1 Basic Concepts

The finite element method provides a formalism for generating discrete (finite) algorithms for approximating the solutions of differential equations.

It should be thought of as a black box into which one puts the differential equation (boundary value problem) and out of which pops an algorithm for approximating the corresponding solutions.

We present a microcosm of the FEM restricted to one-dimensional problems.

1.1 Weak Formulation of Boundary Value Problems

Consider the two-point boundary value problem

$$\begin{aligned} -\frac{d^2u}{dx^2} &= f \text{ in } (0,1) \\ u(0) = 0, \qquad & u'(1) = 0. \end{aligned} \tag{1.1}$$

If u is the solution and v is any (sufficiently regular) function such that $v(0) = 0$, then integration by parts yields

$$\begin{aligned} (f,v) := \int_0^1 f(x)v(x)dx &= \int_0^1 -u''(x)v(x)dx \\ &= \int_0^1 u'(x)v'(x)dx =: a(u,v). \end{aligned} \tag{1.2}$$

Define $V = \{v \in L^2(0,1) : \quad a(v,v) < \infty \text{ and } v(0) = 0\}$. Then we can say that the solution u to (1.1) is characterized by

$$u \in V \quad \text{such that} \quad a(u,v) = (f,v) \qquad \forall v \in V, \tag{1.3}$$

which is called the *variational* or *weak* formulation of (1.1).

Why variational?

The relationship (1.3) is called "variational" because the function v is allowed to vary arbitrarily.

It may seem somewhat unusual at first, but it has a natural interpretation in the setting of *Hilbert spaces*.

(A Hilbert space is a vector space whose topology is defined using an inner-product.)

One example of a Hilbert space is $L^2(0,1)$ with inner-product $(\cdot,\cdot)$.

The space V may be viewed as a Hilbert space with inner-product $a(\cdot,\cdot)$, which was defined in (1.2).

Is it the same?

The central issue is that (1.3) still embodies the original problem (1.1). The following theorem verifies this under some simplifying assumptions.

Theorem 1.1 *Suppose* $f \in C^0([0,1])$ *and* $u \in C^2([0,1])$ *satisfy* (1.3). *Then* u *solves* (1.1).

The boundary condition $u(0) = 0$ is called *essential* as it appears in the variational formulation explicitly, i.e., in the definition of V. This type of boundary condition also frequently goes by the proper name "Dirichlet."

The boundary condition $u'(1) = 0$ is called *natural* because it is incorporated implicitly. This type of boundary condition is often referred to by the name "Neumann." We summarize the different kinds of boundary conditions encountered so far, together with their various names in the following table:

1.2 Naming conventions for two types of boundary conditions

Boundary Condition	**Variational Name**	**Proper Name**
$u(x) = 0$	essential	Dirichlet
$u'(x) = 0$	natural	Neumann

Table 1: Naming conventions for two types of boundary conditions.

The assumptions $f \in C^0([0,1])$ and $u \in C^2([0,1])$ in the theorem allow (1.1) to be interpreted in the usual sense. However, we will see other ways in which to interpret (1.1), and indeed the theorem says that the formulation (1.3) is a way to interpret it that is valid with much less restrictive assumptions on f. For this reason, (1.3) is also called a *weak* formulation of (1.1).

1.3 Ritz-Galerkin Approximation

Let $S \subset V$ be any (finite dimensional) subspace. Let us consider (1.3) with V replaced by S, namely

$$u_S \in S \quad \text{such that} \quad a(u_S, v) = (f, v) \qquad \forall v \in S. \tag{1.4}$$

It is remarkable that a discrete scheme for approximating (1.1) can be defined so easily.

This is only one powerful aspect of the Ritz-Galerkin method.

However, we first must see that (eqn:ohtuone) does indeed *define* an object. In the process we will indicate how (eqn:ohtuone) represents a (square, finite) system of equations for u_S.

These will be done in the following theorem and its proof.

Theorem 1.2 *Given $f \in L^2(0,1)$, (1.4) has a unique solution.*

The proof of Theorem 1.2 reveals important structure of the problem.

Let us write (1.4) in terms of a basis of S:

$$\{\phi_i : \ 1 \leq i \leq n\}$$

Let

$$u_S = \sum_{j=1}^{n} U_j \phi_j$$

Let

$$K_{ij} = a(\phi_j, \phi_i), F_i = (f, \phi_i)$$

for $i, j = 1, ..., n$.

Set $\mathbf{U} = (U_j), \mathbf{K} = (K_{ij})$ and $\mathbf{F} = (F_i)$.

Then (1.4) is equivalent to solving the (square) matrix equation

$$\mathbf{KU} = \mathbf{F}. \tag{1.5}$$

For a square system such as (1.5) we know that uniqueness is equivalent to existence, as this is a *finite dimensional* system.

To prove uniqueness, we show that nonuniqueness implies a contradiction.

Nonuniqueness would imply that there is a nonzero $\mathbf{V}$ such that $\mathbf{KV} = \mathbf{0}$.

Write $v = \sum V_j \phi_j$ and note that the equivalence of (1.4) and (1.5) implies that $a(v, \phi_j) = 0$ for all j.

Multiplying this by V_j and summing over j yields $0 = a(v, v) = \int_0^1 (v')^2(x)\, dx$, from which we conclude that $v' \equiv 0$.

Thus, v is constant, and, since $v \in S \subset V$ implies $v(0) = 0$, we must have $v \equiv 0$. Since $\{\phi_i : 1 \leq i \leq n\}$ is a basis of S, this means that $\mathbf{V} = \mathbf{0}$.

Thus, the solution to (1.5) must be unique (and hence must exist).

Therefore, the solution u_S to (1.4) must also exist and be unique.

The matrix $\mathbf{K}$ is often referred to as the *stiffness* matrix, a name coming from corresponding matrices in the context of structural problems.

It is symmetric, since the *energy* inner-product $a(\cdot, \cdot)$ is symmetric.

It is also *positive definite*, since

$$\sum_{i,j=1}^{n} k_{ij} v_i v_j = a(v, v) \quad \text{where} \quad v = \sum_{j=1}^{n} v_j \phi_j .$$

Clearly, $a(v, v) \geq 0$ for all (v_j) and $a(v, v) = 0$ was already "shown" to imply $v \equiv 0$ in the proof of Theorem 1.5.

1.4 Piecewise Polynomial Spaces – The Finite Element Method

Let $0 = x_0 < x_1 < ... < x_n = 1$ be a partition of $[0, 1]$, and let S be the linear space of functions v such that

- i) $v \in C^0([0, 1])$
- ii) $v|_{[x_{i-1}, x_i]}$ is a linear polynomial, $i = 1, ..., n$, and
- iii) $v(0) = 0$.

We will see later that $S \subset V$. For each $i = 1, .., n$ define ϕ_i by the requirement that $\phi_i(x_j) = \delta_{ij} =$ the Kronecker delta, as shown in Fig. 1.

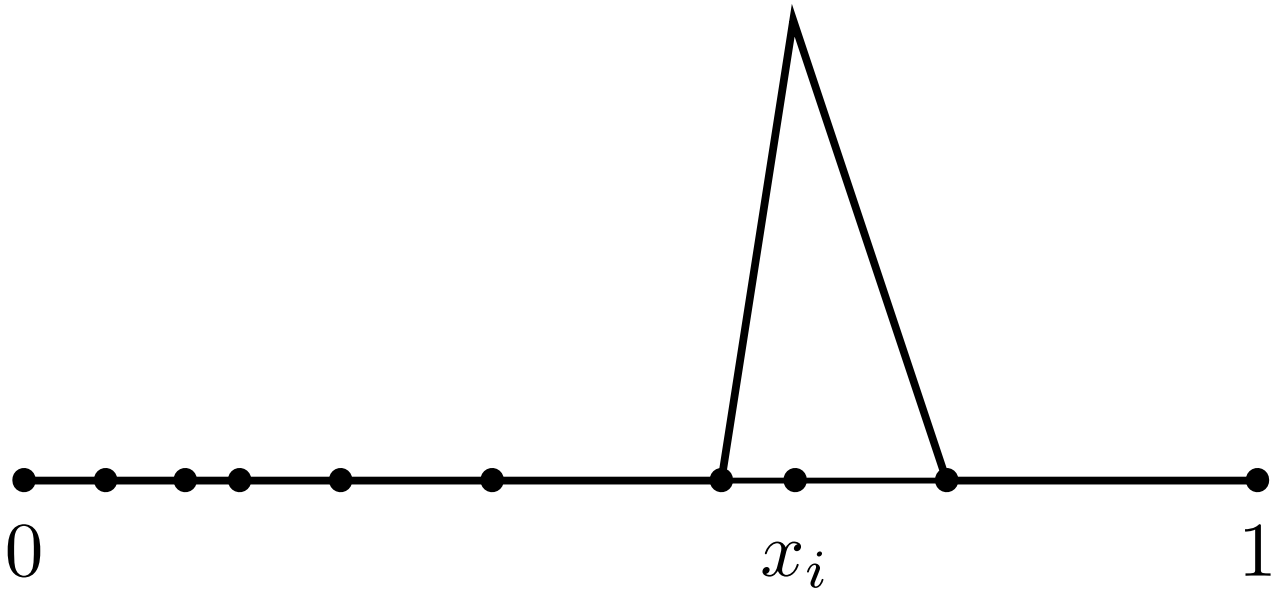

Figure 1: piecewise linear basis function ϕ_i

Lemma 1.1 $\{\phi_i : 1 \leq i \leq n\}$ *is a basis for* S.

$\{\phi_i\}$ is called a **nodal** basis for S, and $\{v(x_i)\}$ are the **nodal values** of a function v. (The points $\{x_i\}$ are called the **nodes**.) The set $\{\phi_i\}$ is linearly independent since $\sum_{i=1}^{n} c_i \phi_i(x_j) = 0$ implies $c_j = 0$. To see that it spans S, consider the following:

Definition 1.1 *Given* $v \in C^0([0,1])$, *the* **interpolant** $v_I \in S$ *of* v *is determined by* $v_I := \sum_{i=1}^{n} v(x_i)\phi_i$.

Clearly, the set $\{\phi_i\}$ spans S if the following is true.

Lemma 1.2 $v \in S \Rightarrow v = v_I$.

$v - v_I$ is linear on each $[x_{i-1}, x_i]$ and zero at the endpoints, hence must be identically zero.

The interpolant defines a linear operator $\mathcal{I}: C^0([0,1]) \to S$ where $\mathcal{I}v = v_I$. Lemma 0.4.4 says that $\mathcal{I}$ is a *projection* (i.e., $\mathcal{I}^2 = \mathcal{I}$).

1.5 Relationship to Difference Methods

The stiffness matrix $\mathbf{K}$ as defined in (0.2.3), using the basis $\{\phi_i\}$ described above, can be interpreted as a difference operator.

Let $h_i = x_i - x_{i-1}$.

Then the matrix entries $K_{ij} = a(\phi_i, \phi_j)$ can be easily calculated to be

$$(0.5.1) \quad K_{ii} = h_i^{-1} + h_{i+1}^{-1}, K_{i,i+1} = K_{i+1,i} = -h_{i+1}^{-1} \qquad (i = 1, ..., n-1)$$

and $K_{nn} = h_n^{-1}$ with the rest of the entries of $\mathbf{K}$ being zero.

Similarly, the entries of $\mathbf{F}$ can be approximated if f is sufficiently smooth:

$$(0.5.2) \qquad (f, \phi_i) = \frac{1}{2}(h_i + h_{i+1})(f(x_i) + \mathcal{O}(h))$$

where $h = \max h_i$.

This follows easily from Taylor's Theorem since the integral of ϕ_i is $(h_i + h_{i+1})/2$. Note that the error is *not* $\mathcal{O}(h^2)$ unless $1 - (h_i/h_{i+1}) = \mathcal{O}(h)$.

Thus, the $i-th$ equation of $\mathbf{KU} = \mathbf{F}$ (for $1 \leq i \leq n-1$) can be written as

$$(0.5.3) \quad \frac{-2}{h_i + h_{i+1}} \left[\frac{U_{i+1} - U_i}{h_{i+1}} - \frac{U_i - U_{i-1}}{h_i} \right] = \frac{2(f, \phi_i)}{h_i + h_{i+1}} = f(x_i) + \mathcal{O}(h).$$

The difference operator on the left side of this equation can also be seen to be an $\mathcal{O}(h)$ accurate approximation to the differential operator $-d^2/dx^2$ (and *not* $\mathcal{O}(h^2)$ accurate in the usual sense unless $1 - h_i/h_{i+1} = \mathcal{O}(h)$.)

For a uniform mesh, the equations reduce to the familiar difference equations

$$(0.5.4) \qquad -\frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} = f(x_i) + \mathcal{O}(h^2)$$

which are well known to be second-order accurate.

Characteristics of the finite element formalism

Being based on the variational formulation of boundary value problems, it is quite systematic, handling different boundary conditions with ease; one simply replaces infinite dimensional spaces with finite dimensional subspaces.

What results, as in (0.5.3), is the same as a finite difference equation, in keeping with the *dictum* that different numerical methods are usually more similar than they are distinct.

However, we are able to derive very quickly the convergence properties of the finite element method.

Finally, the notation for the discrete scheme is quite compact in the finite element formulation.

This can be utilized to automate coding the algorithm via appropriate software support.

1.6 Computer Implementation of Finite Element Methods

One key to the success of the finite element method, as developed in engineering practice, was the systematic way that computer codes could be implemented.

One important step in this process is the *assembly* of the inner-product $a(u, v)$ by summing its constituent parts over each sub-interval, or *element*, which are computed separately.

This is facilitated through the use of a numbering scheme called the *local-to-global* index.

This index, $i(e, j)$, relates the local node number, j, on a particular element, e, to its position in the global data structure.

In our one-dimensional example with piecewise linear functions, this index is particularly simple: the "elements" are based on the intervals $I_e := [x_{e-1}, x_e]$ where e is an integer in the range $1, \ldots, n$ and

$$i(e, j) \ := \ e + j - 1 \text{ for } e = 1, \ldots, n \text{ and } j = 0, 1.$$

That is, for each element there are two nodal parameters of interest, one corresponding to the left end of the interval ($j = 0$) and one at the right ($j = 1$). Their relationship is represented by the mapping $i(e, j)$.

We may write the interpolant of a continuous function for the space of all piecewise linear functions (no boundary conditions imposed) via

$$f_I \;:=\; \sum_{e} \sum_{j=0}^{1} f(x_{i(e,j)}) \phi_j^e \tag{0.6.1}$$

where $\left\{ \phi_j^e \;\ni\; j = 0, 1 \right\}$ denotes the set of basis functions for linear functions on the single interval $I_e = [x_{e-1}, x_e]$:

$$\phi_j^e(x) \;=\; \phi_j \left((x - x_{e-1})/(x_e - x_{e-1}) \right)$$

where

$$\phi_0(x) \;:=\; \begin{cases} 1 - x & x \in [0, 1] \\ 0 & otherwise \end{cases}$$

and

$$\phi_1(x) \;:=\; \begin{cases} x & x \in [0, 1] \\ 0 & otherwise. \end{cases}$$

Note that we have related all of the "local" basis functions ϕ_j^e to a fixed set of basis functions on a "reference" element, $[0,1]$, via an affine mapping of $[0,1]$ to $[x_{e-1}, x_e]$. (By definition, the local basis functions, ϕ_j^e, are extended by zero outside the interval I_e.)

The expression (0.6.1) for the interpolant shows (cf. Lemma 0.4.4) that any piecewise linear function f (no boundary conditions imposed) can be written in the form

$$f := \sum_e \sum_{j=0}^{1} f_{i(e,j)} \phi_j^e \tag{0.6.2}$$

where $f_i = f(x_i)$ for all i. In particular, the cardinality of the image of the index mapping $i(e,j)$ is the dimension of the space of piecewise linear functions. Note that the expression (0.6.2) represents f incorrectly at the nodal points, but this has no effect on the evaluation of multilinear forms involving integrals of f.

The bilinear forms defined in (1.2) can be easily evaluated (assembled) using this representation as well. For example,

$$a(v,w) \;=\; \sum_e a_e(v,w)$$

where the "local" bilinear form is defined (and evaluated) via

$$\begin{aligned} a_e(v,w) \;&:=\; \int_{I_e} v'w'\,dx \\ &=\; (x_e - x_{e-1})^{-1} \int_0^1 \left(\Sigma_j v_{i(e,j)}\phi_j\right)' \left(\Sigma_j w_{i(e,j)}\phi_j\right)'\,dx \\ &=\; (x_e - x_{e-1})^{-1} \begin{pmatrix} v_{i(e,0)} \\ v_{i(e,1)} \end{pmatrix}^t \mathbf{K} \begin{pmatrix} w_{i(e,0)} \\ w_{i(e,1)} \end{pmatrix}. \end{aligned} \tag{1.6}$$

Here, the *local stiffness matrix*, $\mathbf{K}$, is given by

$$\mathrm{K}_{i,j} \;:=\; \int_0^1 \phi'_{i-1}\phi'_{j-1}\,dx \text{ for } i,j = 1,2.$$

Note that we have identified the space of piecewise linear functions, v, with the vector space of values, (v_i), at the nodes.

The subspace, S, of piecewise linear functions that vanish at $x = 0$, defined in Sect. 0.4, can be identified with the subspace $\{(v_i) \ \ni \ v_0 = 0\}$.

Including v_0 in the data structure (with a value of zero) makes the assembly of bilinear forms equally easy in the presence of boundary conditions.

2 Two dimensional flow

The finite element method can be applied in the same way in any number of simulation dimensions.

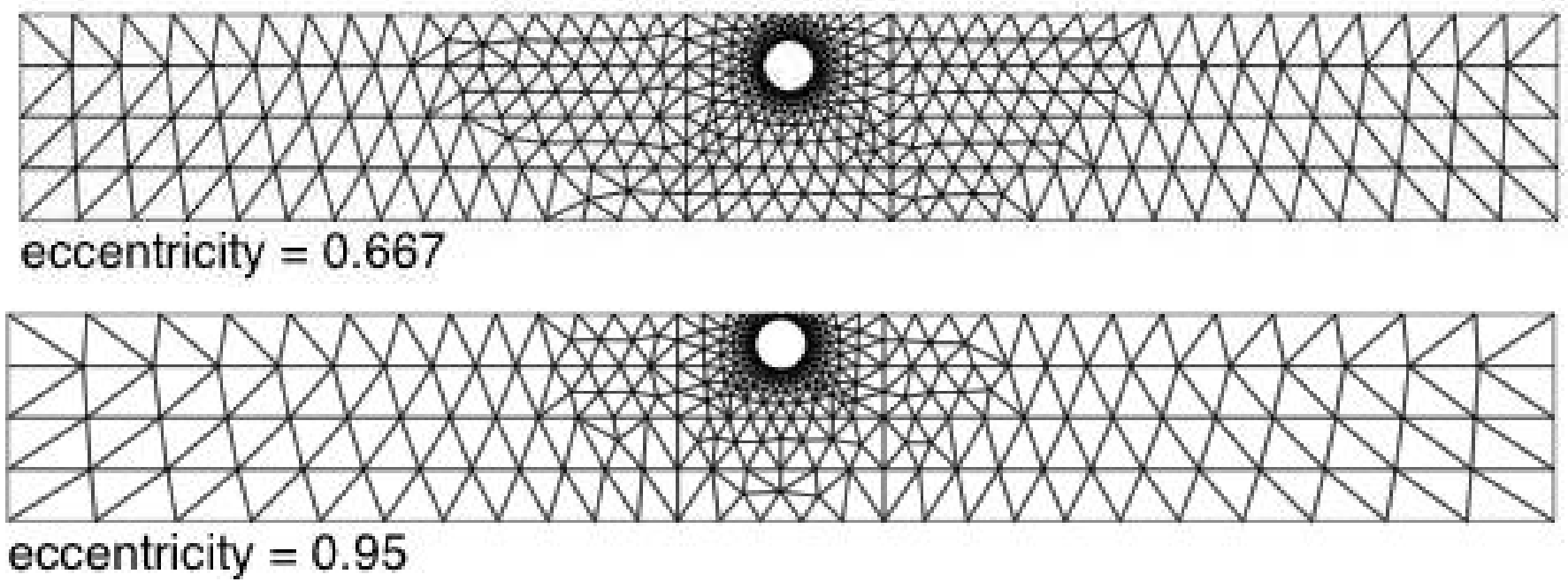

Figure 2: Mesh for pump flow in two dimensions.

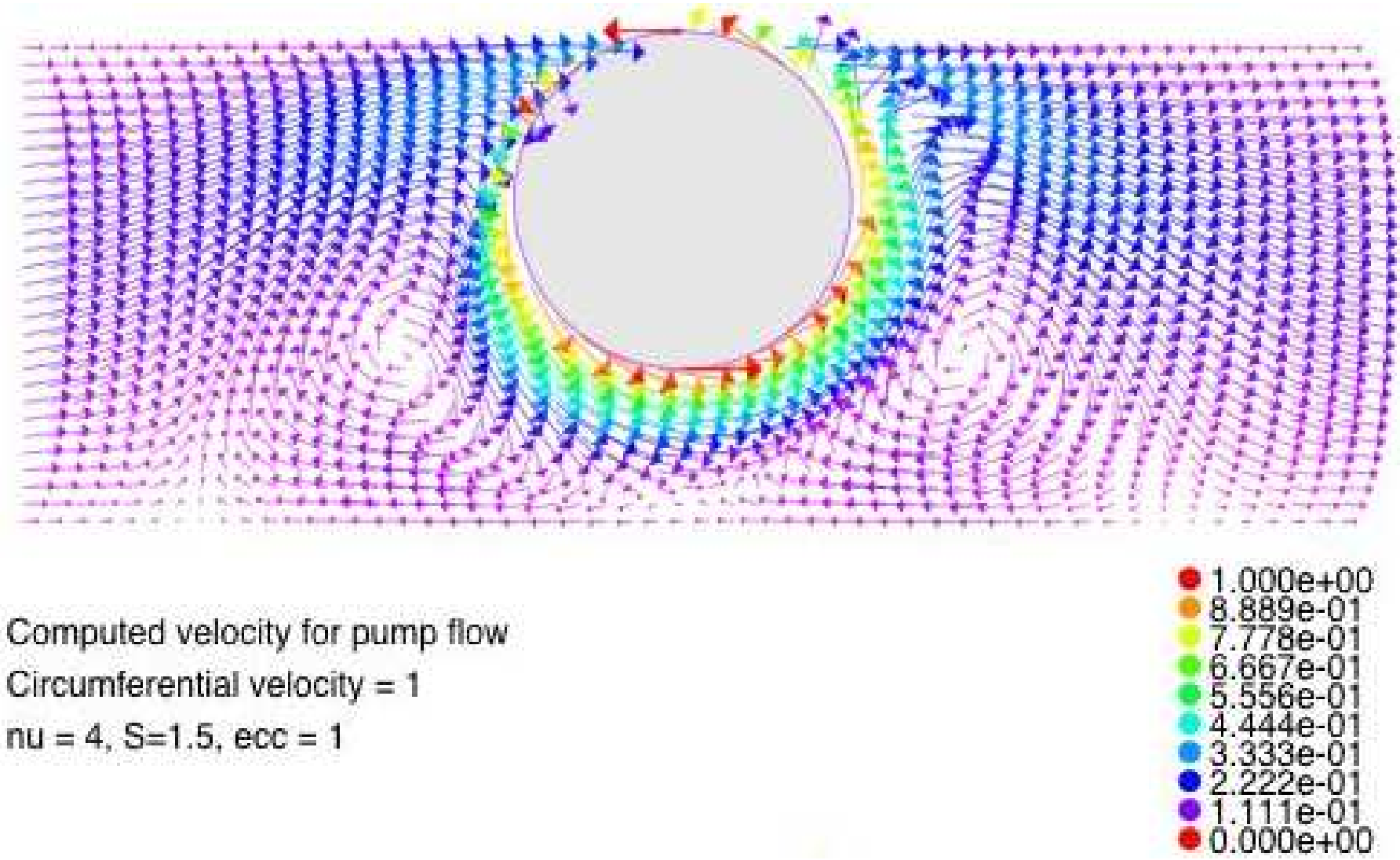

Figure 3: Pump flow in two dimensions

Model problem

We consider a variational problem with "energy" form

$$a(v,w) = \int_\Omega \alpha(x) \nabla v \cdot \nabla w \, dx \tag{2.7}$$

Let $V_h =$ piecewise linear functions on a non-degenerate mesh $\mathcal{T}_h$, and assume that the discontinuities of α and f, fall on mesh lines in $\mathcal{T}_h$.

Solve for $u_h \in V_h$ such that

$$a(u_h, v) = (f, v) \quad \forall v \in V_h \tag{2.8}$$

The application of a finite element method is similar to the one-dimensional case.

A mesh as in Figure 2 is created, and a corresponding space V_h of piecewise polynomials is defined.

In this way, simulations as depicted in Figure 3 can be performed.

2.1 Error estimators and adapted meshes

It is also possible to predict the error on a given mesh automatically.

Based on the error prediction a new mesh can be created.

The error $e_h := u - u_h$ satisfies the *residual equation*

$$a(e_h, v) = R(v) \quad \forall v \in V \tag{2.9}$$

where the *residual* $R \in V'$ is defined by $R(v) :=$

$$\sum_T \int_T (f - \nabla\alpha \cdot \nabla u_h) v \, dx + \sum_e \oint_e [\alpha \mathbf{n} \cdot \nabla u_h] v \, ds \tag{2.10}$$

One part of R is *absolutely continuous*

$$R_A|_T := (f - \nabla\alpha \cdot \nabla u_h)\,|_T = (f - \nabla \cdot (\alpha \nabla u_h))\,|_T$$

since ∇u_h and α are smooth on each T.

The other term in the definition of the residual is the "jump" term

$$R_J(v) := \sum_e \oint_e [\alpha \mathbf{n} \cdot \nabla u_h] v \, ds \quad \forall v \in V \tag{2.11}$$

where $[\phi]$ denotes the jump in ϕ (across the face in question). More precisely,

$$[\phi](x) := \lim_{\epsilon \to 0} \phi(x + \epsilon \mathbf{n}) - \phi(x - \epsilon \mathbf{n})$$

so that the expression in (2.8) is independent of the choice of normal $\mathbf{n}$ on each face.

If $\mathcal{A}$ is the differential operator associated with the form (2.7), namely, $\mathcal{A}v := -\nabla \cdot (\alpha \nabla v)$, then we see that $R_A = \mathcal{A}(u - u_h) = \mathcal{A}e_h$ on each T.

Relations (2.9–2.10) are derived simply by integrating by parts on each T, and the resulting boundary terms are collected in the term R_J.

Although (2.9–2.10) can be viewed as just a re-writing of (2.9), it gives an expression of the error in terms of a right-hand side $R \in V'$.

Inserting $v = e_h$ in (2.9), we see that

$$\alpha_0 |e_h|^2_{H^1(\Omega)} \leq |R(e_h)| \leq \|R\|_{H^{-1}(\Omega)} \|e_h\|_{H^1(\Omega)}. \tag{2.12}$$

Therefore

$$\alpha_0 \|e_h\|_{H^1(\Omega)} \leq \|R\|_{H^{-1}(\Omega)}. \tag{2.13}$$

Error estimated by $\|R\|_{H^{-1}(\Omega)}$ involves only data (f and α) and and something we have computed (u_h).

Difficult to compute a negative norm explicitly since R has two different parts: standard (integrable) function plus "interface Delta functions."

But can provide an effective estimate of $\|R\|_{H^{-1}(\Omega)}$ as follows.

The residual has special properties. In particular, the fundamental orthogonality implies that

$$R(v) := a(e_h, v) = 0 \quad \forall v \in V_h.$$

For each interior face e, let T_e denote the union of the two elements sharing that face. Then using a non-smooth data interpolant $\mathcal{I}_h$ [Scott-Zhang] we find

$$\begin{aligned} |R(v)| =& |R(v - \mathcal{I}_h v)| \\ \leq& \gamma \Big(\sum_T \| f - \nabla\alpha \cdot \nabla u_h \|^2_{L^2(T)} h_T^2 \\ &+ \sum_e \| \left[\alpha \mathbf{n} \cdot \nabla u_h \right] \|^2_{L^2(e)} h_e \Big)^{1/2} |v|_{H^1(\Omega)} \end{aligned} \tag{2.14}$$

Here h_e (resp. h_T) is a measure of the size of e (resp. T), and $\widehat{T}$ (resp. $\widehat{T}_e$) denotes the neighborhood of elements touching T (resp. T_e). For this reason, we define the *local error indicator* $\mathcal{E}_e$ by

$$\begin{aligned}\mathcal{E}_e(u_h)^2 := &\sum_{T\subset T_e} h_T^2 \|f - \nabla\alpha\cdot\nabla u_h\|^2_{L^2(T)} \\ &+ h_e \| \left[\alpha \mathbf{n}\cdot\nabla u_h\right] \|^2_{L^2(e)}\end{aligned} \tag{2.15}$$

where a natural choice for h_e (resp. h_T) is the length of e (resp. square root of the area of T) unless the elements are anisotropic.

The error estimator (2.15) can be generated automatically from the description (2.7).

With this definition, the previous inequalities can be summarized as

$$|R(v)| \leq \gamma \left(\sum_e \mathcal{E}_e(u_h)^2 \right)^{1/2} |v|_{H^1(\Omega)}$$

which in view of (2.12) implies that

$$|e_h|_{H^1(\Omega)} \leq \frac{\gamma}{\alpha_0} \left(\sum_e \mathcal{E}_e(u_h)^2 \right)^{1/2} \tag{2.16}$$

where γ is only related to interpolation error.

From the error estimate, a better mesh can be determined, and the process repeated to get a more accurate simulation.

The use of adaptivity in the mesh makes the simulation process much more efficient, although more complicated!

But it all can be done automatically.