STUDY AND IMPLEMENTATION OF A MONOTONE FINITE ELEMENT SCHEME FOR CONVECTION-DIFFUSION EQUATIONS

DANA TOBIN, BRADLEY THOMPSON, AND LUDMIL ZIKATANOV

Abstract. These notes are for studying and exploring a monotone finite element discretization for convection diffusion equations with tensor coefficients. This scheme has applications in modeling semiconductor devices (the well known drift diffusion model) and also in modeling the transport of chemical species in an electrically charged fluid (Nernst-Planck equations).

1. Introduction

This project is on studying the Edge Average Finite Element (EAFE) [1, 2] scheme and incorporating it as part of the iFEM package [3]. The focus of this research is the numerical solution of convection diffusion equations of the following form

$$\begin{cases} \mathcal{L}u \equiv -\nabla \cdot (D(x)\nabla u + \boldsymbol{\beta}(x)u) = f(x), & x \in \Omega, \\ u = 0, & x \in \partial\Omega. \end{cases} \tag{1}$$

We assume the following: D is a matrix valued piece-wise constant function and $D(x)$ is symmetric and positive definite for all $x \in \Omega$; $\boldsymbol{\beta}$ is a given velocity field (also piece-wise constant); and a given right hand side $f(x)$.

We study finite element discretizations for such equations. The main part of the research is on the implementation of the aforementioned Edge Average Finite Element (EAFE) scheme proposed in [1, 2]. This scheme is a generalization of finite difference 1D (one dimensional) discretization of current continuity equations in semiconductor device modeling [4].

As part of the project, we also briefly studied standard Finite Difference and Finite Element methods. Some of these studies are summarized in the Appendix.

1.1. **EAFE scheme in 1D.**

We now look for an approximation of the one-dimensional case for a boundary value problem of the form

$$-(u' + \beta u)' = f, \quad u(0) = 0, \quad u(1) = 0.$$

We can also write this equation in the form

$$-[J(u)]' = f, \quad \text{where the flux } J(u) \text{ is:} \quad J(u) = u' + \beta u.$$

We split the interval $[0, 1]$ into n subintervals and aim to approximate the flux as a constant on every interval. That is to say that on every interval we have

$$u' + \beta u = c, \quad \text{where } c \text{ is an unknown constant.}$$

Let us now change the variables to where $u = e^{-\beta x} w$. We have

$$u' = -\beta e^{-\beta x} w + e^{-\beta x} w'.$$

Thus

$$u' + \beta u = e^{-\beta x} w' = c.$$

These notes are based on the 2013 summer research as a part of the Computational Mathematics for Undergraduate Students in the Department of Mathematics, Penn State, University Park, PA, 16802, USA; see `http://sites.psu.edu/cmus2013/` for details.

Substituting w from above back into this equation we get

$$e^{-\beta x}(ue^{\beta x})' = c, \quad \text{or} \quad (ue^{\beta x})' = ce^{\beta x}.$$

Integrating both sides from x_i to x_{i+1} yields

$$e^{\beta x_{i+1}}u_{i+1} - e^{\beta x_i}u_i = c(\frac{1}{\beta}e^{\beta x_{i+1}} - \frac{1}{\beta}e^{\beta x_i}).$$

After some simplification, we get the unknown constant c to be

$$c = \frac{\beta}{e^{\beta x_{i+1}} - e^{\beta x_i}}(e^{\beta x_{i+1}}u_{i+1} - e^{\beta x_i}u_i).$$

Since u is piecewise linear and continuous, we can take the inner product with a test function v, where v is the hat function $\phi_i(x)$ to obtain

$$\int_{x_i}^{x_{i+1}} Jv'\, dx = c\int_{x_i}^{x_{i+1}} v'\, dx.$$

The slope of $\phi(x_i)$ is negative between x_i and x_{i+1}, so this simplifies to $\frac{-1}{h}hc = -c$. The i-th row of the stiffness matrix is then simply $-c$ (which we will denote as $(-J)$). This equates to

$$\frac{\beta e^{\beta x_i}}{e^{\beta x_{i+1}} - e^{\beta x_i}}u_i - \frac{\beta e^{\beta x_{i+1}}}{e^{\beta x_{i+1}} - e^{\beta x_i}}u_{i+1}.$$

Simplifying this further and multiplying both the top and bottom by h gives

$$\frac{\beta h}{h(e^{\beta h} - 1)}u_i - \frac{\beta h}{h(1 - e^{-\beta h})}u_{i+1}.$$

Substituting s for βh yields

$$\frac{1}{h}(\frac{s}{e^s - 1}u_i - \frac{s}{1 - e^{-s}}u_{i+1}).$$

The terms in front of u_i and u_{i+1} are in fact values of the Bernouilli function, $B(s) = \frac{s}{e^s - 1}$, and hence we get $\frac{1}{h}(B(s)u_i - B(-s)u_{i+1})$. Similar calculations for the interval $[x_{i-1}, x_i]$ then give

$$-\frac{1}{h}(B(s)u_{i-1} - B(s)u_i).$$

Summing these up we get

$$\frac{1}{h}(-B(s)u_{i-1} + (B(s) + B(-s))u_i - B(-s)u_{i+1} = f_i.$$

2. EAFE in 2 Dimensions

2.1. **Mesh Structure.** In this section we will introduce the basic and auxiliary structures of representing a mesh. The code will follow the example of the unit square in two dimensions. The presentation here closely follows the Introduction notes given in [3] and on the iFEM website: `http://www.math.uci.edu/~chenlong/iFEM/doc/html/meshdoc.html`.

2.2. **Basic Mesh Structure.** In MatLab, `node(1:N,1:d)` and `elem(1:NT,1:d+1)` represent a d-dimensional triangulation embedded in $\mathbb{R}^d$. For a triangle t, `elem(t,1:3)`indicates the three vertices. By convention, vertices are ordered such that the signed volume is positive, that is, in two dimensions, triangle vertices are ordered counterclockwise. The function `fixorientation` will permute the vertices of the triangles if necessary. The *label* function permutes vertices such that `elem(t,2:3)` is the longest edge of `t`, while the `label3` function permutes vertices such that `elem(t,1:2)` is the longest edge of t. The following code creates the mesh for a unit square.

```
node=[1,0; 1,1; 0,1; -1,1; -1,0; -1,-1; 0,-1; 1,-1; 0,0];
elem=[1,2,9; 3,9,2; 9,3,5; 4,5,3; 8,7,1; 7,1,9; 9,7,6; 5,9,6];
elem=fixorientation(node,elem);
figure(1); showmesh(node,elem)
axis on
findnode(node)
findelem(node,elem)
```

To obtain a fine mesh for this domain we apply uniform refinement three times in the loop

```
for i=1:3; [node,elem]=uniformrefine(node,elem); end
```

This creates a refined mesh structure which will help to increase the accuracy of approximating the solution. In FIgure 1 we show the basic mesh and the refined mesh.

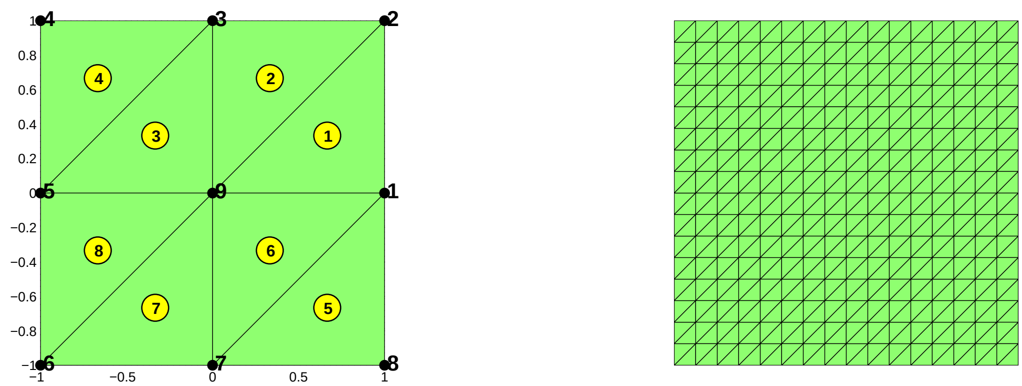

Figure 1. Basic Mesh Structure of Unit Square (left); Refined Mesh Structure of Unit Square (right)

2.3. **Auxiliary Mesh Structure.** The iFEM function `auxstructure.m` gives the auxiliary mesh structure for the domain from the array `elem` as the input. This function constructs the map between elements, edges, and also outputs the boundary information. The convention used in this function follows `a2b` such that it is based on the map from `a` to `b`. On the other hand, `edge(1:NE,1:2)` stores the indices of the starting and ending points of the edges. The `elem` matrix is the correspondence between triangles and vertices. The link from vertices to triangles to find all triangles containing vertex `v` is stored by the sparse matrix

```
t2v=sparse([1:NT,1:NT,1:NT],elem,1,NT,N)
```

The matrix means that the $\mathtt{i}^{th}$ node is the vertex of triangle `t` if `t2v(t,i)=1`. Also, `t2v(:,i)` gives all triangles containing this $\mathtt{i}^{th}$ node. In addition, `nodeStar=find(t2v(:,i))` finds the star of the $\mathtt{i}^{th}$ node, which is all the triangles that are connected to this same node. The cardinality of the node star is given by

```
valence=accumarray(elem(:),ones(3*NT,1),[N,1]).
```

Commands `sparse` and `accumarray` are used to lessen the computational time of using `for` loops for meshes with large number of elements. The array `neighbor(1:NT,1:3)` records neighboring triangles to each triangle by using

```
accumarray([[t1(ix),k1(ix)];[t2,k2]],[t2(ix);t1],[NT 3])
```

We next describe the weak formulation derive the EAFE scheme.

2.4. **Weak formulation in 2D.** The weak (variational) formulation of the problem (1) is: Find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega (D(x)\nabla u + \boldsymbol{\beta}(x)u)\cdot\nabla v dx = \int_\Omega f(x) v dx. \tag{2}$$

Usually, u will model concentration. An important property we want to preserve is that under certain conditions, the solution u is non-negative. More precisely,

$$\text{If } (\mathcal{L}u)(x) \geq 0 \text{ for all } x \in \Omega \text{ then } u(x) \geq 0 \text{ for all } x \in \Omega. \tag{3}$$

The above condition will be referred to as the *monotonicity property*, and it holds regardless of the size of $|\beta(x)|$ (even in the convection dominated case where $|\beta(x)| \gg 1$. We will refer to a finite element scheme that approximates the equation (1) and satisfies the above condition as a *monotone* finite element scheme. If the stiffness matrix is an M-matrix, then the scheme is monotone and the EAFE scheme satisfies this property under appropriate assumptions of the geometry of the mesh. In two dimensions, this amounts to requiring that the triangulation is a Delaunay triangulation.

We will first consider here the case when $D = I$ (the identity matrix). The more general case with any D is given at the end of Section 2.5.

The finite element spaces that we consider are the piece-wise linear and continuous and are defined on a triangulation with triangles in 2D and tetrahedrons in 3D. The entries of the stiffness matrix play a special role for the Laplace equation in what follows (i.e. equation (1) with $\boldsymbol{\beta} = 0$ and $D = I$). Let $\{\varphi_i\}_{i=1}^n$ be the set of "hat" functions.

We set

$$\begin{aligned} \omega_E^T(D) &= \int_T (D\nabla\varphi_i)\cdot\nabla\varphi_j, \quad \text{and} \\ \omega_E^T &= \omega_E^T(I) = \int_T \nabla\varphi_i\cdot\nabla\varphi_j = \frac{1}{d(d-1)}|\kappa_E^T|\cot\theta_E^T. \end{aligned}$$

Here, d is the spatial dimension. The entries in the stiffness matrix for the Laplace equation are denoted by ω_E and we have

$$\omega_E = \sum_{T\supset E}\omega_E^T = \frac{1}{n(n-1)}\sum_{T\supset E}|\kappa_E^T|\cot\theta_E^T \geq 0 \tag{4}$$

where $\sum_{T\supset E}$ means summation over all simplexes T containing E. We refer to Figure 2 for the notation used.

2.5. **Derivation of the EAFE scheme.** The derivation below is independent of spatial dimension and is therefore the same for 2D and 3D.

Given $T \in \mathcal{T}_h$, we introduce the following notation (see Figure 2):

- q_j $(1 \leq j \leq n+1)$: the vertices of T;
- E_{ij} or simply E: the edge connecting the two vertices q_i and q_j;
- F_j: the $(n-1)$-dimensional simplex opposite to the vertex q_j;
- θ_{ij}^T or θ_E^T: the angle between the faces F_i and F_j;
- κ_E^T: $F_i \cap F_j$, the $(n-2)$-dimensional simplex opposite to the edge E;
- $\delta_E\phi = \phi(q_i) - \phi(q_j)$, for any continuous function ϕ on $E = E_{ij}$;

- $\tau_E = \delta_E\, x = q_i - q_j$, a directional vector of E.

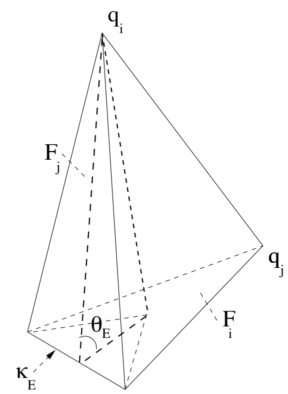

FIGURE 2. Example of a figure: A tetrahedron

Given any edge E, we introduce a function ψ_E defined locally on E (up to an arbitrary constant) by the relation $\dfrac{\partial \psi_E}{\partial \tau_E} = \dfrac{1}{|\tau_E|}(\boldsymbol{\beta} \cdot \tau_E)$. Here, $\partial/\partial\tau_E$ denotes the tangential derivative along E. As a basis for our derivation we use (5) below. For u with sufficiently many derivatives (such that the expressions below make sense) we have that

$$\delta_E(e^{\psi_E} u) = \frac{1}{|\tau_E|} \int_E e^{\psi_E} (J(u) \cdot \tau_E) ds, \tag{5}$$

where $J(u) = \nabla u + \boldsymbol{\beta} u$.

To derive equation (5), we take the dot product of the relation $J(u) = \nabla u + \boldsymbol{\beta} u$ with the directional vector τ_E and obtain

$$(\nabla u \cdot \tau_E) + (\boldsymbol{\beta} \cdot \tau_E) u = (J(u) \cdot \tau_E).$$

Now using the definition of ψ_E in given above we get

$$e^{-\psi_E} \frac{\partial (e^{\psi_E} u)}{\partial \tau_E} = \frac{1}{|\tau_E|} (J(u) \cdot \tau_E). \tag{6}$$

The equality (5) follows from (6) after integration over edge E.

Let $\mathcal{H}(\boldsymbol{\beta})$ be the harmonic average of $e^{-\psi_E}$ over E, defined as:

$$\mathcal{H}(\boldsymbol{\beta}) = \left[\frac{1}{|\tau_E|} \int_E e^{\psi_E} ds \right]^{-1}. \tag{7}$$

First we approximate $J(u)$ over each simplex T by a constant vector $J_T(u)$. Then from (5) we have that

$$J_T(u) \cdot \tau_E \approx \mathcal{H}(\boldsymbol{\beta}) \delta_E (e^{\psi_E} u). \tag{8}$$

From all this we get that for any $v \in V_h$, if $J_T(u)$ is a constant on T we have

$$\int_T J_T(u) \cdot \nabla v dx = \sum_E \omega_E^T (J_T(u) \cdot \tau_E) \delta_E v \approx \sum_{E \subset T} \omega_E^T \mathcal{H}(\boldsymbol{\beta}) \delta_E (e^{\psi_E} u) \delta_E v. \tag{9}$$

The finite element discretization is: Find $u \in V_h$ such that

$$\sum_{T \in \mathcal{T}_h} \left\{ \sum_{E \subset T} \omega_E^T \mathcal{H}(\boldsymbol{\beta_T}) \delta_E (e^{\psi_E} u) \delta_E v \right\} = \int_\Omega f v \quad \text{for all} \quad v \in V_h. \tag{10}$$

Because we have assumed that $\boldsymbol{\beta}$ is a constant on every triangle T, the system of linear equations for $\{u_i\}$ has the form

$$\sum_T \sum_{E\in T} \omega_E^T \left[B(-\boldsymbol{\beta}_T\cdot\tau_E)u_i - B(\boldsymbol{\beta}_T\cdot\tau_E)u_j\right] = G_i, \tag{11}$$

where $G_i = \sum_{T\supset x_i}\int_T f\varphi_i dx$ and $\tau_E = x_i - x_j$. The inner summation is over all edges in an element T and $B(s)$ is the Bernoulli function

$$B(s) = \begin{cases} \dfrac{s}{e^s-1}, & s\neq 0,\\ 1, & s=0. \end{cases}$$

Note that if $s\to+\infty$, then $B(s)$ approaches zero exponentially and $B(-s)$ behaves like s when $s\to\infty$.

In the case where $D\neq I$ we have the following discrete problem (according to [2]):

$$\sum_{T\in\mathcal{T}_h}\left\{\sum_{E\subset T}\omega_E^T(D)\mathcal{H}(\boldsymbol{\beta_T})\delta_E(e^{\psi_E}u)\delta_E v\right\} = \int_\Omega fv \quad \text{for all} \quad v\in V_h. \tag{12}$$

In 2D, the geometric conditions which make the FE scheme monotone are that the triangulation has to be Delaunay with respect to the metric given by D.

2.6. **Matrix Formulation.** Given a triangle with vertices x_1, x_2 and x_3, and with edges (12), (13) and (23) we denote

$$\tau_{12} = \boldsymbol{x}_1 - \boldsymbol{x}_2, \quad \tau_{13} = \boldsymbol{x}_1 - \boldsymbol{x}_3, \quad \tau_{23} = \boldsymbol{x}_2 - \boldsymbol{x}_3.$$

For $1\le i<j\le 3$ and with the Bernoulli function $B(\cdot)$, we have

$$b_{ij} = -B(\boldsymbol{\beta}\cdot\tau_{ij}), \quad b_{ji} = -B(-\boldsymbol{\beta}\cdot\tau_{ij}), \quad 1\le i<j\le 3.$$

Each off-diagonal element in the local stiffness matrix then is of the form

$$a_{ij}^T = \omega_{ij}(D)b_{ij}, \quad i\neq j, \quad i=1:3, \quad j=1:3,$$

where

$$\omega_{ij}(D) = -\int_\Omega \nabla\phi_i D\nabla\phi_j \, d\Omega \tag{13}$$

and hence

$$\begin{pmatrix} a_{11}^T & -\omega_{12}(D)B(\boldsymbol{\beta}\cdot\tau_{12}) & -\omega_{13}(D)B(\boldsymbol{\beta}\cdot\tau_{13})\\ -\omega_{12}(D)B(-\boldsymbol{\beta}\cdot\tau_{12}) & a_{22}^T & -\omega_{23}(D)B(\boldsymbol{\beta}\cdot\tau_{23})\\ -\omega_{13}(D)B(-\boldsymbol{\beta}\cdot\tau_{13}) & -\omega_{23}(D)B(-\boldsymbol{\beta}\cdot\tau_{23}) & a_{33}^T \end{pmatrix} \tag{14}$$

Since the column sum in the local matrix must be zero, we have for the diagonal elements that

$$a_{11}^T = -(a_{21}^T + a_{31}^T), \quad a_{22}^T = -(a_{12}^T + a_{32}^T), \quad a_{33}^T = -(a_{13}^T + a_{23}^T),$$

and this gives

$$\begin{aligned} a_{11}^T &= \omega_{12}B(-\beta\cdot\tau_{12}) + \omega_{13}B(-\beta\cdot\tau_{13})\\ a_{22}^T &= \omega_{12}B(\beta\cdot\tau_{12}) + \omega_{23}B(-\beta\cdot\tau_{23})\\ a_{33}^T &= \omega_{12}B(\beta\cdot\tau_{13} + \omega_{23}B(\beta\cdot\tau_{23}) \end{aligned}$$

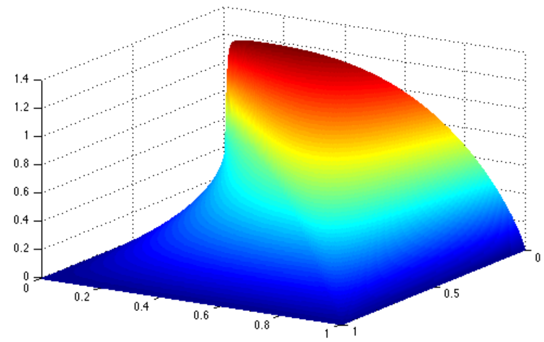

FIGURE 3. Solution to convection-Diffusion equation: Example 1

3. Numerical examples

Our computational domain is the unit square $\Omega = (0,1) \times (0,1)$. We consider the equation

$$-\nabla \cdot (D\nabla u + \boldsymbol{\beta} u) = 1, \tag{15}$$

and set $\boldsymbol{\beta} = \begin{pmatrix} -y \\ x \end{pmatrix}$.

Example 1: In our first example we have a simple choice for D where

$$D = \varepsilon I = \varepsilon \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \varepsilon = 10^{-5}.$$

The graph of the solution is shown in Figure 3. We may think of ε in the above and subsequent equations as the mobility of the species whose concentration is $u(x,y)$. If the species is a fluid, then ε is an approximation to the viscosity. Altering the ε parameter changes the model properties of how the concentration disperses if the fluid is more or less viscous (larger values of ε indicate higher viscosity while smaller values indicate lower viscosity). To visualize this, one can consider the differences in how molasses (a very viscous fluid) would disperse over an area versus a less viscous fluid (i.e. water) given identical initial velocities and boundary conditions.

Example 2: Our second example changes the parameters of D so that we introduce some anisotropy where

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 10^{-4} \end{pmatrix}$$

The solution to this variation is shown in Figure 4 (left).

Example 3: In this example we take

$$\begin{aligned} D &= 10^{-4} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad 0 < x < \frac{1}{2}; \\ D &= 10^{-2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \frac{1}{2} < x < 1. \end{aligned}$$

The solution to this example is shown in Figure 4 (right).

In all of these examples we see that the EAFE scheme correctly captures the behavior of the solutions and that there are no spurious oscillations in the numerical solution. This result shows that the scheme is monotone and applies for the examples we have chosen here.

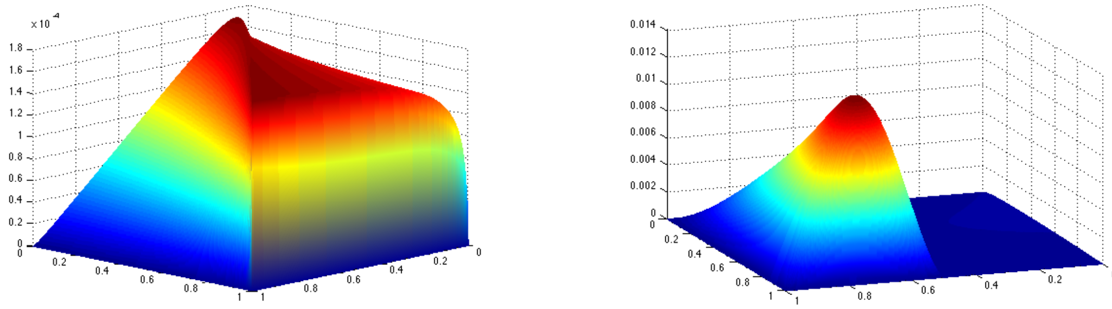

FIGURE 4. Solution plot: Example 2 (left) and Example 3 (right)

4. Our additions to iFEM

The main modifications in implementing the above scheme in the iFEM package are as follows:

- We have more general coefficients by modifying iFEM so that a tensor diffusion coefficient is possible (i.e. D is a matrix). This modification works for both diffusion and convection diffusion equations.
- We have also modified the assembling routines in iFEM to include the EAFE scheme. Our implementation avoids the use of for loops in computing local stiffness matrices.

Appendix A. Standard methods

A.1. **Finite difference methods in 1D.** We begin by looking at a simple Finite Difference Method and later introduce the Finite Element Method. Let us first look at an idealized practical example that is typically modeled by using this method. Consider a string pinned at both ends and acted upon by a force $f(x)$ causing displacement $u(x)$. Hooke's Law gives us that the displacement satisfies $-(ku')' = f$, where k is a constant material coefficient that depends upon the properties of the string. To simplify notation for this example, we take a homogenous material ($k = 1$) and set the ends of the string such that the string is on the interval $(0, 1)$. The ordinary differential equation and the boundary conditions describing this system then are

$$-u'' = f, \quad u(0) = u(1) = 0.$$

To approximate this equation and obtain a solution, we break the interval $[0, 1]$ into n subintervals of length h. This is to say that $[0, 1] = \cup_{i=1}^{n}[x_{i-1}, x_i]$ has some value $x_1, x_2, ..., x_n$ such that each u_k is approximately $u(x_k)$. We assume that $x_{k+1} = x_k + h$. Doing a Taylor Series Expansion we get that

$$\begin{aligned}
u(x_{k-1}) = u(x_k - h) &= u(x_k) - hu'(x_k) + \frac{h^2}{2!}u''(x_k) - (\frac{h^3}{3!})u'''(x_k) + (\frac{h^4}{4!})u^{(iv)}(x_k) + \dots, \\
2u(x_k) &= 2u(x_k), \\
u(x_{k+1}) &= u(x_k) + hu'(x_k) + (\frac{h^2}{2!})u''(x_k) + (\frac{h^3}{3!})u'''(x_k) + (\frac{h^4}{4!})u^{(iv)}(x_k) + \dots.
\end{aligned}$$

Now let us look at $-u(x_{k-1}) + 2u(x_k) - u(x_{k+1})$. From the above expansion we can see that $u(x_k), u'(x_k)$, and $u'''(x_k)$ are cancelled out, leaving $h^2u''(x_k)$ and $\frac{1}{12}h^4u^{(iv)}(x_k)$. Thus we have

$$2u_k - u_{k-1} - u_{k+1} = h^2u''(x_k) + \frac{1}{12}h^4u^{iv)}(x_k) + \dots$$

We are drop the higher order terms from this expression (any term that goes to zero faster than h^2 as $h \to 0$), so we are left with

$$-u''(x_k) \approx \frac{-u_{k-1} + 2u_k - u_{k+1}}{h^2} = f(x_k), \quad \text{for} \quad k = 1, \ldots, n-1.$$

Let us now consider some values of this discretization for integers k near the boundary:

$$\begin{aligned}
k = 1: &\qquad f_1 = 2u_1 - u_2, \quad (u_0 = 0)\\
k = 2: &\qquad f_2 = -u_1 + 2u_2 - u_3\\
\cdots &\\
k = n-1: &\qquad f_{n-1} = u_{n-2} + 2u_{n-1}, \quad (u_n = 0)
\end{aligned}$$

The resulting $[(n-1) \times (n-1)]$ matrix for this linear system of equations is then

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

With this matrix we can now solve the equation $Au = f$ for u and obtain values which approximate the solution.

We have shown how to discretize u'', and will now the discretization of u'. We again consider the Taylor expansions shown above, but now look at three expressions that simplify to u'. Three methods for approximating u' are as follows:

$$\begin{aligned}
u' &\approx \frac{u(x_k) - u(x_{k-1})}{h}, \quad \text{(Backward Difference)},\\
u' &\approx \frac{u(x_{k+1}) - u(x_k)}{h}, \quad \text{(Forward Difference)},\\
u' &\approx \frac{u(x_{k+1}) - u(x_{k-1})}{2h}, \quad \text{(Central Difference)}.
\end{aligned}$$

These techniques can be applied to a convection diffusion equation in 1D of the form

$$-\epsilon u'' + u' = f,$$

An instructive example is to test the above discretizations on this equation with $f = 0$ and boundary conditions of $u(0) = 0$ and $u(1) = 1$.

A.2. **Variational Formulation and Finite Element Method in 1D.** Let us consider the convection-diffusion equation as our example for the Finite Element Method in one dimension. We have

$$-\epsilon u'' + u' = f, \quad \text{where} \quad u(0) = u(1) = 0.$$

Multiplying through by a test function $v(x)$ and integrating yields

$$-\epsilon \int_0^1 u'' v\, dx + \int_0^1 u' v\, dx = \int_0^1 f v\, dx.$$

Integration by parts gives the variational form of the original equation as follows: Find u such that $u(0) = u(1) = 0$ and satisfies the equation

$$\epsilon \int_0^1 u'v'\,dx - \int_0^1 uv'\,dx = \int_0^1 fv\,dx.$$

To approximate this variational formulation we again divide the interval $[0, 1]$ into n subintervals and approximate u as a piecewise linear and continuous function. The solution is represented by a series of broken lines connecting the values at each break point.

Any piecewise linear function can be written as $v(x) = \sum \alpha_i \phi_i(x)$, where $\{\phi_i(x)\}_{i=1}^{n-1}$ are hat functions and are the basis for the space of piecewise linear functions. We take these hat functions to also be our test function. The discretization is then to find u such that

$$\int_0^1 (\epsilon u' - u)\phi_i'\,dx = \int_0^1 f\phi_i\,dx$$

Since u is also piecewise linear and continuous, it follows that

$$u = \sum_{j=1}^{n} u_j \phi_j(x)$$

for unknown coefficients u_j. The discrete variational form then becomes

$$\int_0^1 [\epsilon(\sum_{j=1}^{n-1} u_j\phi_j)' - \sum_{j=1}^{n-1} u_j\phi_j]\phi_i'\,dx = \sum_{j=1}^{n}[\int_0^1 \phi_i'(x)(\epsilon\phi_j'(x) - \phi_j(x))\,dx]u_j$$

We will call the term inside the bracket a_{ij}. Evaluating the integral yields 0 for every term except for

$$\begin{aligned} a_{i,i-1} &= \frac{-\epsilon}{h} - \frac{1}{2}, \\ a_{i,i} &= \frac{2\epsilon}{h}, \\ a_{i,i+1} &= \frac{-\epsilon}{h} + \frac{1}{2}. \end{aligned}$$

This leaves us with a sparse matrix that contains $\frac{2\epsilon}{h}$ on the 0 diagonal, $\frac{-\epsilon}{h} - \frac{1}{2}$ on the -1 diagonal, and $\frac{-\epsilon}{h} + \frac{1}{2}$ on the 1 diagonal. $A = (a_{ij})$ is the stiffness matrix in the equation $Au = f$ and f is a column vector of h for every entry if $f(x) = 1$.

A.3. **Standard Finite Element Method in 2D.** Often, the 1-D version does not model two or three dimensional physical phenomena very well, so we turn our attention to 2D FEM. In what follows, the Laplace operator, $\Delta = \mathrm{div}(\nabla)$ will be used often. The partial differential equation (PDE) of interest is then

$$-\Delta u + \beta \cdot \bigtriangledown u = f(x, y) \tag{16}$$

in a domain $\Omega\epsilon\Re^2$. In our case, Ω is the unit square oriented with the bottom left corner at the origin. In order to find the variational form of the equation in the 2-D case, we must do a multivariate integration by parts, similar to the 1-D case. Doing so gives us

$$\int_\Omega u\frac{\partial v}{\partial x}\,d\Omega = -\int_\Omega v\frac{\partial u}{\partial x}\,d\Omega + \int_{\partial\Omega} uvn_x\,d\gamma \tag{17}$$

where n_x is the vector normal to the boundary.

Two related equations that can be derived almost directly from this one are

$$\int_\Omega Div(v)u\,d\Omega = -\int_\Omega v\cdot\Delta u\,d\Omega + \int_{\partial\Omega}(v\cdot n)u\,d\gamma, \tag{18}$$

$$\int_\Omega(-\Delta u)v\,d\Omega = \int_\Omega \bigtriangledown u\cdot\bigtriangledown v\,d\Omega - \int_{\partial\Omega}(\bigtriangledown u\cdot n)\cdot v\,d\gamma \tag{19}$$

We now turn back to our equation (16) and write it in an alternate form of $Div(J) = f$ where $J = \bigtriangledown u - \beta u$. The inner product of this with our test function v is then

$$-\int_\Omega Div(J)\cdot v\,d\Omega = \int_\Omega J\cdot\bigtriangledown v\,d\Omega + \int_{\partial\Omega}(J\cdot n)\cdot v\,d\gamma$$

We seek u that is zero on the boundary and such that

$$\int_\Omega(\bigtriangledown u - \beta u)\cdot\bigtriangledown v\,d\Omega = \int_\Omega fv\,d\Omega.$$

As in the 1-Dimensional case, we find u as a combination of piecewise linear functions. We split Ω into triangles and approximate u by a function that is continuous and linear over each triangle. We use the analog of the 1-D hat function in 2D with functions that are 1 at a vertex of the triangulation and zero at every other vertex, as shown below where

$$\phi_i(x_i,y_i) = 1,\quad \phi_i(x_j,y_j) = 0,\quad \text{if} j\neq i,\quad u = \sum_{j=1}^n u_j\phi_j(x,y),$$

so that the discrete variational problem then amounts to finding u_j such that

$$\sum_{j=1}^n a_{ij}u_j = b_i,$$

where

$$\begin{aligned} a_{ij} &= \int_\Omega \bigtriangledown\phi_i(\bigtriangledown\phi_j - \beta\phi_j)\,d\Omega \\ &= \int_\Omega \bigtriangledown\phi_j\cdot\bigtriangledown\phi_i\,d\Omega - \int_\Omega(\beta\cdot\bigtriangledown\phi_i)\phi_j\,d\Omega \end{aligned} \tag{20}$$

Instead of taking the integral over the whole domain Ω, we divide it into triangles and then add up those integrals. Over one triangle, $\bigtriangledown\phi_i = \frac{a}{|a|^2}$, where a is the altitude vector at vertex i. The first term on the right side of this integral is then

$$\int_\Omega \bigtriangledown\phi_j\cdot\bigtriangledown\phi_i\,d\Omega = \int_{T_1}\frac{a_1}{|a_1|^2}\cdot\frac{a_2}{|a_2|^2}\,dxdy,$$

which, after some simplification, amounts to

$$\int_\Omega \bigtriangledown\phi_j\cdot\bigtriangledown\phi_i\,d\Omega = -\frac{1}{2}(\cot\alpha_{ij1} + \cot\alpha_{ij2}).$$

For the second term on the right hand side of (20), we again divide the domain into triangles

$$\int_{T_1}\phi_i(\beta\cdot\bigtriangledown\phi_j)\,dxdy + \int_{T_1}(\beta\cdot\bigtriangledown\phi_j)\,dxdy.$$

If β is constant, we have

$$\begin{aligned}\int_\Omega (\beta \cdot \bigtriangledown \phi_i)\phi_j \, d\Omega &= (\beta_{T_1} \cdot \frac{a_{T_1}}{|a|^2}) \int_{T_1} \phi_j \, dxdy + (\beta_{T_2} \cdot \frac{a_{T_2}}{|a|^2}) \int_{T_2} \phi_j \, dxdy \\ &= \frac{1}{3} \left[|T_1|(\beta_{T_1} \cdot \frac{a_{T_1}}{|a_{T_1}|^2}) + |T_2|(\beta_{T_2} \cdot \frac{a_{T_2}}{|a_{T_2}|^2}) \right].\end{aligned}$$

References

1. J. Xu and L. Zikatanov. A monotone finite element scheme for convection-diffusion equations. *Math. Comp.*, 68(228):1429–1446, 1999.
2. RD Lazarov and LT Zikatanov. An exponential fitting scheme for general convection-diffusion equations on tetrahedral meshes. *Comput. Appl. Math.(Obchysljuval'na ta prykladna matematyka, Kiev)*, 1(92):60–69, 2005. Also available as arXiv preprint 1211.0869, November, 2012.
3. L Chen. *i*FEM: An Integrated Finite Element Methods Package in MATLAB. Technical Report, University of California at Irvine, 2009.
4. D. Scharfetter and H. Gummel. Large-signal analysis of a silicon read diod oscilator. *IEEE Trans. Electron Devices*, ED-16(205):959–962, 1969.

Department of Mathematics, The Pennsylvania State University, University Park, PA, 16802