

<sigma, tau> + <div(tau), u> = 0 for all t ,u

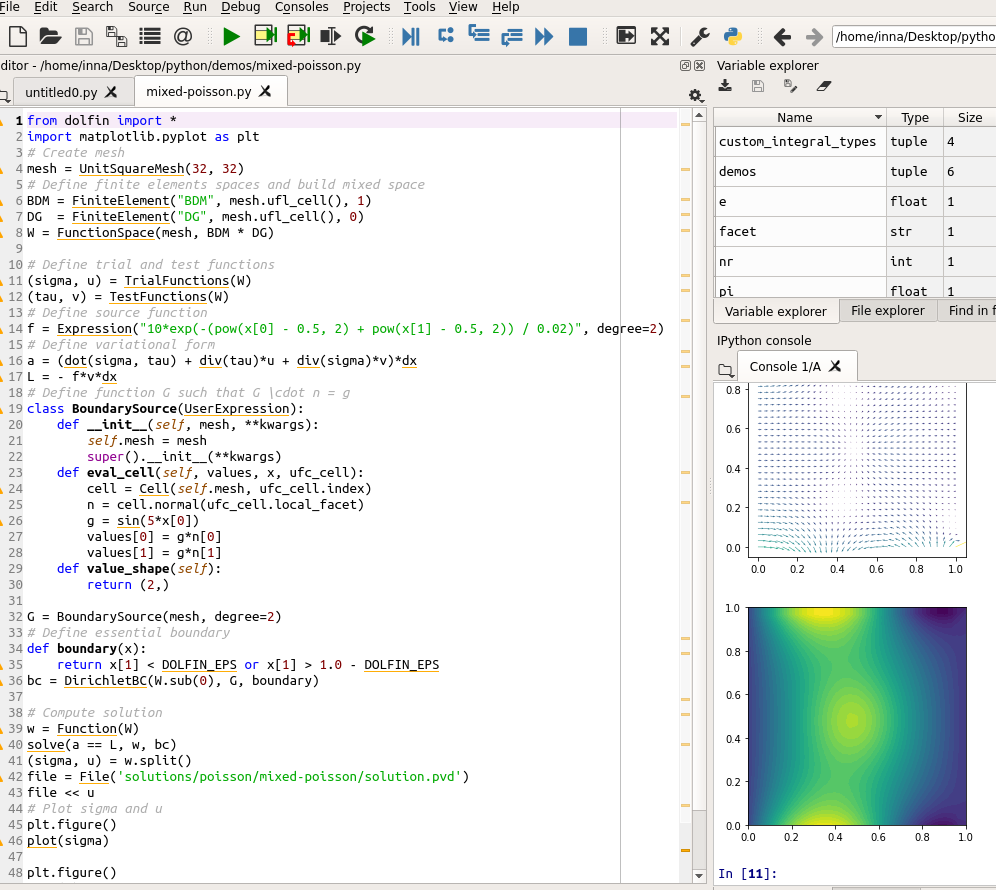
<div(sigma), v> = <f, v> for all v

is solved using BDM (Brezzi-Douglas-Marini) elements of degree k for

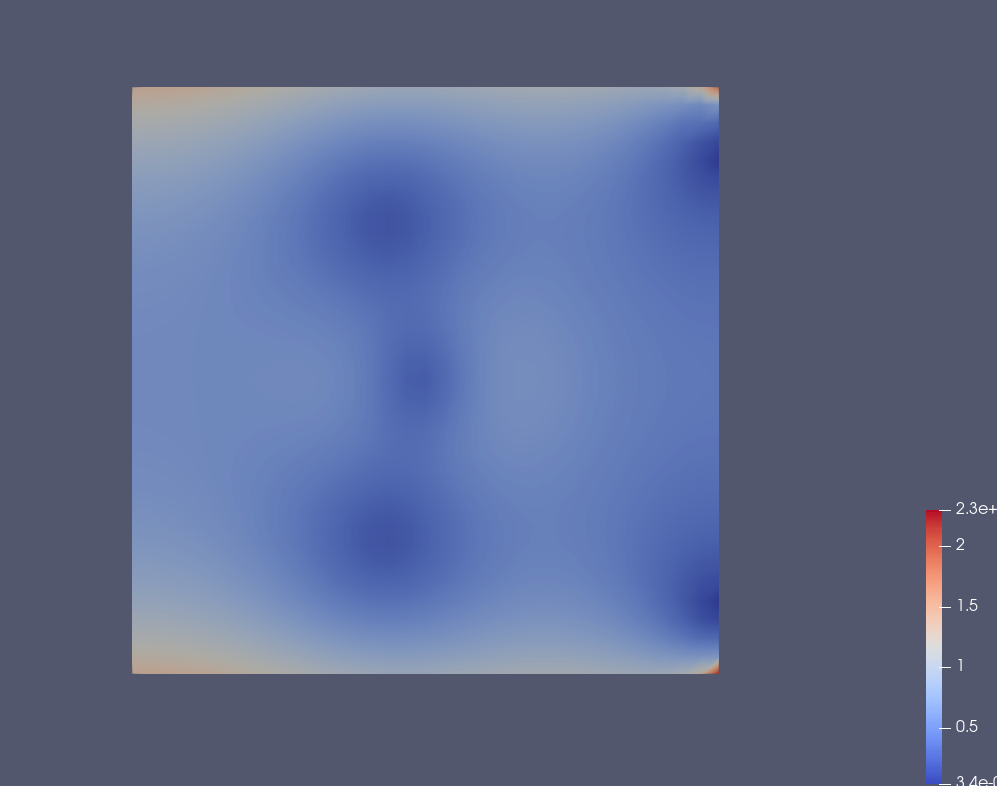
(sigma, tau) and DG (discontinuous Galerkin) elements of degree k - 1

for (u, v).

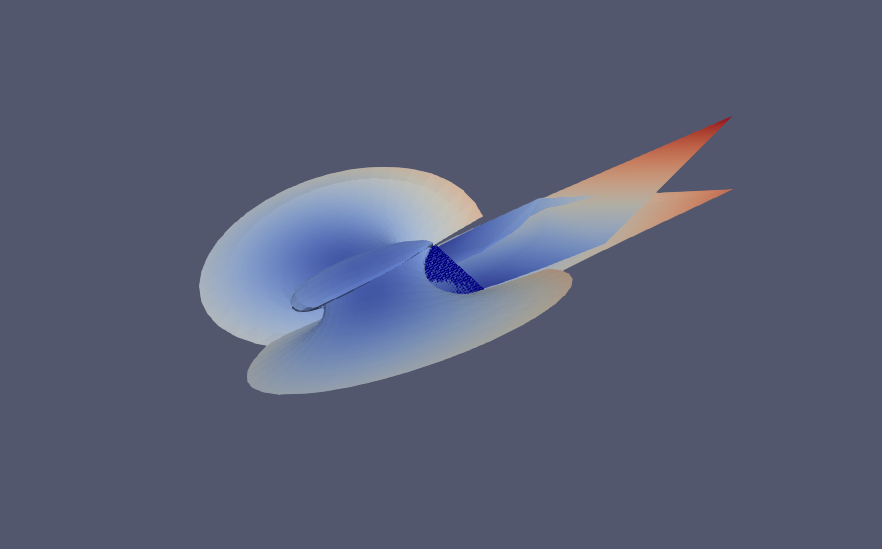
Original implementation: ../cpp/main.cpp by Anders Logg and Marie Rognes



Paraview



3D



**Non-Linear Poisson on the unit square**

div (1 + u^2) grad u(x, y) = f(x, y)

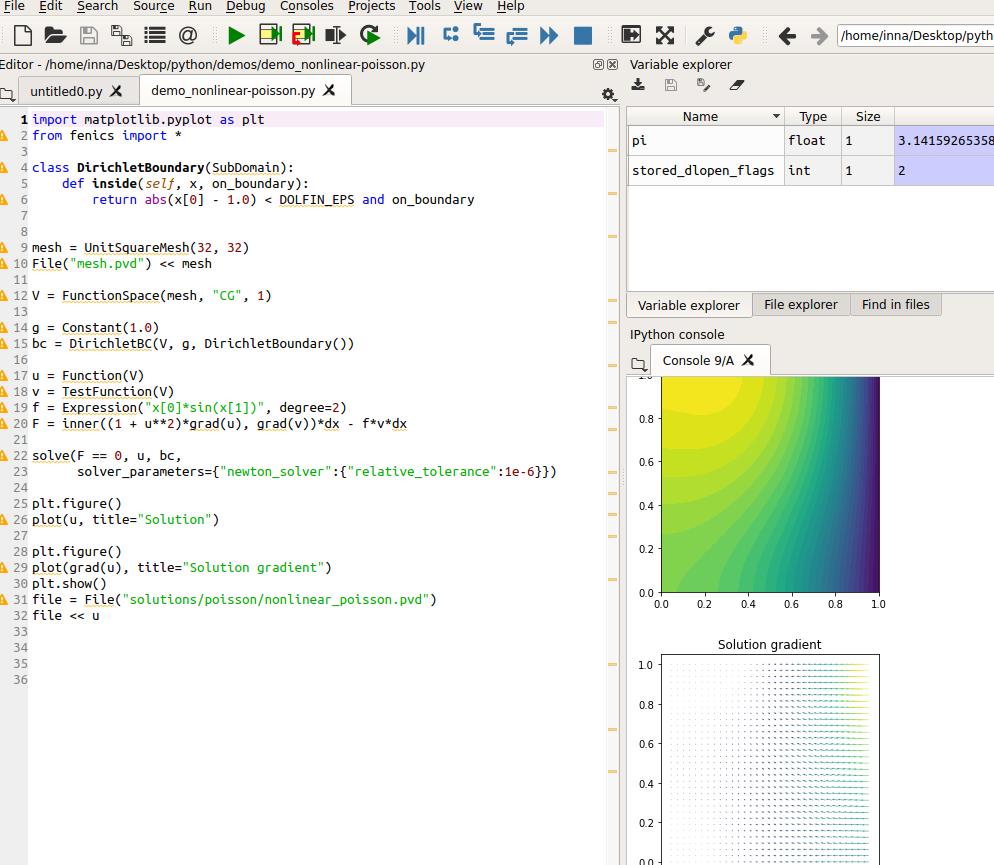
f(x, y) = x\*sin(y)

u(x, y) = 1 for x = 0

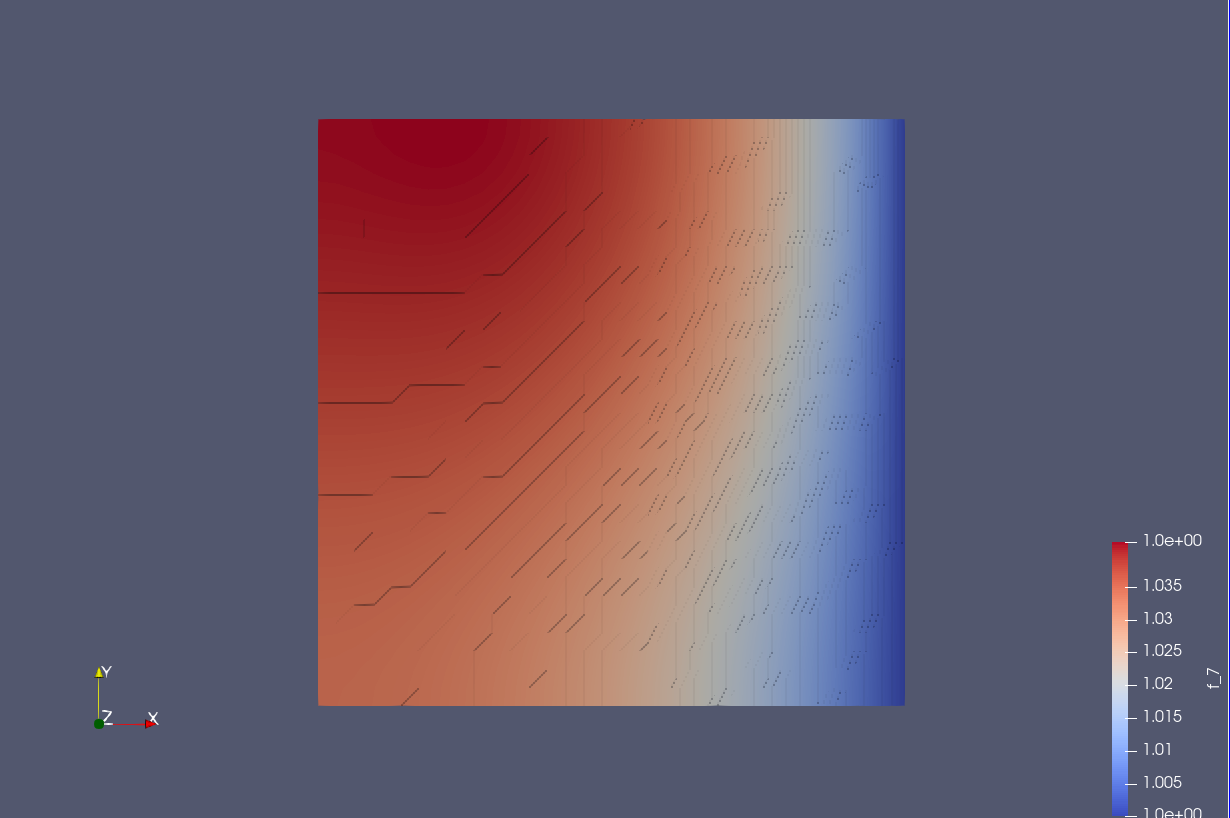
du/dn(x, y) = 0 otherwise

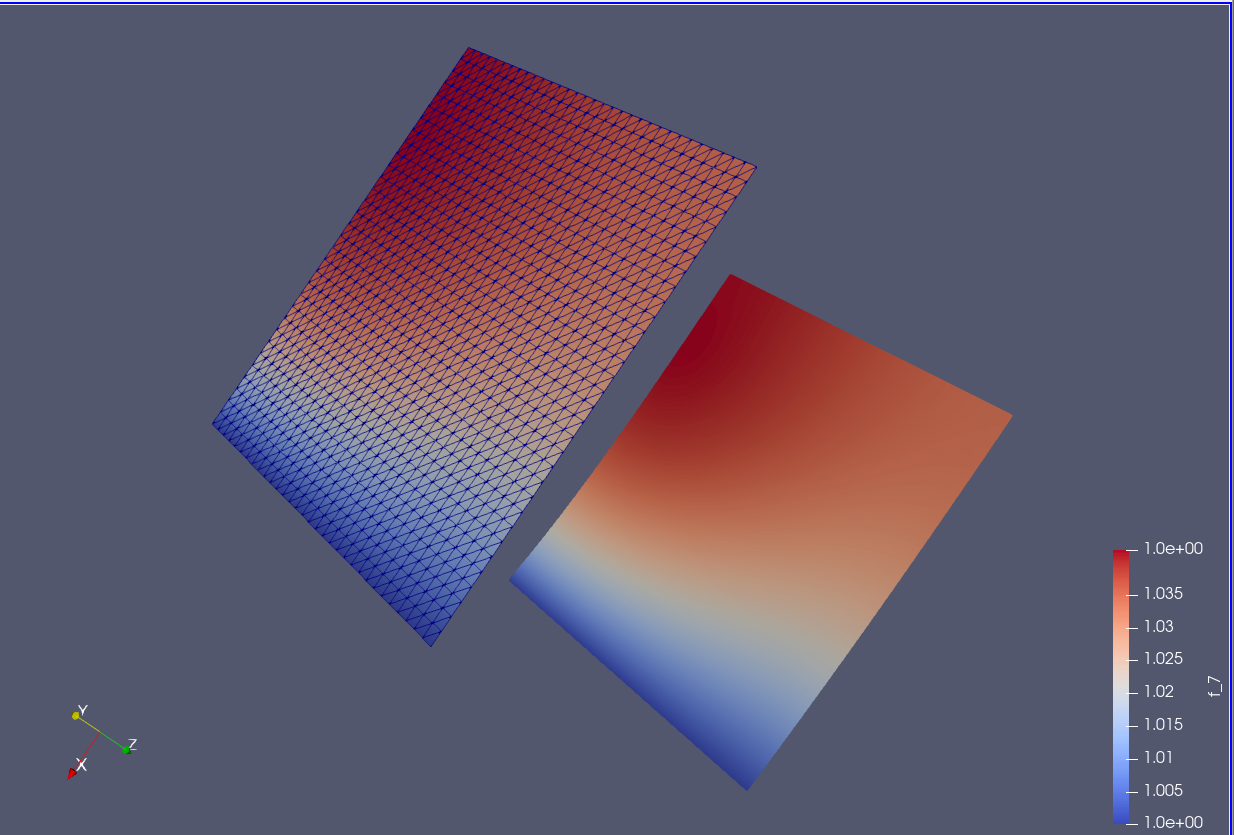
Or it can be defined as a variational problem

F(u) = ((1 + u^2)\*grad(u), grad(v)) - (f, v) = 0



Paraview images

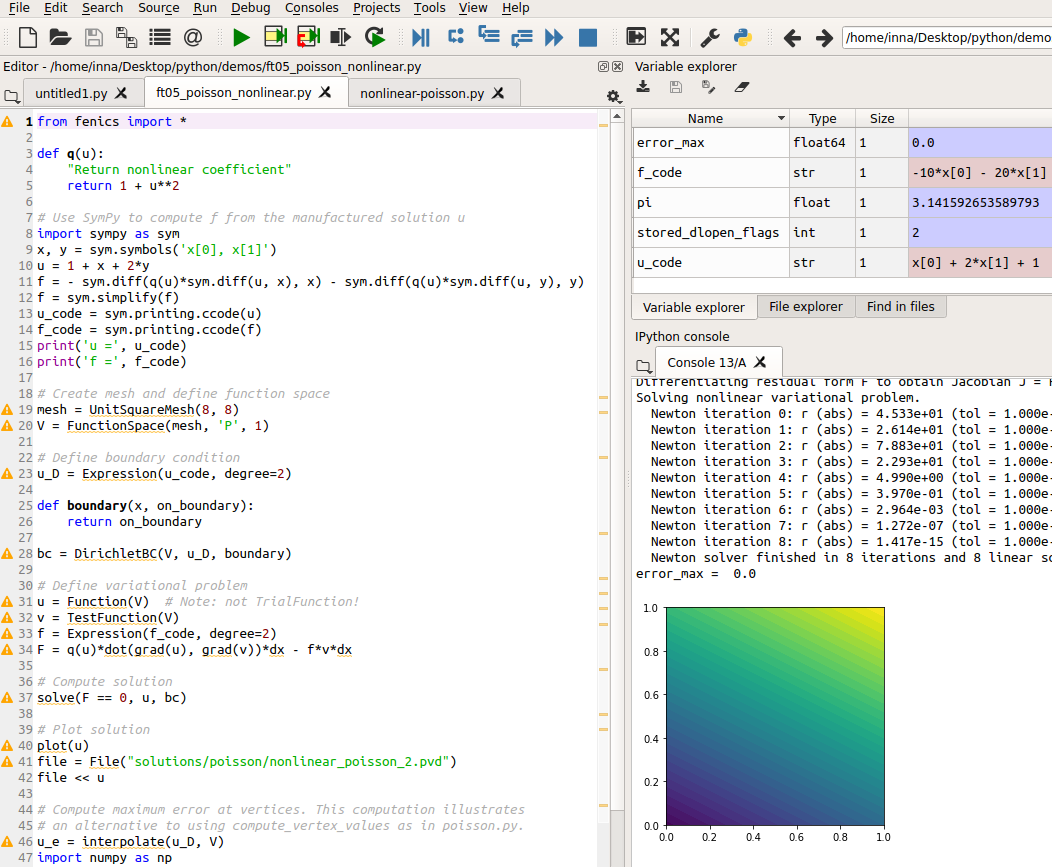




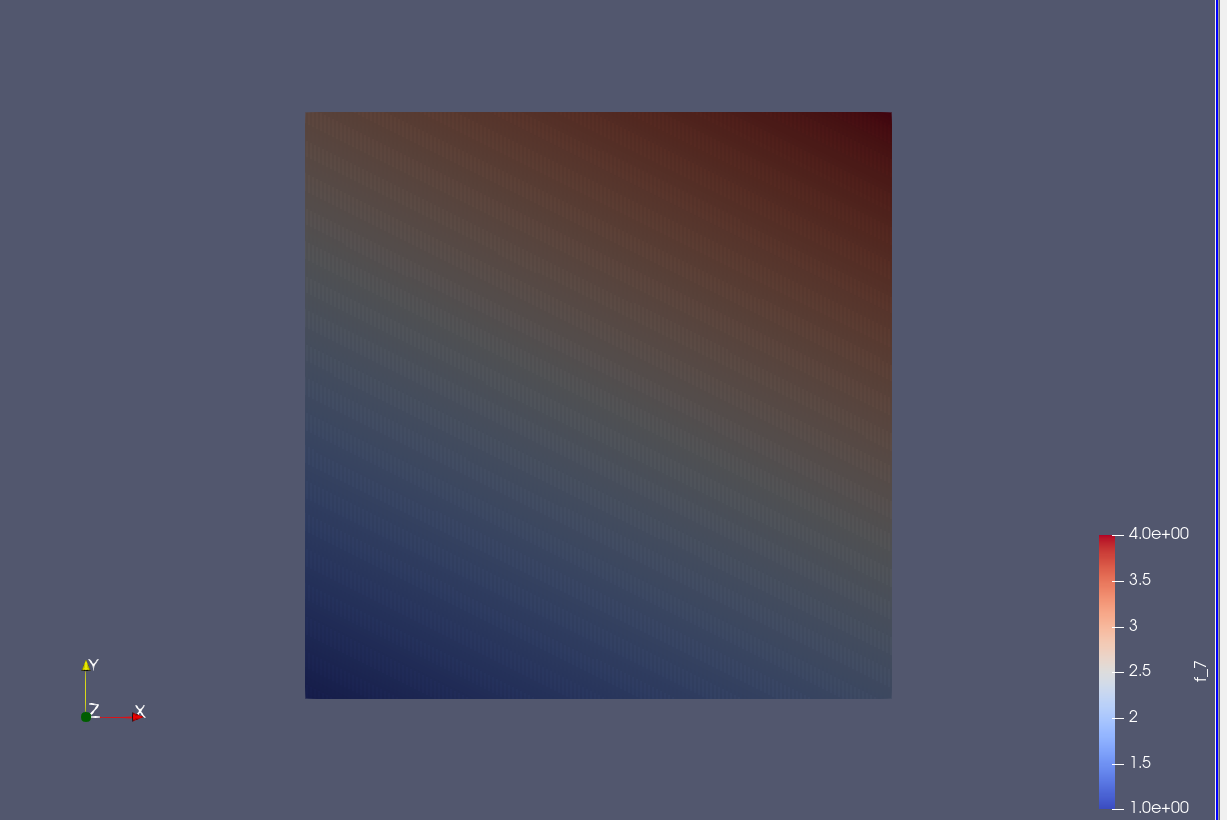
**Poisson Nonlinear 2**

div(q(u)\*grad(u)) = f in the unit square.

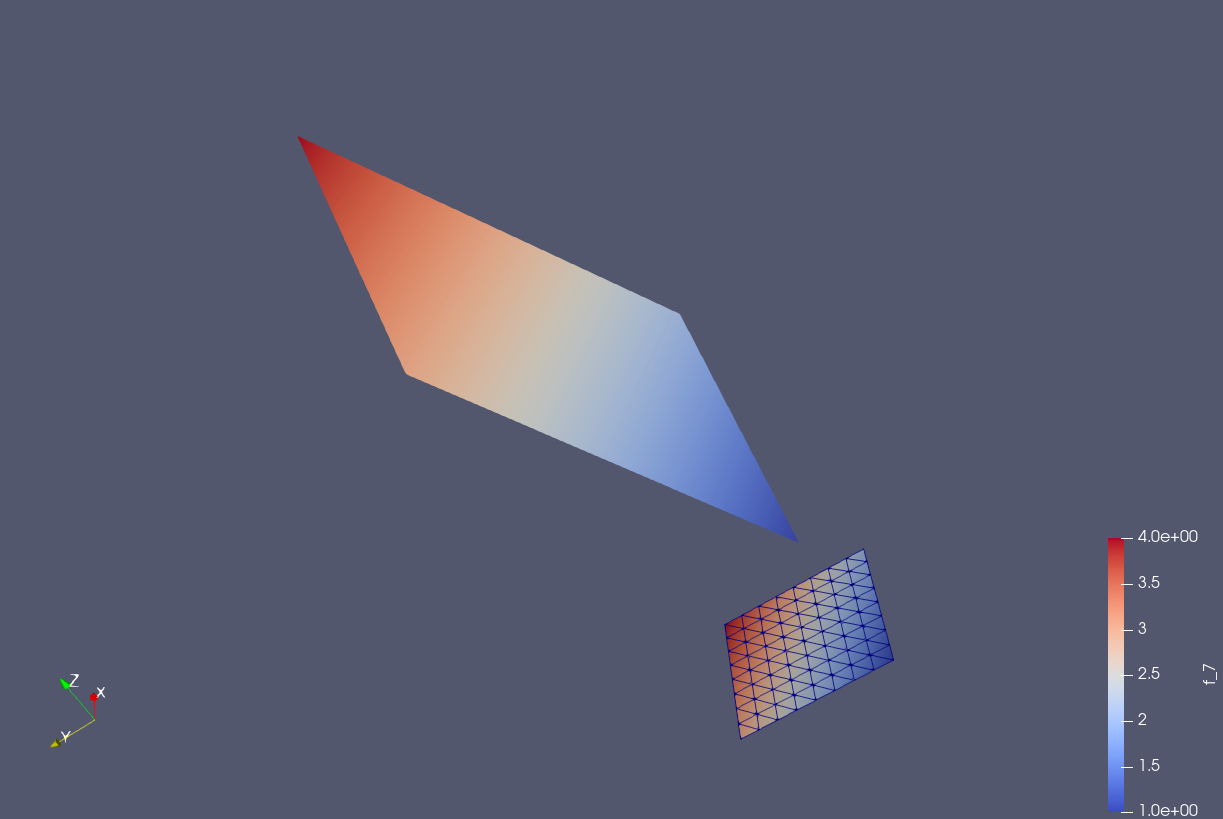
u = u\_D on the boundary.



Paraview files



Paraview 3D

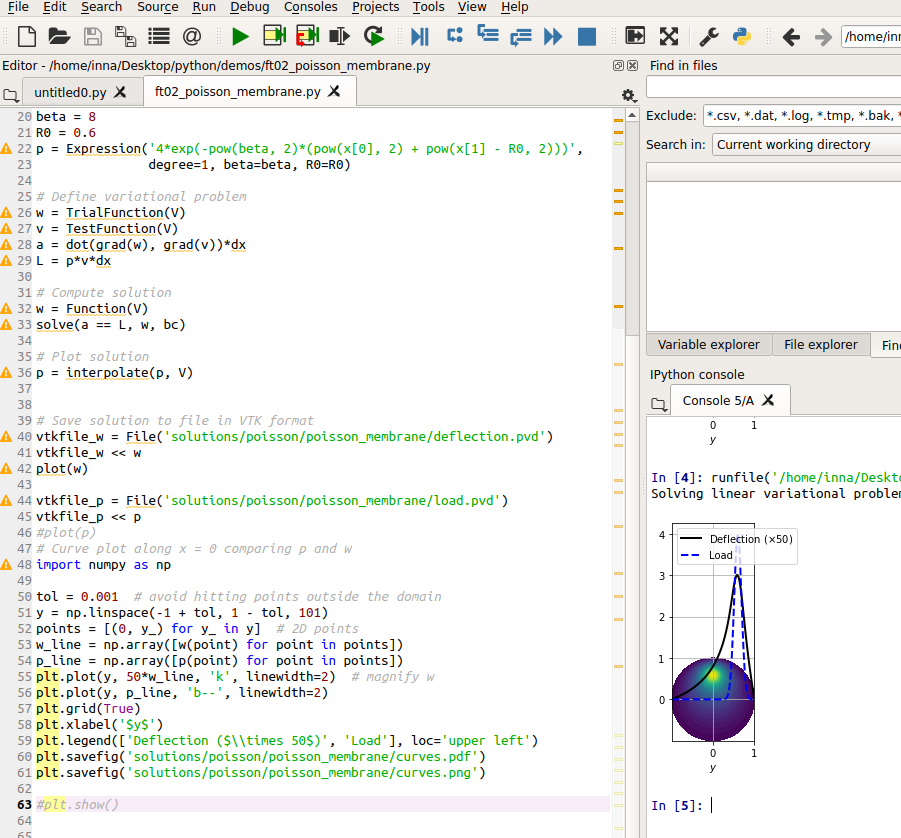


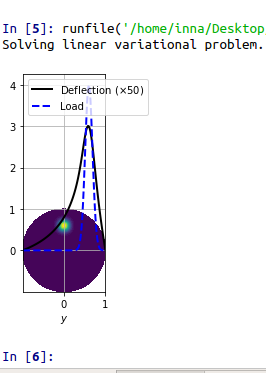
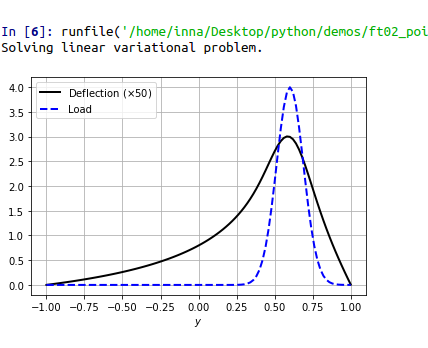
**Deflection of a membrane with DirichletBoundary conditions**

Laplace(w) = p in the unit circle

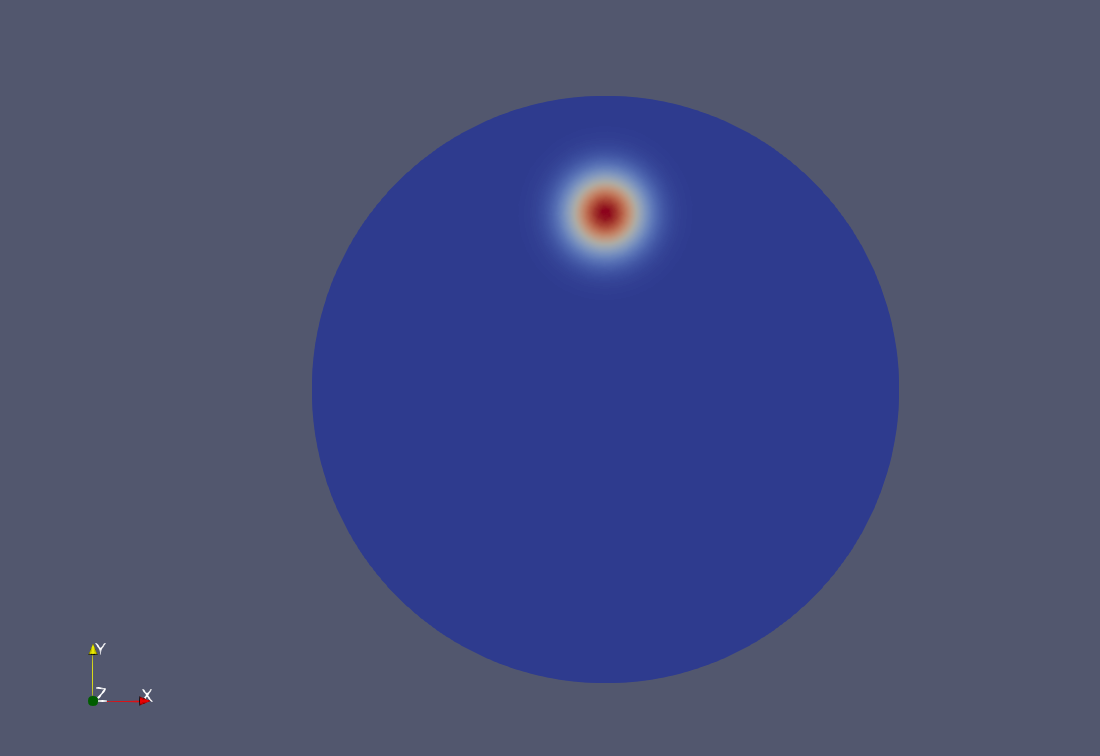
w = 0 on the boundary

The load p is a Gaussian function centered at (0, 0.6).

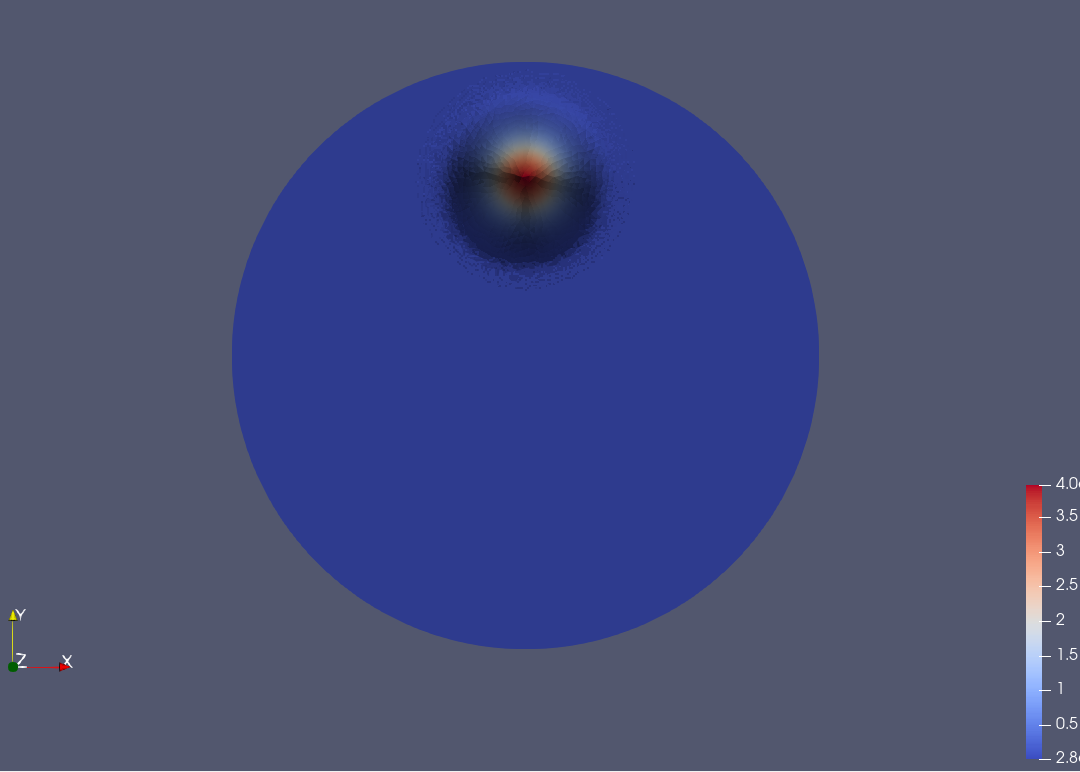


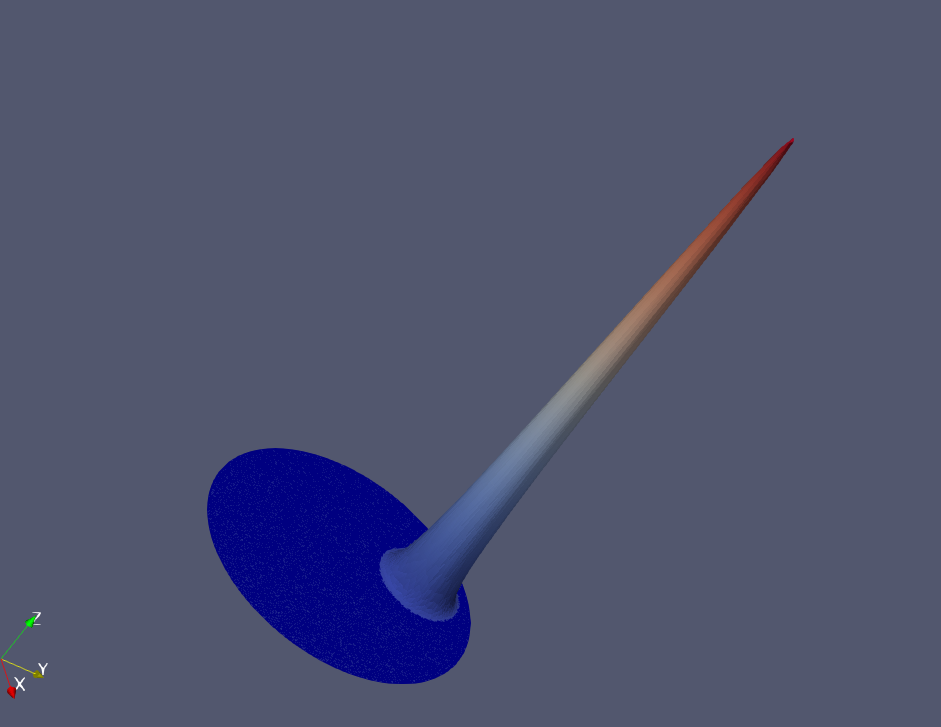
Paraview Load



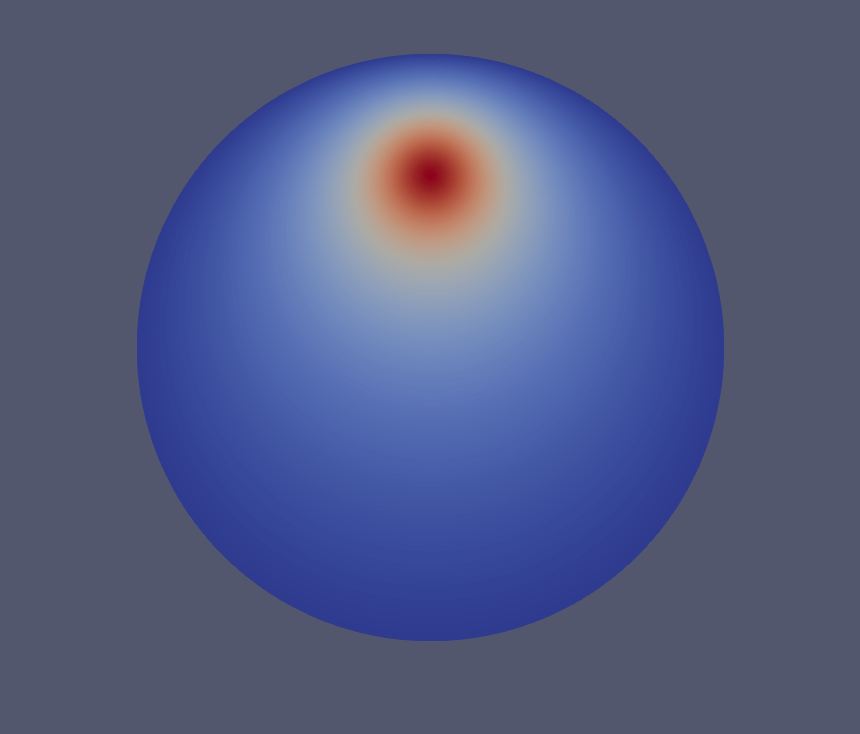
After filter by scalar is applied



3D



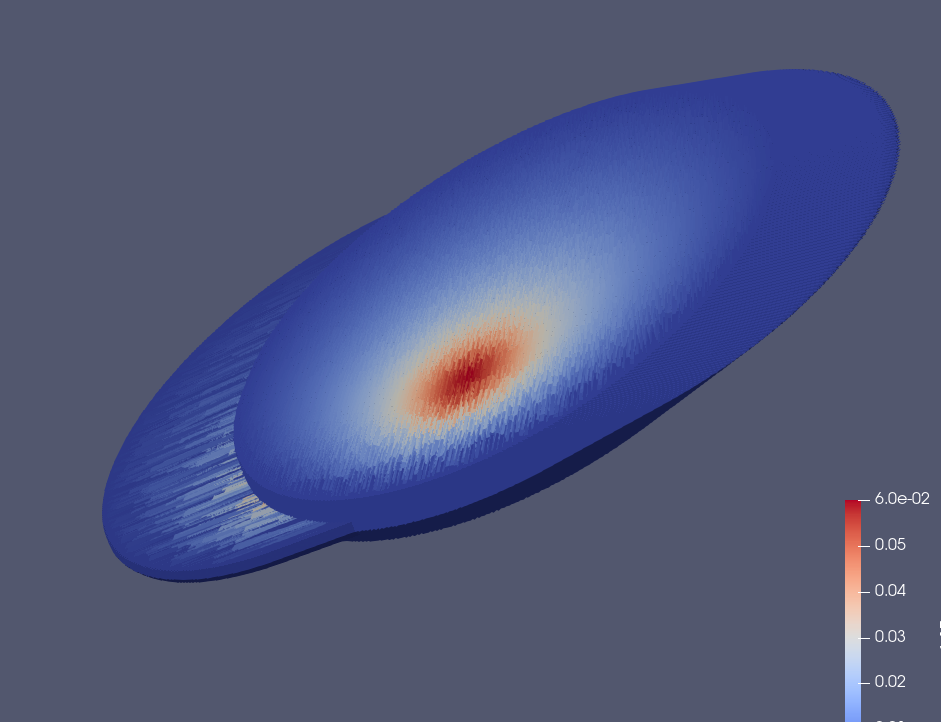
Deflection



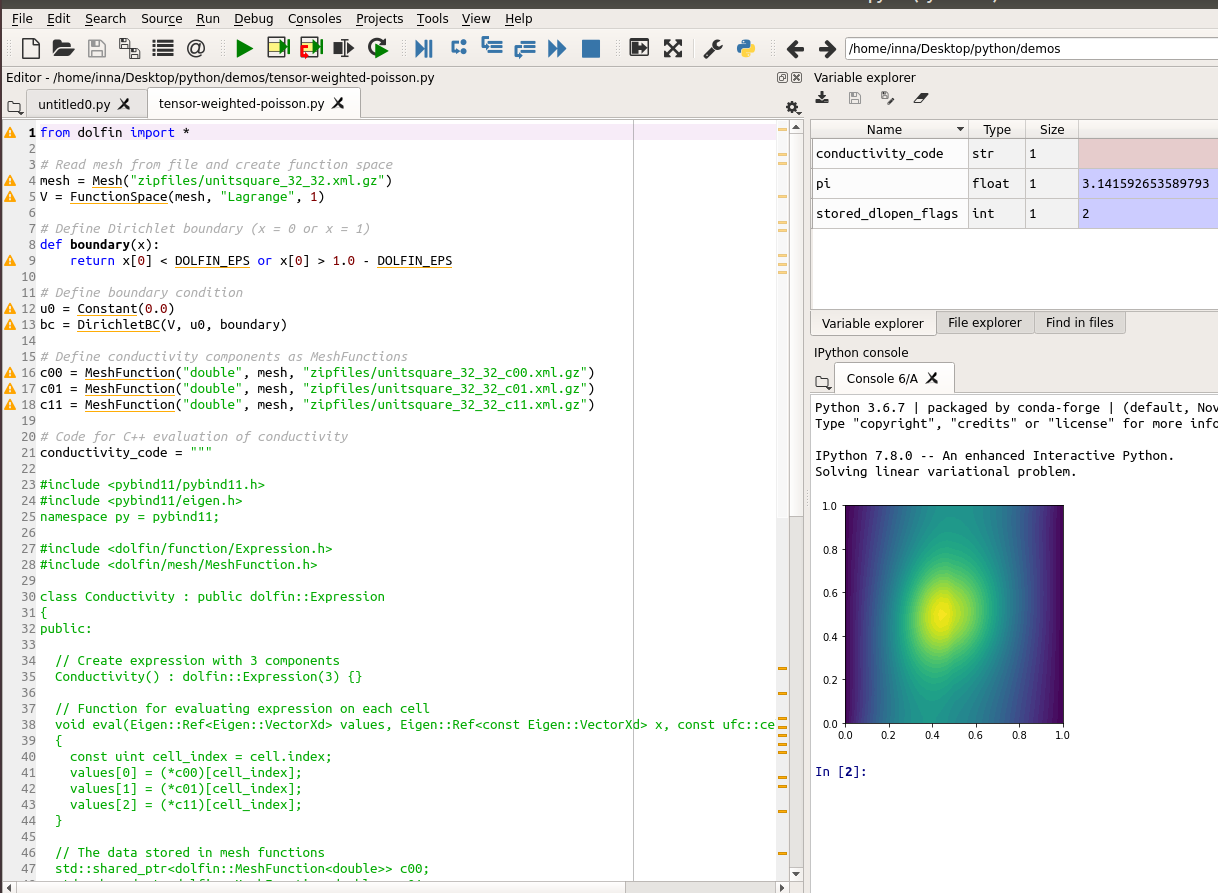
After Filter by scalar is applied



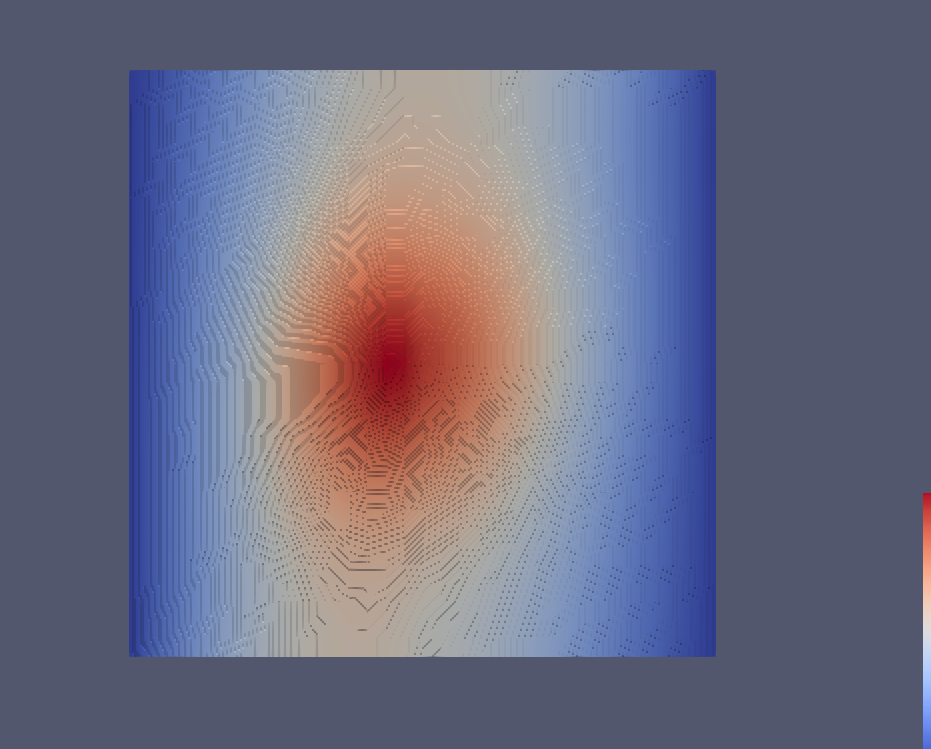
Paraview 3D



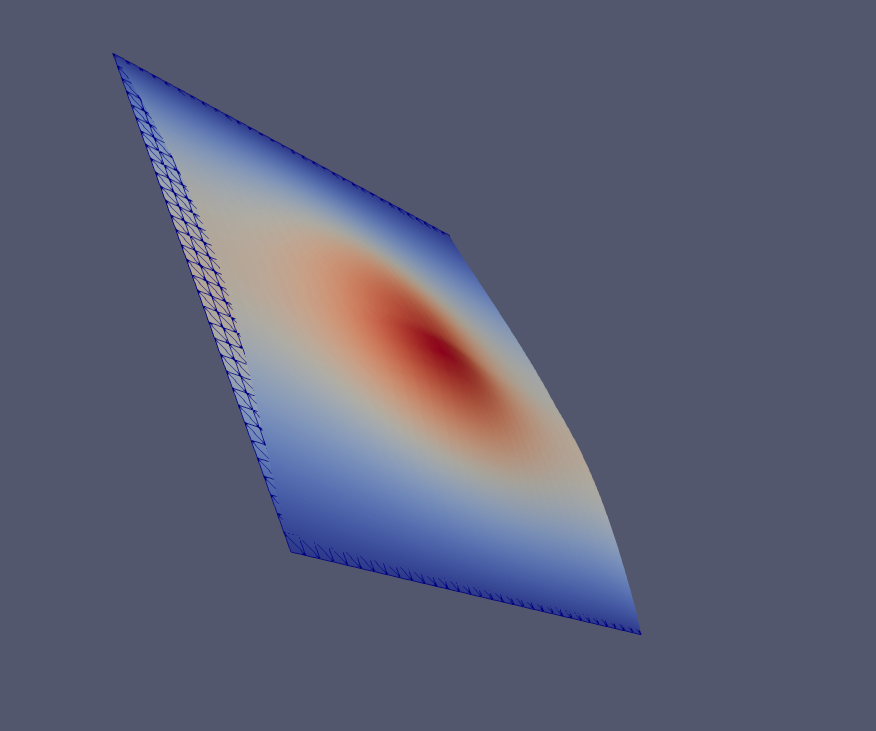
**Read Mesh from the zip file example**



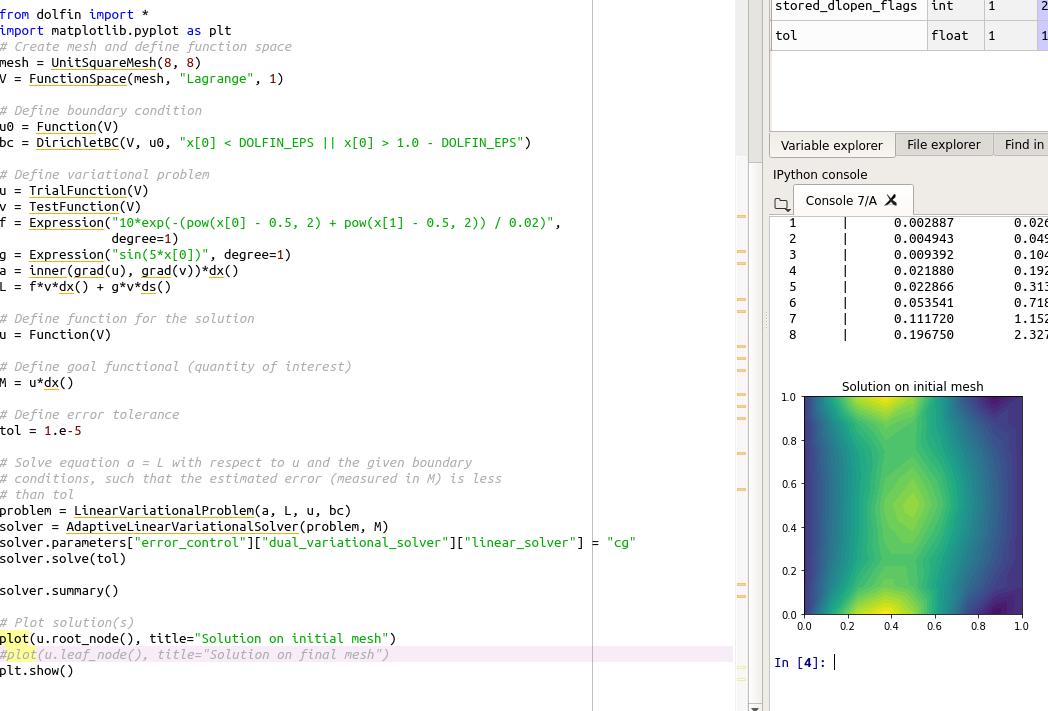
Paraview files

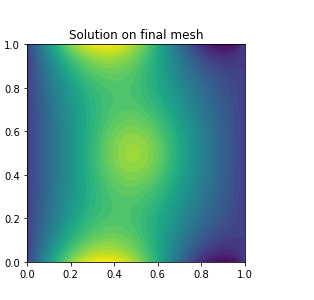


Paraview 3D

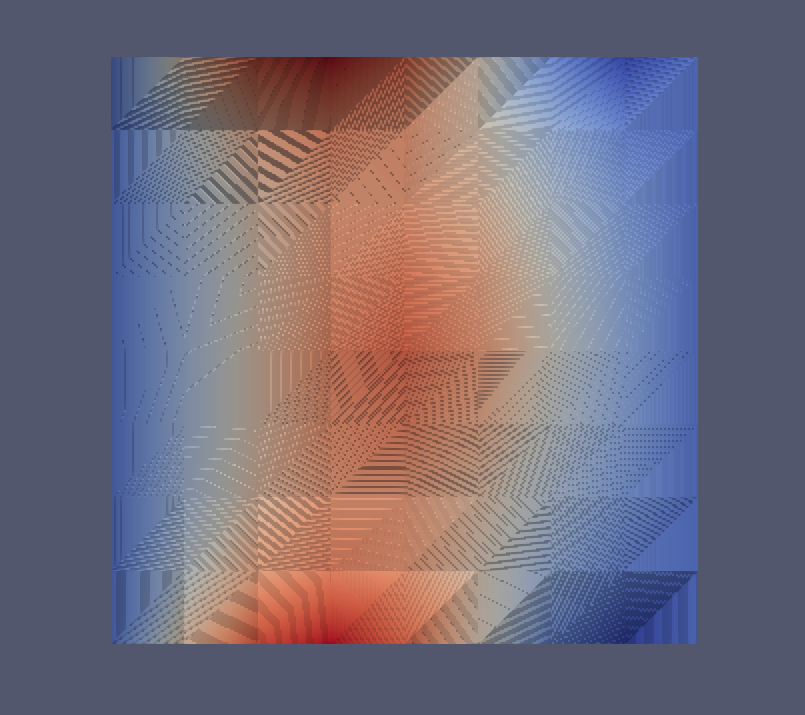


**Adaptive poisson**

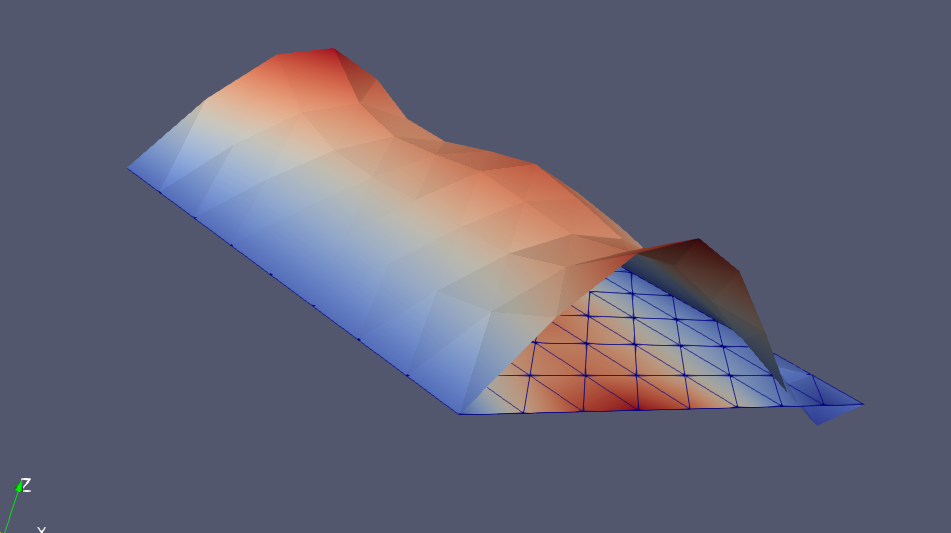
****



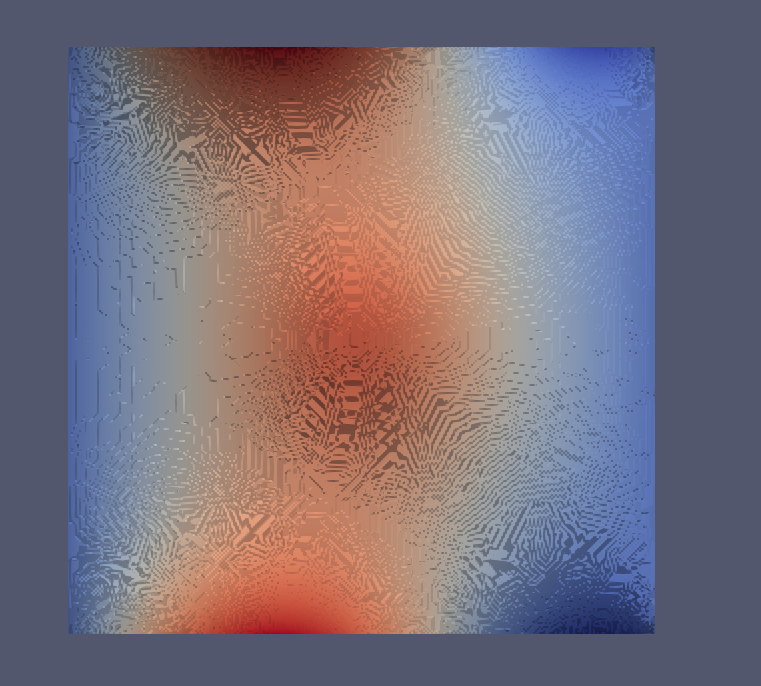
Paraview Initial mesh



3D Initial mesh



Paraview Final Mesh



3D Final Mesh

