1 Revue of the FEM

FEM provides formalism for generating discrete (finite) algorithms for approximating the solutions of differential equations.

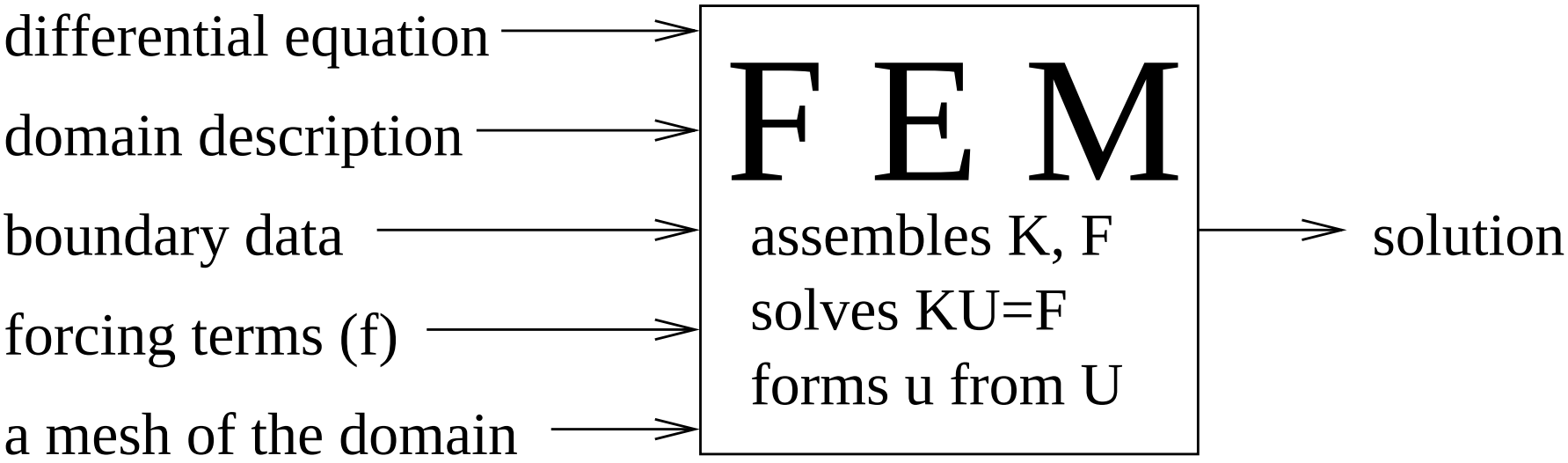

Figure 1: Black box into which one puts model problem and out of which pops an algorithm ($KU = F$) for approximating the corresponding solutions.

1.1 Weak Formulation of Boundary Value Problems

Consider the two-point boundary value problem

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

where $\Omega = (0, 1)$. Define a space incorporating the essential boundary condition:

$$V = \{v \in L^2(\Omega) : \quad a(v, v) < \infty \quad \text{and} \quad v(0) = 0\}.$$

(1) Multiply the differential equation in (1.1) by $v \in V$:

$$f(x)v(x) = -u''(x)v(x) \tag{1.2}$$

(2) Integrate (1.2) over the domain Ω:

$$\int_0^1 f(x)v(x)dx = \int_0^1 -u''(x)v(x)dx \tag{1.3}$$

Continuation of three-step recipe

(3) Integrate by parts (1.3) to get

$$\begin{aligned}(f,v) := \int_0^1 f(x)v(x)dx &= \int_0^1 -u''(x)v(x)dx \\ &= \int_0^1 u'(x)v'(x)dx =: a(u,v).\end{aligned} \tag{1.4}$$

Then we can say that the solution u to (1.1) is characterized by

$$u \in V \quad \text{such that} \quad a(u,v) = (f,v) \qquad \forall v \in V, \tag{1.5}$$

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

Theorem 1.1 *Suppose* $f \in C^0([0,1])$ *and* $u \in C^2([0,1])$ *satisfy* (1.5). *Then* u *solves* (1.1).

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.

But the variational problem can be solved with much more general f, including ones that are not functions, such as the Dirac δ:

$$(\delta, f) = f(a)$$

for some $a \in \Omega$.

For this reason, (1.5) 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 (1.5) with V replaced by S, namely

$$u_S \in S \quad \text{such that} \quad a(u_S, v) = (f, v) \qquad \forall v \in S. \tag{1.6}$$

Theorem 1.2 *Given $f \in L^2(0,1)$, (1.6) has a unique solution.*

Write (1.6) in terms of a basis of S: $\{\phi_i : \ 1 \leq i \leq n\}$, and expand

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

$$K_{ij} = a(\phi_j, \phi_i), F_i = (f, \phi_i) \quad \text{for} \quad i, j = 1, ..., n.$$

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

Then (1.6) is equivalent to solving the (square) matrix equation

$$\mathbf{KU} = \mathbf{F}. \tag{1.7}$$

For a square system such as (1.7) 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 $\mathbf{KV} = \mathbf{0}$.

For $v = \sum V_j \phi_j$, this means that $0 = a(v, v) = \int_0^1 (v')^2(x)\, dx$, from which we conclude that $\mathbf{V} = \mathbf{0}$.

Thus, the solution to (1.7) must be unique (and hence must exist).

Therefore, the solution u_S to (1.6) 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

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

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 generated by the basis functions shown in Fig. 2.

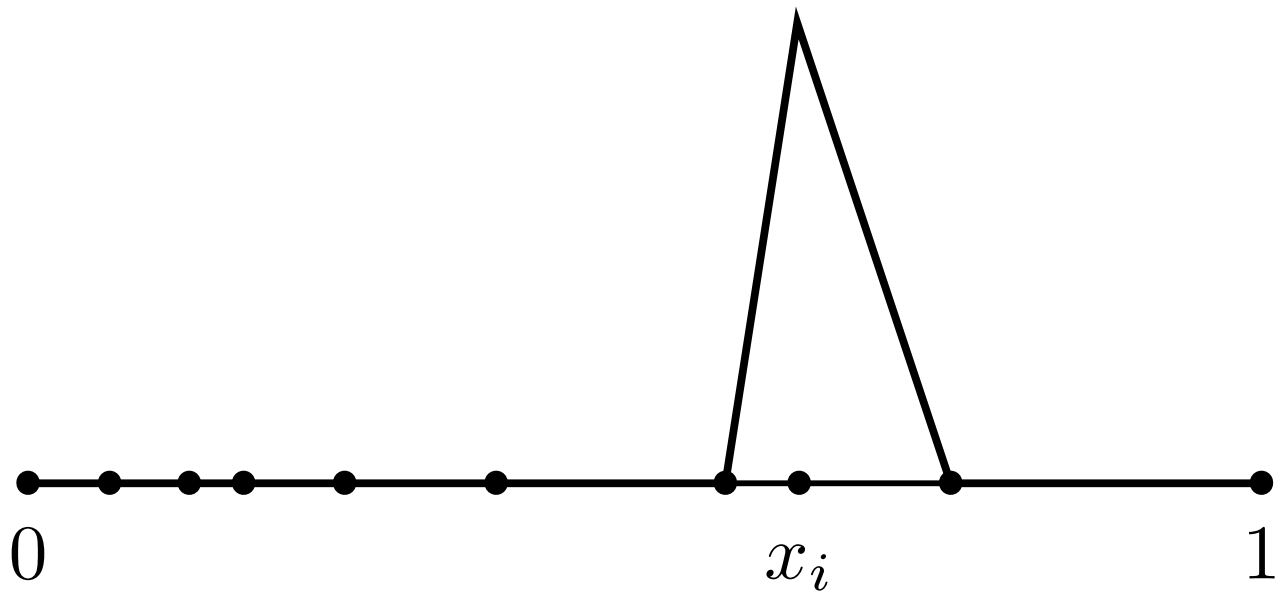

Figure 2: piecewise linear basis function ϕ_i

$\{\phi_i\}$ 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**.)

1.5 Computer Implementation of Finite Element Methods

The key 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 f or of a vector F as

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

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 bilinear forms defined in (1.4) can be assembled using this representation:

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

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

$$\begin{aligned} a_e(v,w) \;&:=\; \int_{I_e} v'w'\,dx \\ &=\; (x_e - x_{e-1})^{-1} \begin{pmatrix} v_{i(e,0)} \\ v_{i(e,1)} \end{pmatrix}^t \mathbf{K} \begin{pmatrix} w_{i(e,0)} \\ w_{i(e,1)} \end{pmatrix}. \end{aligned} \tag{1.8}$$

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

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

1.6 More weak formulations

Consider the two-point boundary value problem

$$\begin{aligned} -\frac{d^2u}{dx^2} + \lambda u &= f \text{ in } \Omega \\ u(0) = 0, \qquad & u'(1) = 0. \end{aligned} \tag{1.9}$$

where $\Omega = (0, 1)$. Define

$$V = \{v \in L^2(\Omega): \quad a(v, v) < \infty \quad \text{and} \quad v(0) = 0\}.$$

(1) Multiply the differential equation in (1.9) by $v \in V$,

(2) Integrate over the domain Ω, and (3) Integrate by parts to get

$$(f, v) = \int_0^1 u'(x)v'(x) + \lambda v(x)u(x)dx =: a(u, v). \tag{1.10}$$

This leads to $KU + \lambda MU = F$, where M is called the mass matrix: $M_{ij} = (\phi_i, \phi_j)$. This equation will be singular for some $\lambda < 0$ since it corresponds to an eigenvalue problem: $M^{-1}KU = -\lambda U$.

1.7 Different boundary conditions

Consider the boundary conditions

$$\begin{aligned} u(0) &= 0, & u(1) &= 0 \\ u(0) &= 0, & u'(1) &= 0 \\ u'(0) &= 0, & u(1) &= 0 \\ u'(0) &= 0, & u'(1) &= 0 \end{aligned} \tag{1.11}$$

These correspond to the spaces

$$\begin{aligned} V &= \{v \in W : v(0) = 0 \quad \text{and} \quad v(1) = 0\} \\ V &= \{v \in W : v(0) = 0\} \\ V &= \{v \in W : v(1) = 0\} \\ V &= W \end{aligned} \tag{1.12}$$

respectively, where W is the space

$$W = \{v \in L^2(\Omega) : \quad a(v, v) < \infty\}$$

1.8 Multiple weak formulations

Consider the two-point boundary value problem

$$-\frac{d^2u}{dx^2} + \frac{du}{dx} = f \text{ in } \Omega \tag{1.13}$$

where $\Omega = (0, 1)$. This leads to multiple variational formulations:

$$\begin{aligned} \int_0^1 u'(x)v'(x) + v(x)u'(x)dx &=: a_1(u, v). \\ \int_0^1 u'(x)v'(x) - v'(x)u(x)dx &=: a_2(u, v). \end{aligned} \tag{1.14}$$

depending on whether or not we integrate by parts on the second term on the left hand side in (1.13)

For this reason, using variational forms as a language avoids some ambiguity in the model definition.

2 Matrix Evaluation by Assembly

The *assembly* of integrated differential forms is done by summing its constituent parts over each *element*, which are computed separately through the use of a numbering scheme called the *local-to-global* index. This index, $\iota(e, \lambda)$, relates the local (or element) node number, $\lambda \in \mathcal{L}$, on a particular element, indexed by e, to its position in the global data structure.

We may write a finite element function f in the form

$$\sum_{e} \sum_{\lambda \in \mathcal{L}} f_{\iota(e,\lambda)} \phi^{e}_{\lambda} \tag{2.15}$$

where f_i denotes the "nodal value" of the finite element function at the i-th node in the global numbering scheme and $\{\phi^{e}_{\lambda} \ \ni \ \lambda \in \mathcal{L}\}$ denotes the set of basis functions on the element domain T_e.

The element basis functions, ϕ^e_λ, are extended by zero outside T_e.

Can relate "element" basis functions ϕ^e_λ to fixed set of basis functions on "reference" element, $\mathcal{T}$, via mapping of $\mathcal{T}$ to T_e.

Could involve changing both the "x" values and the "ϕ" values in a coordinated way, as with the Piola transform , or it could be one whose Jacobian is non-constant, as with tensor-product elements or isoparametric elements.

For an affine mapping, $\xi \to J\xi + x_e$, of $\mathcal{T}$ to T_e:

$$\phi^e_\lambda(x) = \phi_\lambda \left(J^{-1}(x - x_e) \right) .$$

The inverse mapping, $x \to \xi = J^{-1}(x - x_e)$ has as its Jacobian

$$J^{-1}_{mj} = \frac{\partial \xi_m}{\partial x_j} ,$$

and this is the quantity which appears in the evaluation of the bilinear forms. Of course, $\det J = 1/\det J^{-1}$.

2.1 Evaluation of bilinear forms

The assembly algorithm utiizes the decomposition of a variational form as a sum over "element" forms

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

where "element" bilinear form for Laplace's equation defined via

$$\begin{aligned} a_e(v,w) &:= \int_{T_e} \nabla v(x) \cdot \nabla w(x)\, dx \\ &= \int_{\mathcal{T}} \sum_{j=1}^{d} \frac{\partial}{\partial x_j} v(J\xi + x_e) \frac{\partial}{\partial x_j} w(J\xi + x_e) \det(J)\, d\xi \end{aligned} \tag{2.16}$$

by transofrming to the reference element.

Finite element matrices computed via assembly in a similar way.

The local element form is computed as follows.

2.2 Evaluation of bilinear forms—continued

$$
\begin{aligned}
a_e(v,w) &= \int_{\mathcal{T}} \sum_{j=1}^{d} \frac{\partial}{\partial x_j} v(J\xi + x_e) \frac{\partial}{\partial x_j} w(J\xi + x_e) \det(J)\, d\xi \\
&= \int_{\mathcal{T}} \sum_{j,m,m'=1}^{d} \frac{\partial \xi_m}{\partial x_j} \frac{\partial}{\partial \xi_m} \left(\sum_{\lambda \in \mathcal{L}} v_{\iota(e,\lambda)} \phi_\lambda(\xi) \right) \times \\
&\qquad\qquad \frac{\partial \xi_{m'}}{\partial x_j} \frac{\partial}{\partial \xi_{m'}} \left(\sum_{\mu \in \mathcal{L}} w_{\iota(e,\mu)} \phi_\mu(\xi) \right) \det(J)\, d\xi \qquad (2.17) \\
&= \begin{pmatrix} v_{\iota(e,1)} \\ \cdot \\ \cdot \\ v_{\iota(e,|\mathcal{L}|)} \end{pmatrix}^t \mathbf{K}^e \begin{pmatrix} w_{\iota(e,1)} \\ \cdot \\ \cdot \\ w_{\iota(e,|\mathcal{L}|)} \end{pmatrix}.
\end{aligned}
$$

Here, the *element stiffness matrix*, $\mathbf{K}^e$, is given by

$$K^e_{\lambda,\mu} := \sum_{j,m,m'=1}^{d} \frac{\partial \xi_m}{\partial x_j} \frac{\partial \xi_{m'}}{\partial x_j} \det(J) \int_{\mathcal{T}} \frac{\partial}{\partial \xi_m} \phi_\lambda(\xi) \frac{\partial}{\partial \xi_{m'}} \phi_\mu(\xi)\, d\xi \tag{2.18}$$

$$= \sum_{m,m'=1}^{d} G^e_{m,m'} K_{\lambda,\mu,m,m'}$$

where

$$K_{\lambda,\mu,m,m'} = \int_{\mathcal{T}} \frac{\partial}{\partial \xi_m} \phi_\lambda(\xi) \frac{\partial}{\partial \xi_{m'}} \phi_\mu(\xi)\, d\xi \tag{2.19}$$

and

$$G^e_{m,m'} := \det(J) \sum_{j=1}^{d} \frac{\partial \xi_m}{\partial x_j} \frac{\partial \xi_{m'}}{\partial x_j} \tag{2.20}$$

for $\lambda, \mu \in \mathcal{L}$ and $m, m' = 1, \ldots, d$.

3 Revue of Adaptivity

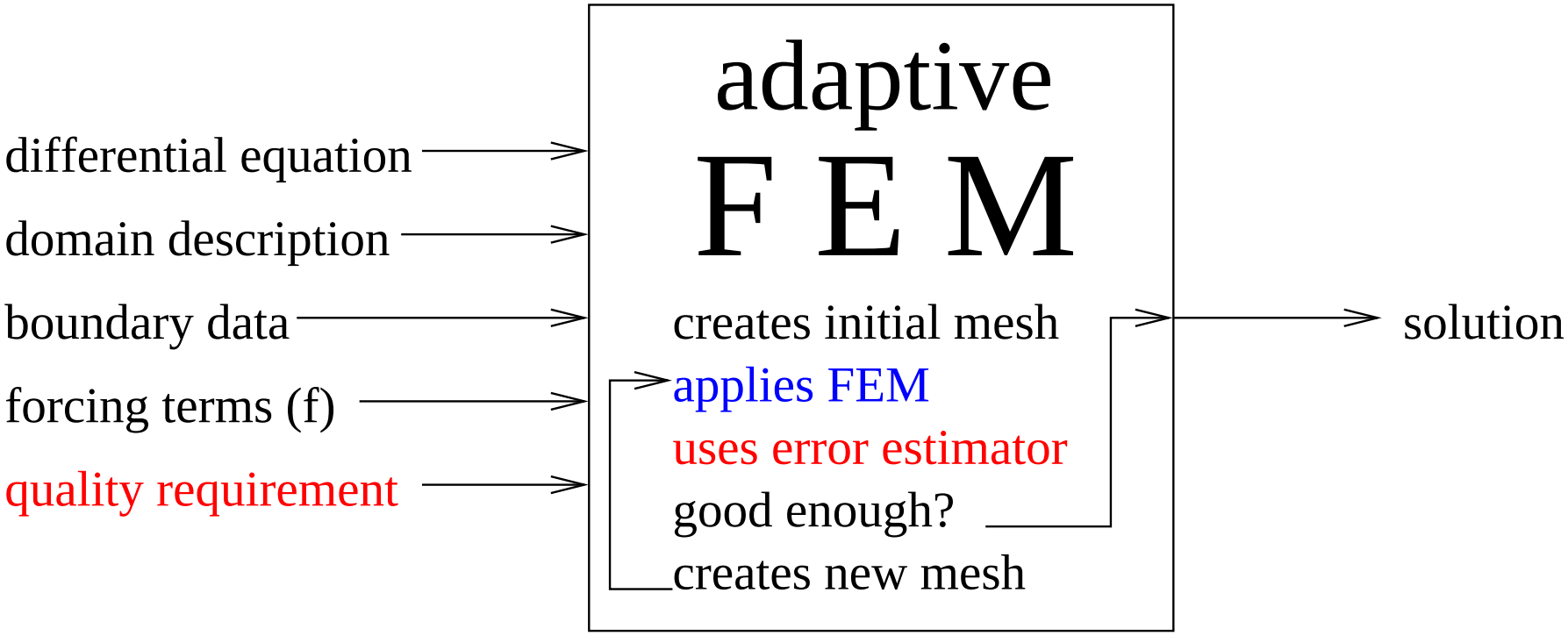

Figure 3: Black box for adaptive FEM; requires no mesh initially, only quality requirement. Generates a sequence of meshes and applies standard FEM until quality is assured.

Model problem

We consider a variational problem with "energy" form

$$a(v,w) = \int_\Omega \alpha(x) \nabla v \cdot \nabla w \, dx \tag{3.21}$$

Solve for $u_h \in V_h$ such that

$$a(u_h, v) = (f, v) \quad \forall v \in V_h \tag{3.22}$$

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

$$a(e_h, v) = R(v) \quad \forall v \in V \tag{3.23}$$

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

$$\sum_T \int_T (f - \nabla\alpha \cdot \nabla u_h) v \, dx + \sum_e \oint_e [\alpha \mathbf{n} \cdot \nabla u_h] v \, ds \tag{3.24}$$

If $\mathcal{A}$ is the differential operator associated with the form (3.21), 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 (3.23–3.24) can be derived automatically.

The *local error indicator* $\mathcal{E}_e$ by

$$\begin{aligned}\mathcal{E}_e(u_h)^2 := &\sum_{T\subset T_e} h_T^2 \|f - \nabla\alpha\cdot\nabla u_h\|^2_{L^2(T)} \\ &+h_e\| \left[\alpha\mathbf{n}\cdot\nabla u_h\right] \|^2_{L^2(e)}\end{aligned} \tag{3.25}$$

can also be generated automatically from the description (3.21).

With this definition, we showed that

$$|e_h|_{H^1(\Omega)} \leq \frac{\gamma}{\alpha_0}\left(\sum_e \mathcal{E}_e(u_h)^2\right)^{1/2} \tag{3.26}$$

where γ is only related to interpolation error.

From the error estimate, a better mesh can be determined: we refine where $\mathcal{E}_e(u_h)$ is large.

The process is repeated to get a more accurate simulation.

The use of adaptivity is more complicated but makes the simulation process much more efficient.

But it all can be done automatically.