Poisson equation solution with DEAL.II

Dara Rahmat Samii

February 20, 2023

Abstract

In this Report theory behind solving Poisson's equation with Finite Element Method is discussed. Three different geometeries of a cube, a circle and a ring and their solution is visualized with GnuPlot and ParaView.

Chapter 1

Theory

Poisson's equation strong formulation is defined as:

$$\begin{cases} -\Delta u = f & in \quad \Omega \\ u = 0 & in \quad \partial\Omega \end{cases}$$

The weak formulation of Poisson's equation can be defined as:

$$(u', \phi') = (f, \phi) \quad \forall \phi \in V \tag{1.1}$$

where (f, ϕ) is the Scalar Product of f and ϕ is can be shown that the weak form and the strong form are equivalent:

$$-u'' = f \longrightarrow \times\phi dx \longrightarrow -u''\phi dx = f\phi dx \longrightarrow \int \longrightarrow$$

$$\longrightarrow -\int_\Omega u'\phi' dx = \int_\Omega f\phi dx \longleftrightarrow \boxed{(u', \phi') = (f, \phi)}$$

After discretization of the domain it is achieved:

$$Find \;\; u \in V : (u', \phi') = (f, \phi) \longrightarrow Find \;\; u_h \in V_h : (u'_h, \phi'_h) = (f, \phi_h)$$

$$\longrightarrow ((\sum_{j=1}^{n} u_j\phi_j)', \phi'_h) = (f, \phi_h) \longrightarrow \sum_{j=1}^{n} u_j(\phi'_j, \phi'_h) = (f, \phi_h)$$

Rewriting in the Matrix format:

$$\begin{bmatrix} (\phi'_1, \phi'_1) & (\phi'_1, \phi'_2) & \dots & (\phi'_1, \phi'_n) \\ (\phi'_2, \phi'_1) & (\phi'_2, \phi'_2) & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ (\phi'_n, \phi'_1) & \dots & \dots & (\phi'_n, \phi'_n) \end{bmatrix} \times \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} (f, \phi'_1) \\ (f, \phi'_2) \\ \vdots \\ (f, \phi'_n) \end{bmatrix}$$

with solving the Matrix $AU = F$ the solution can be achieved $U = FA^{-1}$

Chapter 2

cube

In the first solution chapter, the simplest geometry will be solved. a square shape with width of 2 is generated with the code below:

```
void Poisson::make_grid()
{
    Point<2> center(0,0);
    GridGenerator::hyper_cube(triangulation, -1, 1);
    triangulation.refine_global(5);

    std::ofstream out("grid.svg");
    GridOut       grid_out;
    grid_out.write_svg(triangulation, out);
}
```

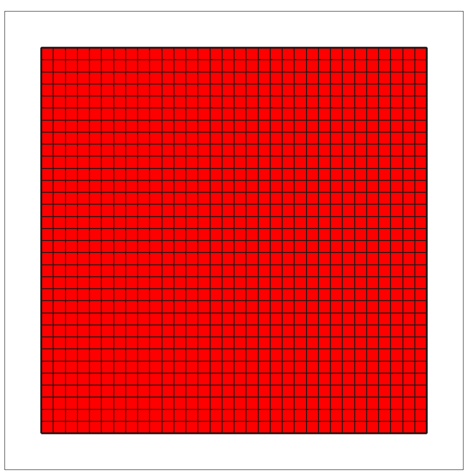

Figure 2.1: mesh grid for square geometry

In the Poisson's equation there is a term of f. For the simplicity with assigned $f = 1$. The surface 3D plot and contour plot are visualized with GnuPlot and ParaView, respectively.

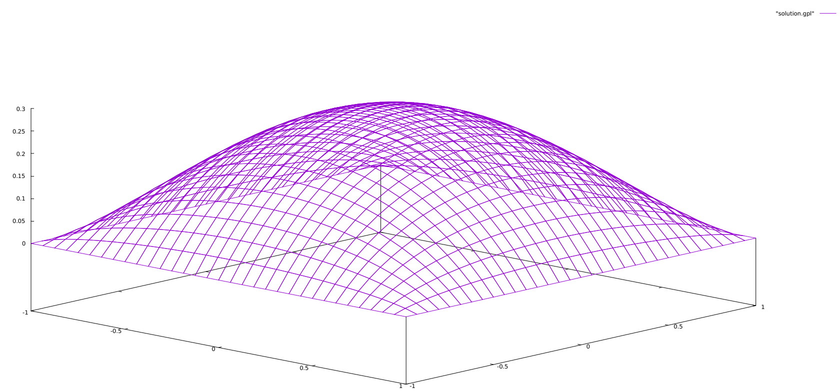

Figure 2.2: 3D surface solution for Poisson's equation with $f = 1.0$

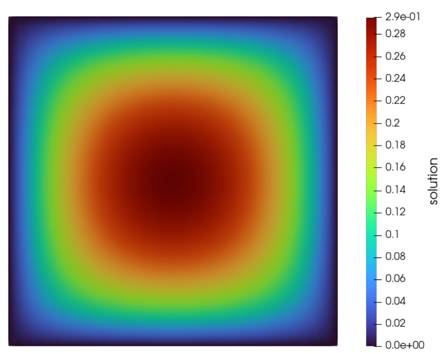

Figure 2.3: Solution contour for Poisson's equation with $f = 1.0$ on square geometry

Chapter 3

circle

In the second case, the Poisson's equation will be solved on a circle domain with Radius of 1.0. The code which generates the circle mesh domain is:

```
void Poisson::make_grid(){
    Point<2> center(0,0);
    const double Radius = 1.0;
    GridGenerator::hyper_ball(triangulation,
                              center, Radius, 10);
    triangulation.refine_global(5);
    std::ofstream out("grid.svg");
    GridOut       grid_out;
    grid_out.write_svg(triangulation, out);
}
```

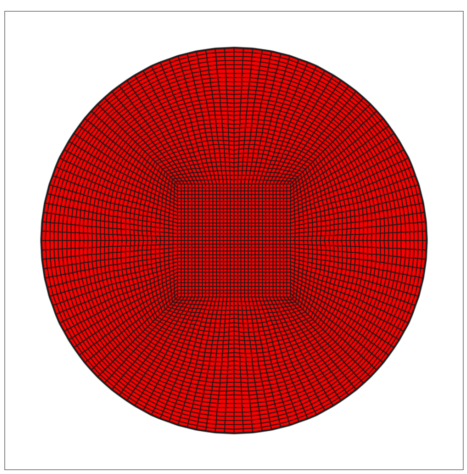

Figure 3.1: mesh grid for circle geometry

For the simplicity with assigned $f = 1$. The surface 3D plot and contour plot are visualized with GnuPlot and ParaView, respectively.

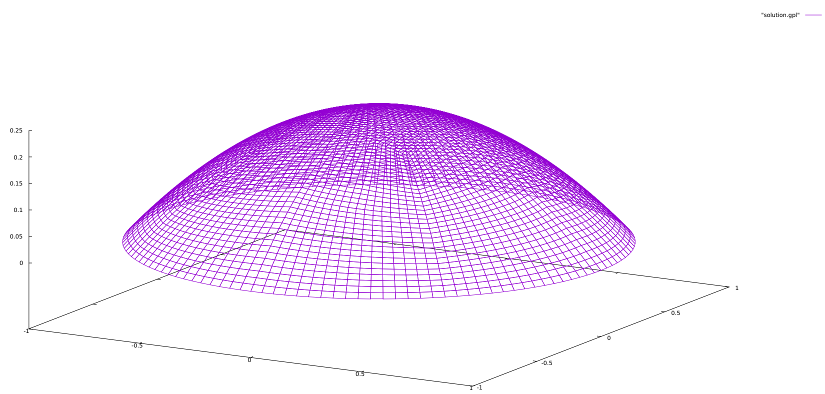

Figure 3.2: 3D surface solution for Poisson's equation with on circle domain

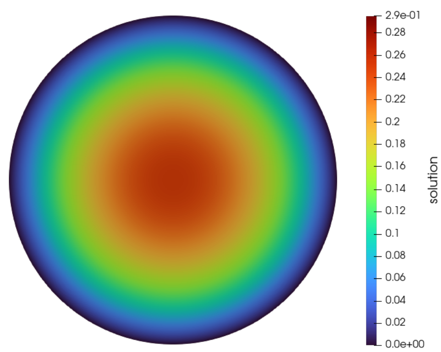

Figure 3.3: Solution contour for Poisson's equation on circle domain

Chapter 4

ring

In the Final case, the Poisson's equation will be solved on more complex domain which consist of to hole in the middle of a circle domain with outer radius of 1.0 and inner radius of 0.2 . The code which generates the circle mesh domain is:

```
void Poisson::make_grid(){
    Point<2> center(0,0);
    const double Radius_in = 0.2, Radius_out=1.0;
    GridGenerator::hyper_shell(triangulation, center,
                               Radius_in, Radius_out, 10);
    triangulation.refine_global(5);
    std::ofstream out("grid.svg");
    GridOut       grid_out;
    grid_out.write_svg(triangulation, out);
}
```

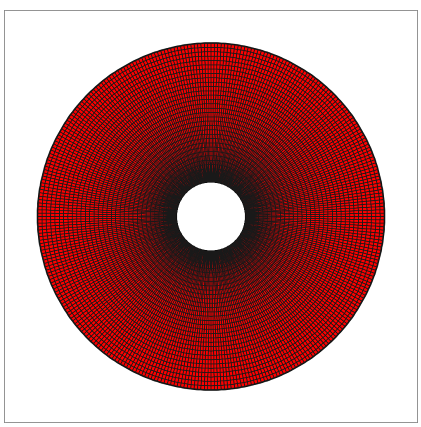

Figure 4.1: mesh grid for Ring geometry

For the simplicity with assigned $f = 1$. The surface 3D plot and contour plot are visualized with GnuPlot and ParaView, respectively.

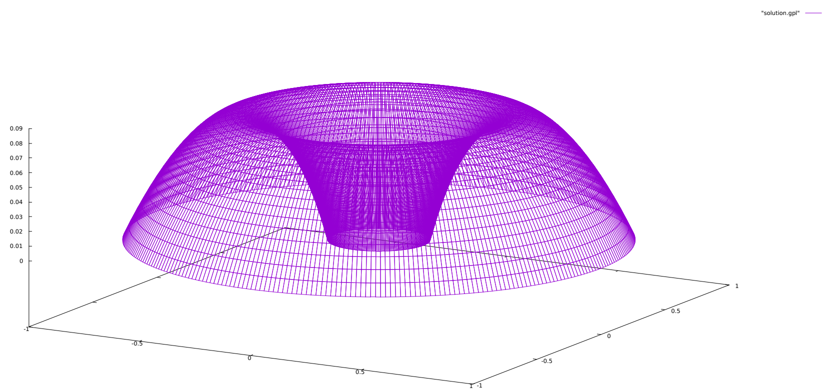

Figure 4.2: 3D surface solution for Poisson's equation with on Ring domain

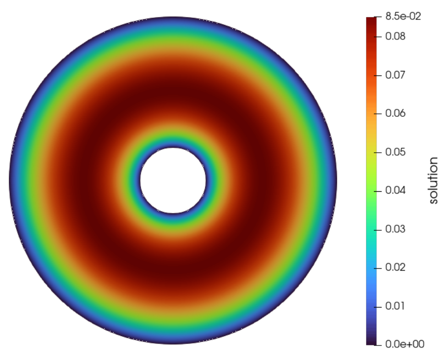

Figure 4.3: Solution contour for Poisson's equation on Ring domain