**A Simple 2D Transport Model with Penalization Method**

**Emir Muhammad**

**Introduction**

Modelling the transport of particles through a perforated domain is one of the major problems found within the field of porous media. This transport obeys the Advection-Diffusion Equation, which is challenging to simulate in a domain with multiple obstacles. Several solvers already exist that can simulate this transport; however, they are complicated and are difficult for student programmers to understand. As such, a solver based of Dr Muljadi’s program was developed in MATLAB as it is a popular introductory programming language for engineers and would, therefore, be a great tool to get started in numerical modelling. This MATLAB solver simulates the steady-state Advection-Diffusion equation with Dirichlet boundary conditions on a regular 2D cartesian mesh using the Q1 Finite Elements Method with the penalization method.

**The domain**

The domain of the problem is a rectangular area with regular perforations, shown in Figure 1a. The domain is called Ω, and it includes everything in the rectangular area. The set of perforations is called Bε, and the domain without the perforations is Ωε = Ω\Bε.

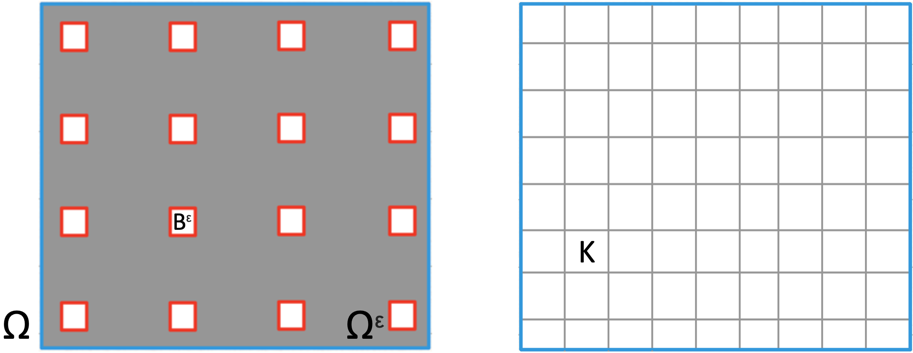


Figure 1a) Left shows the Domain Ω, including the perforations b) Right shows the discretized domain. Boundaries have been coloured similarly to Equation 1.

The domain is discretized in Figure 1b by dividing Ω into a series of rectangular elements K. The 4 vertices at each element are the nodes, and the Dirichlet boundary nodes are Γp.

**The Governing Equations**

The Governing Equations in this domain, which are the Advection Diffusion Equation and boundary conditions, are

where D is the scalar diffusion coefficient, **w** is the velocity vector field, u is a scalar field, f is the source terms, and g is the scalar representing the value at the boundary. The boundary condition is labelled with colours to correspond with Figure 1.

The basis functions are the set of unique linear functions where the value is 1 at node i and 0 at every other node, satisfying

and used as our test function. Given that the solution uh­ can be written as

the weak form can be written as a system of linear equations expressed as a matrix

for n, m = 1, …, number of nodes, and p = 1, …, number of boundary nodes, where

with λp being Lagrange multipliers used to impose the boundary conditions.

**Penalization Method**

The penalization method allows us to continue using the uniform cartesian mesh instead of a complicated mesh. To use the penalization method, the ADE equation is modified as follows,

,

The term forces the values of u to vanish if the element is within a perforation, and the sigma term disappears in Ωε, recovering the original equation.

**Numerical Solution**

As an example, the simulation is run using 128x128 elements, with a vector field of (2y(1-x2), -2x(1-y2)), diffusion coefficient D set to 0.05, and the source term {−1≤x≤1, −1≤y≤−0.7}

A picture containing screenshot

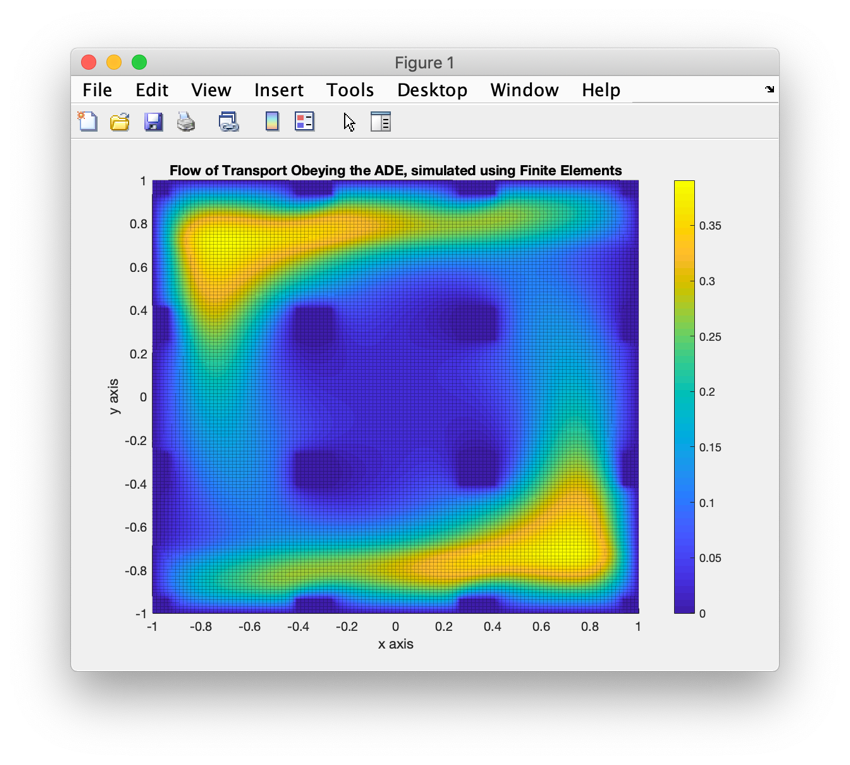
Description automatically generated

Figure 2 The graph generated by the MATLAB Solver using the above conditions.

**Conclusion**

This MATLAB program simulates the steady-state Advection-Diffusion Equation using the Q1 finite elements method, generating the matrix using equation X and solves the system and graphing it as shown in Figure something. The program makes it easy to change the parameters, making it versatile. This was a simple implementation of the Finite Elements method, making it easy for programmers to get their feet wet into finite element modelling systems.