Problem: Numerical simulation of advection-diffusion problem using Dealii frame work

Introduction:

In this report the advection-diffusion equation (1) is simulated using Dealii, FEM open source code for PDE.

$$\begin{aligned} \beta.\nabla u-\Delta u &= f \qquad in \quad \Omega \\ u &= g \qquad on \quad \partial\Omega \end{aligned} \tag{1}$$

Numerical procedure:

1:Transforming the equation (1) into weak form by multiplying a test fumction (φ) to fasilitate the numerical procedure.

$$\left(\varphi,\beta.\nabla u\right)-\left(\varphi,\Delta u\right)=\left(\varphi,f\right) \qquad in \quad \Omega \tag{2}$$

Integration by part and divergence theorem:

$$\int\nabla.\left(\varphi\nabla u\right)=\int\nabla\varphi.\nabla u+\int\varphi\Delta u \;\; and \qquad \int_{\Omega}\nabla.\left(\varphi\nabla u\right)=\int_{\partial\Omega}\varphi n.\nabla u \quad (Divergence\, theorem)$$

If u=g on the boundary, then: $\varphi=0 \quad on \;\; \partial\Omega, \quad -> \;\; \left(\varphi,n.\nabla u\right)=0 \quad on \;\; \partial\Omega$. So, the boundary condition is strong and consequently: $\left(\varphi,\Delta u\right)=-\left(\nabla\varphi,\nabla u\right)$

(3)

By combination of equations 2 and 3, the weak form is given by:

$$\left(\varphi,\beta.\nabla u\right)+\left(\nabla\varphi,\nabla u\right)=\left(\varphi,f\right) \qquad in \quad \Omega \tag{4}$$

After linearizing the PDE, the representation $u_h\left(x\right)=\sum_j U_j\varphi_j\left(x\right)$ is inserted to form **AU=F**:

$$\begin{aligned} A\left(i,j\right)&=\left(\varphi_i,\beta.\nabla\varphi_j\right)+\left(\nabla\varphi_i,\nabla\varphi_j\right) \\ F\left(i,j\right)&=\left(\varphi_i,f\right) \end{aligned} \tag{5}$$

Boundary conditions of the domain, which is the square $[-1,\, 1] \times [-1,\, 1]$, is as follows;

$$\beta\left(x\right)=\begin{pmatrix} 2 \\ 1+\frac{4}{5}\sin\left(8\pi x\right) \end{pmatrix}, \qquad f\left(x,y\right)=0.01*\left(x^2+y^2\right):\left(Right\_hand\_side\right)$$

$$g\left(y\right)=-y^2+1 \quad inlet$$

Where β is a vector field that describes the advection direction and speed, the g(y) is considered at inflow, and the other boundaries are treated as internal nodes.

2: making grid and associating degree of freedom (Dof) numbers to each vertex, line or cell, and enumerates Dof on the mesh.

3: Considering the degree of polynomial for type of finite elements, i.e. linear, and etc. (describing the shape functions we want to use on the reference cell)

4: To constitute the algebraic equation, the quadrature points, weight functions and etc. are calculated by Gaussian quadrature formulas in Gaussian domain, then shape functions, their gradients, and etc. are evaluated by FEValue class. In fact, the integration is calculated using *Gaussian* quadrature rule.

5: Considering **AU=F**, after setting up all data structures in A and F (setup_system and assemble_system) and applying appropriate boundary condition, the solver is called and output_results would be the outcome.

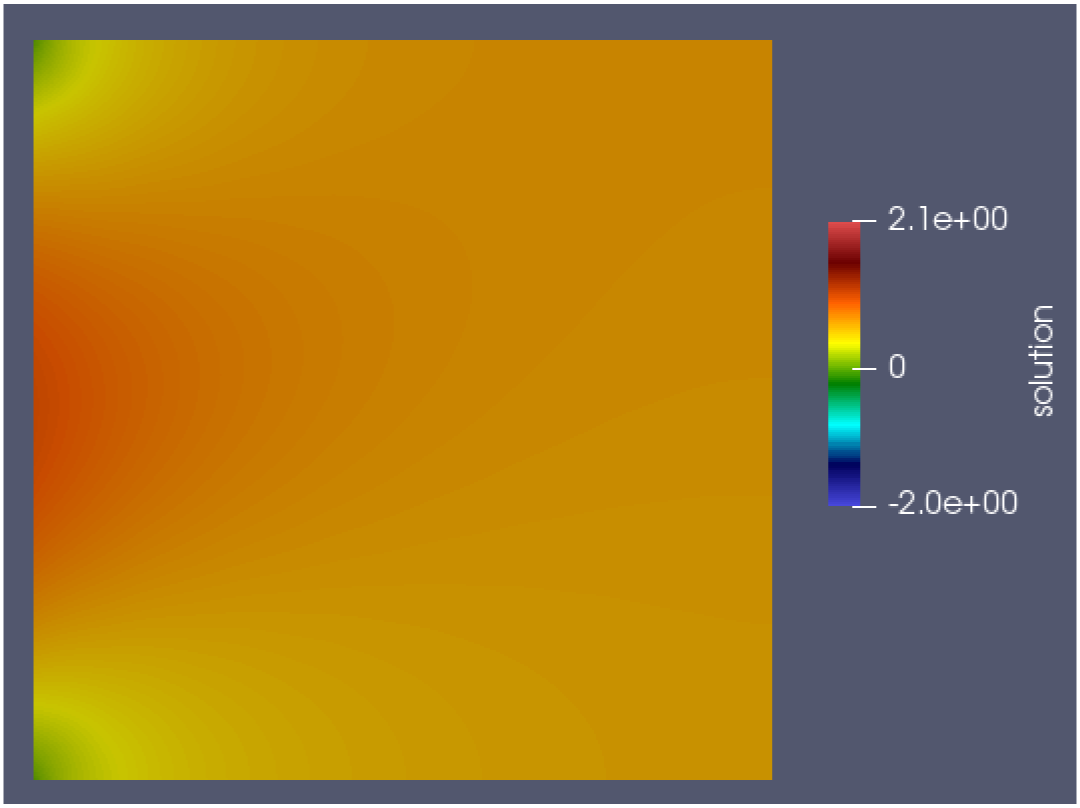
Result:

Number of active cells: 64

Number of degrees of freedom: 625

Iterations required for convergence: 231

Max norm of residual: 8.33695e-10



Contour plot of “u” as a solution of the advection-diffusion problem

Discussion:

The influence of parabolic inflow at left boundary and also the source function which has a direct relation to (x^2 +y^2) can be seen clearly from the contour plot of u.