

Note: This document contains the 'gist' of FEM. This is not mathematically rigorous and contains handwavy logic. For a better introduction, I recommend 'A Gentle Introduction to the FEM' by Francisco-Javier Sayas, 2008.

We would like to model the flow of the concentration of a substance through a Perforated Medium. The flow obeys the advection-diffusion equation.

The Domain

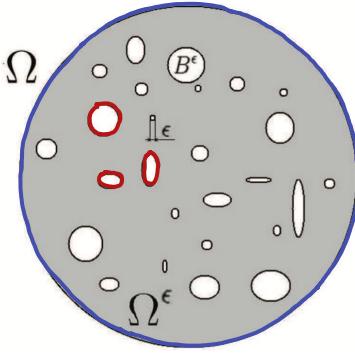


Figure 1: Perforated domain $\Omega \in \mathbb{R}^d$ within which a set B^ϵ of perforations is included. The perforated domain with voids left by perforations is denoted $\Omega^\epsilon = \Omega \setminus B^\epsilon$, where ϵ denotes the minimum width of perforations.

The Governing Equations

ADE

$$-\nabla \cdot (D \nabla v) + \underline{w} \cdot \nabla v = f$$

v = scalar field variable (eg. concentration, Temp)

D = diffusion coefficient

\underline{w} = velocity vector

f = source term

Boundary Conditions

$$v = 0 \text{ on } \partial B^\epsilon \cap \partial \Omega^\epsilon,$$

$$v = g \text{ on } \partial \Omega \cap \partial \Omega^\epsilon$$

The Problem

- Second order PDE
- Implies that U is both continuous and very smooth
- As we are simulating using an approximation, don't want it to be smooth everywhere
- Need to convert strong form (PDE) into weak form (integrals)

Greens theorem

$$\int_{\Omega} (\nabla^2 v) r + \int_{\Omega} \nabla v \cdot \nabla r = \int_{\Gamma} (\partial_n v) r, \quad \Gamma = \text{boundary}$$

This is the heart of converting to weak form

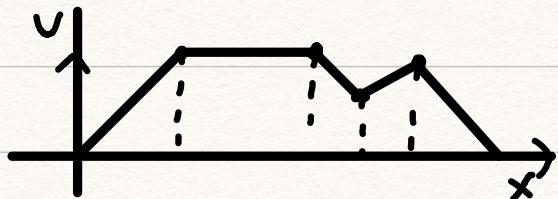
Weak Form

Sub in $\nabla^2 v = \frac{1}{D} (\underline{w} \cdot \nabla v - f)$ and rearrange

$$\int_{\Omega} (\underline{w} \cdot \nabla v) r + D \int_{\Omega} \nabla v \cdot \nabla r = \int_{\Omega} f r,$$

r = test function

Solution therefore does not need to be smooth
(have a derivative) everywhere.



← Continuous, but not smooth.
derivative not defined at
the Sharp Points.

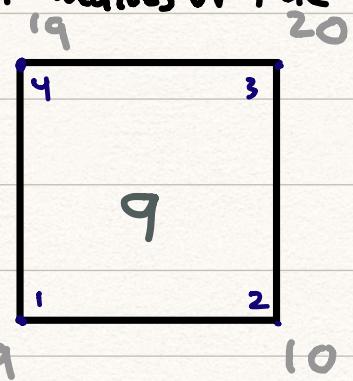
Now we have to discretize the domain

Discretizing the Domain

- Need to split up the domain into grids
- The computer can then calculate the numerical value at each vertex, which is called a node
- The entire set of numerical values at each part makes up the solution

Domain Split into Grids

41	42	43	44	45	46	47	48	49	50
29	29	30	31	32	33	34	35	36	30
31	32	33	34	32	33	34	35	34	30
19	20	21	22	23	24	25	26	27	30
31	32	23	24	25	26	27	28	29	30
10	11	12	13	14	15	16	17	18	10
11	10	12	13	14	15	16	17	18	10
1	2	3	4	5	6	7	8	9	10



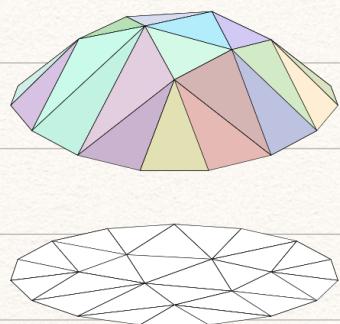
Note: In general, the grids don't have to be uniform, but in this program it is so we will only consider these.

We can define several features of this domain.

The grey numbers are the global node numbers, while the dark grey numbers are the element numbers. The blue ones are the local node numbers. Their relationship is written as follows.
 $G = n_d^k$, where d = local node number and k = element number

FEM approximates the solution by using a series of linear functions as shown here. All linear functions belong in the functional space

$$\textcircled{Q} \quad 1 = \{ q_0 + q_1 x + q_2 y + q_3 xy \mid q_0, \dots, q_3 \in \mathbb{R} \}$$



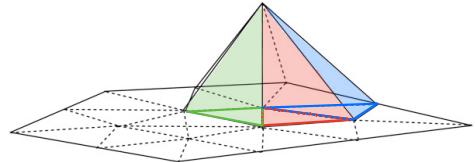
Next we need to define the Basis Functions, which belong in Q_2 , and are extremely important.

2d triangulation with linear elements

Basis Functions

The Basis Functions are a set of linear functions that have a value of 1 at the i^{th} node, and 0 at every other node.

There exists basis functions for each vertex.



The basis function for this particular node

Mathematically, $\underline{P}_j = \text{location of } \varphi_i(\underline{P}_j) = S_{ij}, \quad i^{\text{th}} \text{ vertex}$

We Should be more Precise with our definitions.

Global basis function is the entire tent above, and denoted with φ_i , where i is the node number

But the Global basis function is obviously made up of individual linear functions from different elements. The red function originates from the red element, and so on. These are called the local basis functions, denoted with N_j^k , where j is the local node and k is the element.

Weak form to Matrix form

Weak form, Shown again

$$D \int_{\Omega} \nabla v \cdot \nabla r + \int_{\Omega} (\underline{w} \cdot \nabla v) r = \int_{\Omega} f r,$$

rewriting this, while being explicit that v and r are linear,

$$D \int_{\Omega} \nabla v_h \cdot \nabla r_h + \int_{\Omega} (\underline{w} \cdot \nabla v_h) r_h = \int_{\Omega} f r_h,$$

where v_h and r_h are linear.

Have the test function $r_h = \varphi_i$ for all nodes, and
given that $v_h = \sum_{j \in \text{nodes}} v_j \varphi_j$

we have

$$\sum_{j \in \text{Node}} \left(\int_{\Omega} D \nabla \varphi_j \cdot \nabla \varphi_i + \int_{\Omega} (\underline{w} \cdot \nabla \varphi_j) \varphi_i \right) v_j = \int_{\Omega} f \varphi_i$$

Matrix Form

$$\underline{\underline{A}}_{ij} \underline{v}_j = \underline{f}, \Rightarrow \text{computer solvable.}$$

$$\underline{\underline{A}}_{ij} = D_{ij} + W_{ij}$$

$$= \int_{\Omega} D \nabla \varphi_j \cdot \nabla \varphi_i + \int_{\Omega} (\underline{w} \cdot \nabla \varphi_j) \varphi_i$$

Now all that is left is to assemble the matrix

Assembly

We can decompose the integral over the domain S_n as a sum of integrals over each element $\sum_k S_k$.

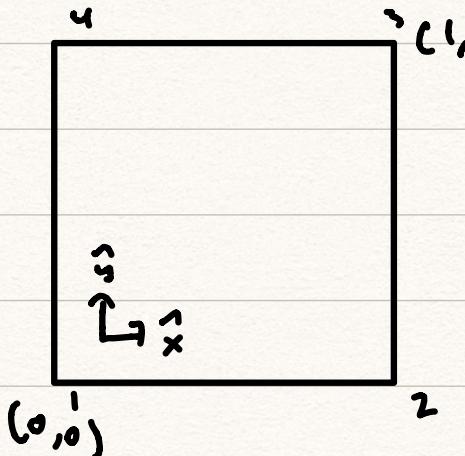
$$D_{ij} = \int_n D \nabla \varphi_j \cdot \nabla \varphi_i = \sum_k \int_k D \nabla \varphi_j \cdot \nabla \varphi_i = \sum_k D_{ij}^k$$

As we have defined local basis functions, we can use them to calculate the local matrix D_{ij}^k

$$D_{ij}^k = \int_k D \nabla N_\beta^k \cdot \nabla N_\alpha^k, \quad \alpha, \beta = \text{local node numbers}$$

Expressions for Basis Functions

For an example element



$$\begin{aligned}\hat{N}_1 &= (1-\hat{x})(1-\hat{y}) \\ \hat{N}_2 &= (\hat{x})(1-\hat{y}) \\ \hat{N}_3 &= (\hat{x})(\hat{y}) \\ \hat{N}_4 &= (1-\hat{x})(\hat{y})\end{aligned}$$

Just need to modify these equations to fit the particular element.

Gaussian Quadrature

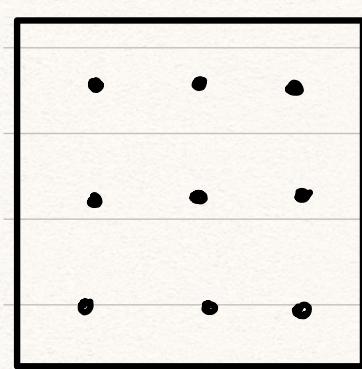
To numerically solve the integral, Gaussian Quadrature is used.

$$\int_{-1}^1 f(x) = \sum_{i=1}^n w_i f(x_i), \quad w_i = \text{weights}$$

$x_i = \text{pos. coordinates.}$

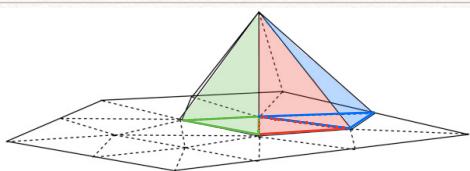
In this context, we would need to modify it into 2D.

$$\int_K D \nabla N_p^k \cdot \nabla N_q^k = \sum_{i,j} D w_i w_j \nabla N_p^k(x_i, y_j) \cdot \nabla N_q^k(x_i, y_j)$$



(1,1)
Example: Calculate values at those 9 points
and add them together, weighted.

Once this is done, what is left to do is to assemble the local matrix elements into the global matrix.



The basis function for this particular node

Onto their respective global node. Basically, the local elements hold red, green, and blue values on different elements, and we have to combine them together.

In a simplified explanation, we need to add up the contributions from each element

This is the gist of the global assembly, should be enough background knowledge to start understanding the global map and assembly.

Lagrangian Multipliers

So far we have yet to consider the boundary conditions. A good way to deal with the Dirichlet conditions, which are a series of constraints, is to use Lagrange multipliers.

Say we have a 7×7 system, $Ax = b$

Subject to conditions $x_1 \geq 0, x_2 \geq 0, x_4 = x_6$

These can be written as $Bx \geq h$, expanded such that

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

B is known as the boundary condition matrix.

Now, similar to the normal use of Lagrange multipliers, we introduce $\lambda^T (Bx - h) \geq 0$, where λ is a set of quantities called Lagrange multipliers.

If this is weird, refresh your memory on 'How to use Lagrange multipliers to find the max or min of a function under a constraint'.

So to therefore solve the system with those boundaries, we make a matrix so that it looks like

$$\left[\begin{array}{c|c} A & B \\ \hline B & 0 \end{array} \right] \left[\begin{array}{c} x \\ \lambda \end{array} \right] = \left[\begin{array}{c} b \\ h \end{array} \right]$$

Expanded it is

$$\left[\begin{array}{ccccccc|ccc} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} & 1 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} & 0 & 1 & 0 \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & A_{37} & 0 & 0 & 0 \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & A_{47} & 0 & 0 & 1 \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} & A_{57} & 0 & 0 & 0 \\ A_{61} & A_{62} & A_{63} & A_{64} & A_{65} & A_{66} & A_{67} & 0 & 0 & -1 \\ A_{71} & A_{72} & A_{73} & A_{74} & A_{75} & A_{76} & A_{77} & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{array} \right] = \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ \hline \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{array} \right] = \left[\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ \hline 0 \\ a \\ 0 \end{array} \right]$$

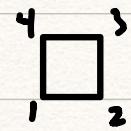
Modifying Our Equation

$$\sum_{j \in \text{Node}} \left(\int_{\Omega} D \nabla \varphi_j \cdot \nabla \varphi_i + \int_{\Omega} (\omega \cdot \nabla \varphi_j) \varphi_i \right) v_j + \sum_r \lambda_r \varphi_i = \int_{\Omega} f \varphi_i$$

with $\sum_i (\varphi_i) \int_{\Omega} v = h$ to get the above format

Huh? What's going on? What does all that mean?

$$\sum_{j \in \text{Node}} \left(\int_{\Omega} D \nabla \varphi_j \cdot \nabla \varphi_i + \int_{\Omega} (w \cdot \nabla \varphi_j) \varphi_i \right) v_j = \int_{\Omega} f \varphi_i$$

Say our domain consists of only 4 points,  so there are 4 nodes.

Let's examine node $i=1$. For this node, we have to get the contributions of the node pairs $(1,1)$, $(1,2)$, $(1,3)$ and $(1,4)$. So for the $(1,1)$ pair, the equation looks like

(i,j)

$$(1,1) = \left(\int_{\Omega} D \nabla \varphi_1 \cdot \nabla \varphi_1 + \int_{\Omega} (w \cdot \nabla \varphi_1) \varphi_1 \right) v_1 = A_{11} v_1$$

$$(1,2) = \left(\int_{\Omega} D \nabla \varphi_2 \cdot \nabla \varphi_1 + \int_{\Omega} (w \cdot \nabla \varphi_2) \varphi_1 \right) v_2 = A_{12} v_2$$

$$(1,3) = \left(\int_{\Omega} D \nabla \varphi_3 \cdot \nabla \varphi_1 + \int_{\Omega} (w \cdot \nabla \varphi_3) \varphi_1 \right) v_3 = A_{13} v_3$$

$$(1,4) = \left(\int_{\Omega} D \nabla \varphi_4 \cdot \nabla \varphi_1 + \int_{\Omega} (w \cdot \nabla \varphi_4) \varphi_1 \right) v_4 = A_{14} v_4$$

We can do it for node 2 also. Example using pair $(2,1)$,

$$(2,1) = \left(\int_{\Omega} D \nabla \varphi_2 \cdot \nabla \varphi_1 + \int_{\Omega} (w \cdot \nabla \varphi_2) \varphi_1 \right) v_2 = A_{21} v_2$$

Now the \sum means add up all of the v_j s for the same i ,

$$\therefore A_{11} v_1 + A_{12} v_2 + A_{13} v_3 + A_{14} v_4 = f_1 \quad i=1$$

Do the same for $i = 2, 3, 4$ and you get a matrix equation!

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$