Introduction to Finite Element Methods*

Paul S. Heckbert
Carnegie Mellon University

Global Illumination Course
SIGGRAPH '93

Most simulation problems cannot be solved symbolically, and must instead be solved numerically. In this chapter we explain some of the basics of the finite element method, a numerical technique for breaking up the domain of a problem into smaller pieces, thereby transforming one big problem into a number of simpler problems.

1 Motivation

In a differential equations course, one typically studies a few classes of problems for which there are closed form solutions, such as ordinary, linear differential equations with constant coefficients. Most problems of interest in real world simulation problems are much more complex, however, involving domains of two or more dimensions or nonlinear effects, yielding partial differential equations or nonlinear differential equations, respectively. Many phenomena in nature, such as heat conduction or convection, stress in mechanical structures, electromagnetic fields, and fluid mechanics are described by partial differential equations involving first and second spatial derivatives and time derivatives of multidimensional functions. Examples of such equations are Maxwell's equations, the heat equation, and the Schrödinger wave equation.

Other physical phenomena, such as thermal radiation, are described by integral equations or integro-differential equations. Thermal radiation is the study of heat transfer by radiation. The character of the equations governing thermal radiation depends on whether the medium (volume) through which the radiation passes is a *participating medium*, which emits, scatters, or absorbs radiation, or a non-participating medium. An example of a participating medium is fog, and an example of a non-participating medium is a vacuum. Unlike conduction and convection, in which all flow of heat is determined by local phenomena (such as conductivity, pressure, or temperature differences in an infinitesimal neighborhood), thermal radiation is determined by non-local phenomena. The radiation incident on a surface is a function of the integral of the radiation on other surfaces. For these reasons, thermal radiation in a non-participating medium is governed by an integral equation, and thermal radiation in a participating medium is governed by a integro-differential equation.

Neither differential equations, integral equations, nor integro-differential equations can be solved symbolically or analytically, in general. Nevertheless, we want to simulate these

*Appeared in Global Illumination course notes, SIGGRAPH '93, Anaheim, Aug. 1993 phenomena in order to design faster airplanes, lighter bridges, smaller circuits, more livable buildings, and a host of other science and engineering problems.

To do so we use numerical methods. Suppose we have a simulation problem that can be expressed formally as $\mathcal{F}u = g$, where $\mathcal{F}$ is a differential, integral, or integro-differential operator which when applied to a function $u(x)$ yields a function $g(x)$. $\mathcal{F}$ and g are known, but u is unknown. The problem, put abstractly, is that we can't just solve $u = \mathcal{F}^{-1}g$ because $\mathcal{F}$ is uninvertible. To find the real solution u could entail a search through an infinite-dimensional space of all functions over the domain. Instead we will simplify the problem to a search in a finite-dimensional function space.

Most numerical methods involve an approximation to the unknown function u by a new function $\hat{u}$ which is a linear combination of basis functions

$$\hat{u}(x) = \sum_{i=1}^{n} u_i W_i(x)$$

where n is the number of basis functions, u_i are the unknown coefficients, W_i are the chosen basis functions, and x is the domain variable, possibly multidimensional. Basis functions are also referred to as "shape functions" in the finite element literature. The coefficients u_i are coordinates in this n-dimensional function space, and each function $\hat{u}$ corresponds to a point in this space.

Classical methods of mathematical analysis allow problems with simple geometry to be solved, but they are impractical for most real world problems. Take, for example, the problem of stress analysis of a solid with applied forces. Classical techniques employ global basis functions $W_i(x)$ (basis functions that are nonzero over the entire domain) such as polynomials or sinusoids of various frequencies. Approximations of this form will work adequately for simple geometries such as a cube, but will yield poor approximations for concave shapes such as an "L". They work poorly on complex geometries because simplistic, global basis functions cannot reflect properties of the solution function that are induced by the geometry – in this case, the concentration of stress at reflex corners.

2 Finite Elements

Instead of employing global basis functions, the finite element methods breaks the domain into a finite number of pieces called *elements*, and uses basis functions that are local to each element [Becker et al. 81,Strang-Fix73]. Common choices for basis functions are piecewise polynomials.

2.1 One-Dimensional Elements

For a one-dimensional domain $[\alpha, \beta]$, polynomial elements are defined in terms of a sequence of element endpoints x_i, where $\alpha = x_0 \leq x_1 \leq \ldots \leq x_n = \beta$. This sequence of elements is a one-dimensional *mesh*. These points are analogous to knot vectors for splines [Bartels et al. 87].

Constant elements employ a *box* basis function that is 1 inside the interval and 0 outside. Linear elements employ two *half hat* basis functions, one that rises from 0 to 1 across the interval, and one which falls from 1 to 0 (see figure 1):

$$\text{box:} \qquad W_i^0(x) = \begin{cases} 1 & \text{if } x_i \leq x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

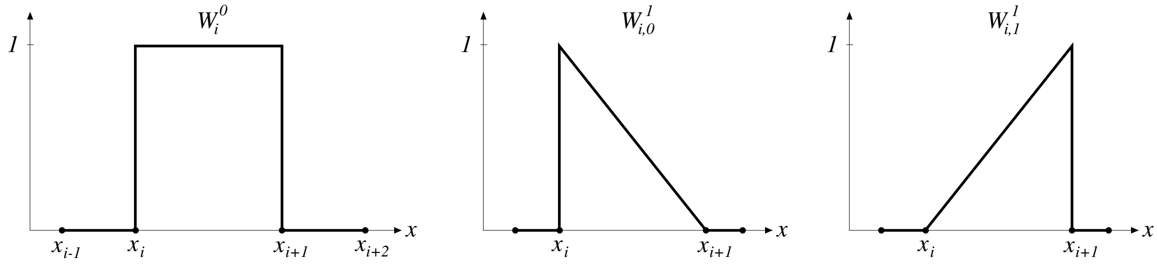

Figure 1: *Box basis function, and left and right half hat basis functions.*

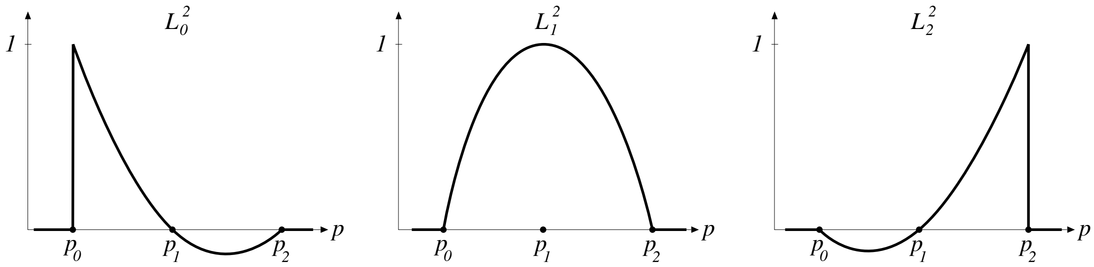

Figure 2: *The three basis functions for quadratic Lagrange interpolation.*

$$\text{left half-hat:} \qquad W_{i,0}^1(x) = \begin{cases} \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{if } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{right half-hat:} \qquad W_{i,1}^1(x) = \begin{cases} \frac{x-x_i}{x_{i+1}-x_i} & \text{if } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

(superscripts here denote degree, not exponentiation). In one dimension, a constant element has one degree of freedom which is the value of the approximation $\hat{u}(x)$ in the interior of the interval. Within an interval $[x_i, x_{i+1}]$, a linear element uses an approximation of the form $\hat{u}(x) = u_i W_{i,0}^1(x) + u_{i+1} W_{i,1}^1(x)$. The two coefficients u_i and u_{i+1}, which are the values of $\hat{u}$ at x_i and x_{i+1}, are the two degrees of freedom. When the degrees of freedom are associated with the value of the function at a particular point, we speak of these points as *nodes*.

Quadratic and higher degree elements are common as well. The choice of basis functions is somewhat arbitrary. Sometimes a set of basis functions is chosen because it has an intuitive parameterization; other times, because it yields lower numerical error. The most obvious set of basis functions is probably the *power basis*: $1,\ x,\ x^2,\ \ldots,\ x^d$. This basis function has poor numerics and the coefficients are non-intuitive. Other possibilities are the B-spline basis functions and Bernstein polynomials common to the spline literature [Farin88]. All of these span the same function space.

The most commonly used basis functions in the finite element literature are not any of the above, but are defined in terms of Lagrange interpolation. The degrees of freedom are chosen to be the value of the function at selected points. Formulas for these are most easily given in terms of a *master element* with domain $p \in [0, 1]$. This master element can be scaled and translated to map it to any interval using the transformation $x = (x_{i+1} - x_i)p + x_i$. To create a degree d element, $d - 1$ points are chosen in the master interval, such that

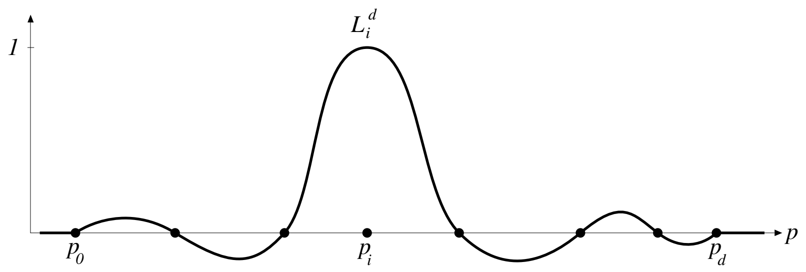

Figure 3: *Lagrange basis function for node i, degree d.*

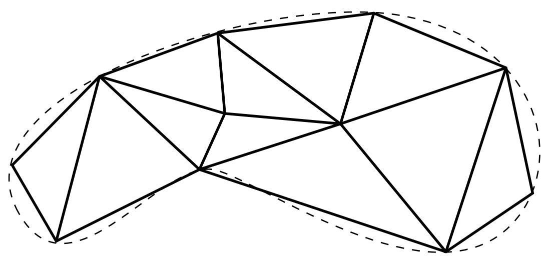

Figure 4: *Triangulation of a 2-D domain.*

$0 = p_0 < p_1 < \ldots < p_d = 1$. The ith Lagrange polynomial of degree d, for $i = 0, 1, ..., d$ is

$$L_i^d(p) = \prod_{\substack{j=0\ldots d \\ j \neq i}} \frac{p - p_j}{p_i - p_j}$$

Quadratic Lagrange polynomials are shown in figure 2, and a degree d Lagrange is shown in figure 3. This basis function has value 1 at $p = p_i$ and value 0 at all other nodes p_j. The box and half hat basis functions given earlier are just the degree 0 and degree 1 Lagrange basis functions. The interior nodes are typically taken to be equispaced. There are $d+1$ Lagrange polynomials of degree d, and the degrees of freedom are the values of the approximation $\hat{u}$ at the endpoints of the interval and at the $d-1$ interior nodes.

2.2 Two-Dimensional Elements

In two or more dimensions, the nodes and edges of the elements define the vertices and edges of a *mesh*. The most general meshing technique in two dimensions is *triangulation*: subdivision of the domain into a network of non-overlapping triangles (figure 4). After the mesh is chosen, a set of basis functions for the mesh is selected. The mesh and the basis functions together determine the function space, and hence constrain the level of approximation that is attainable.

The simplest basis functions for triangular elements are low-order polynomials. And the simplest of these is the constant element: a function which is 1 everywhere inside the triangle and is 0 outside the triangle. As in 1-D, constant elements have a single degree of freedom which can be associated with the value at a point inside the triangle.

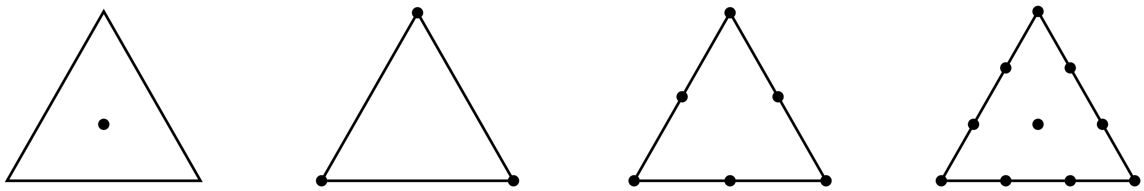

Figure 5: *Grids of nodes for Lagrange triangles of degree 0, 1, 2, and 3. Dots represent nodes.*

A linear element provides three degrees of freedom, since the resulting approximation has the form:

$$\hat{u}(x, y) = ax + by + c$$

Rather than use a, b, and c as the parameters of this function space (with corresponding "power basis" functions x, y, and 1), however, we use three linear basis functions each of which has the value 1 at one vertex of the triangle and the value 0 at the other two vertices.

The Lagrange polynomials described earlier can be generalized to 2-D by considering a canonical a master element with *barycentric coordinates* (p, q, r) [Zienkiewicz-Taylor89]. If we assume that the three vertices of the triangle have barycentric coordinates $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, and that they are an affine transformation of (x, y) (that is, each of p, q, and r is of the form $ax + by + c$), then $p + q + r = 1$ always. Intuitively, the barycentric coordinates measure normalized distance from each of the sides of the triangle.

The nodes are chosen to have coordinates in a triangular grid. This grid is constructed by subdividing each of the axes p, q, and r into d intervals (typically equispaced), and placing the nodes at the intersection points $\mathbf{p}_{ijk} = (p_i, q_j, r_k)$ where $i + j + k = d$, and $i, j, k \geq 0$. There are $\binom{d+2}{2}$ such nodes. Thus, a constant element of this type has 1 node, a linear element has 3 nodes, a quadratic element has 6 nodes, and a cubic element has 10 nodes (see figure 5). The degree d triangular Lagrange polynomial given by indices i, j, and k is then:

$$L^d_{ijk}(p, q, r) = L^i_i(p) L^j_j(q) L^k_k(r)$$

Note that only the first i nodes $p_0 \ldots p_{i-1}$ along the p axis are used in the first factor $L^i_i(p)$, and likewise for q and r. This product of 1-D basis functions yields a 2-D basis function that has the value 1 at node ijk, and the value 0 at all other nodes. Hence, as in one dimension, the degrees of freedom for this type of element are the values of the approximation function at the nodes.

Formulas for other polynomial basis functions for triangles can be found in any finite element textbook [Becker et al. 81], or in geometric modeling textbooks under "Bezier triangles" [Farin88].

2.3 Continuity Constraints

If one desires continuous approximation functions then neighboring nodes from abutting elements must be constrained. These conditions are equivalent to "parametric patch continuity constraints" discussed in the spline literature. In most applications, C^0 approximation functions are desired, in which case, coincident nodes at vertices and along edges are merged.

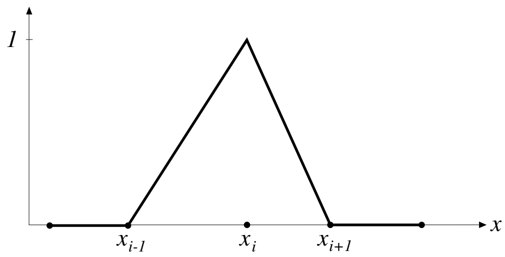

Figure 6: *Hat basis function.*

In some cases, higher or lower degrees (e.g. C^1 or C^{-1}) of continuity are desired, in which case basis functions can be customized.

Once the mesh and basis functions are chosen, the degrees of freedom of the function space are determined by the values of the approximation at each of the nodes. When the continuous problem to be solved (a differential equation or integral equation, say) is discretized, a system of equations results, with one unknown for each of these nodes.

The number of degrees of freedom in a discretization is $O(d/h)$ in 1-D and $O(d^2/h^2)$ in 2-D. The exact number of degrees of freedom depends on the precise choice of basis functions used, the topology of the mesh, and the continuity constraints.

Continuity constraints can be regarded as either additional equations constraining the existing degrees of freedom or a reduction in the number of degrees of freedom and a change in the corresponding basis functions. In one dimension, C^0 continuity merges nodes at interval endpoints, fusing the left and right half hat basis functions together, yielding hat basis functions (figure 6).

If half hat basis functions are used, then there will be no continuity constraints, each element will have two degrees of freedom, and an n-element domain will have $2n$ degrees of freedom. When hat basis functions are used, C^0 continuity is guaranteed, there will be one degree of freedom for each element endpoint, so an n-element domain will have $n+1$ degrees of freedom. Thus, the number of degrees of freedom for 1-D problems with C^0 linear elements is just one greater than the number for constant elements.

In two dimensions, the analogous procedure at a vertex of degree m (one having m adjacent triangles) merges d vertex nodes into 1, merges pairs of edge nodes along each of the m edges radiating from the vertex, and fuses the basis functions corresponding to each of these nodes. For example, the m triangular basis functions for the center vertex are fused into a new m-sided pyramid basis function. This new basis function will have value 0 all around the perimeter, so it will be C^0. We call such a basis function a 2-D hat.

3 Accuracy and Error Measures

Armed with all of the options described above, how would one choose a mesh and basis function to give a fast, accurate solution to a given simulation problem? The answer to this question depends on the particular problem, since the solution functions vary in character. In fluid flow, for example, velocity functions will vary in character between regions of laminar flow and regions of turbulent flow. It can thus be necessary to use meshing criteria that vary with space and time.

The choice of mesh and basis functions determines the function space, and hence determines what functions can be well approximated. Constant elements are perfect for representing functions that are piecewise constant with step discontinuities at the boundaries between elements, but few functions in the real world have such character. The hat basis functions described above are continuous, so any linear combination of them will be, as well. They are better suited to the approximation of smoothly varying functions than box basis functions, but note that they are poor at approximating a function containing a step discontinuity. A poor choice of mesh can lead to an inaccurate simulation because it fails to approximate features of the true solution function that are important to the application.

3.1 Error Metrics

The concept of accuracy for a particular problem can be quantified using *error metrics.* This will allow us to compare the accuracy of various solution methods more objectively.

A function norm gives a scalar measure of the "size" of a function over a domain Γ. A common class is the L_p norm:

$$||f||_p = \left(\int_\Gamma dx\, |f(x)|^p\right)^{1/p}$$

the most common of which are the L_1 norm, the L_2 norm, and the L_∞ norm. The L_2 norm is also known as "root mean square (RMS) error".

The relative "distance" in function space between the real solution function $u(x)$ and an approximation $\hat{u}(x)$ is a good error metric:

$$\text{Error}_2(\hat{u}) = \frac{||\hat{u} - u||_2}{||u||_2}$$

Although it is typically impossible to compute the above error measure because u is unknown, approximations can be computed, and the above is useful as a theoretical tool, nevertheless.

4 Mesh Generation and Refinement

Many mesh generation techniques have been used over the years, ranging from simple to complex. An excellent survey of mesh generation and triangulation algorithms is [Bern-Eppstein92]. Meshes with the topology of a rectangular or triangular grid are particularly easy to implement, but yield poor results for some problems. Better results can often be achieved using *adaptive meshes* whose element size, orientation, shape, and basis functions are adaptive to the problem at hand.

Many of the early mesh generation techniques required manual labor, or they were very application-specific, but general techniques and rules are emerging. In general, the error of an approximation is limited by the following factors:

- the size of the largest element
- the largest angle in the triangulation (hence sliver-like triangles are to be avoided)
- the accuracy of the approximation to the geometry (especially important when approximating curved geometry by linear elements)

- the ability of the function space to represent singularities and discontinuities in the solution function

.

For a simulation problem without singularities, the asymptotic error for a mesh with largest element size h and smallest basis function degree d is

$$\text{Error}_2 = O(h^{d+1})$$

The above assumes that the remaining steps in the solution process, which typically involve numerical methods for integration and solving systems of equations, are carried out exactly. Note that the constants and exponent in the asymptotic error is dependent on the particular error measure used. The above equation says that the error can be driven to zero by reducing the element size h or by increasing the element degree d. Reducing the element size is called h-refinement and increasing the degree (or power) is called p-refinement.

When a problem contains singularities that are not resolved by the basis functions, however, the error converges more slowly (by a smaller power of h) [Zienkiewicz-Taylor89].

In many cases it is impossible to predict theoretically the error of a given algorithm, so errors are measured empirically instead, by performing experiments, using the best known solution as the "exact" solution u, and plotting error as a function of h or d. If log(Error) is plotted against $\log h$, the points typically cluster close to a line, as one would predict from the formula above. The slope of the line gives the *convergence rate* of the method.

When it is not possible to predict beforehand whether a given mesh and set of basis functions will yield the desired accuracy, an a posteriori adaptive technique is used:

1. An initial mesh and basis functions are chosen.
2. The problem is discretized and solved, and error local estimates are made.
3. If error is small enough, stop.
4. Perform h-refinement and/or p-refinement. Go to step 2.

In some cases, the most cost-effective solution (the one yielding best accuracy for a given amount of computer time) involves a mixture of h-refinement and p-refinement. Practitioners have found that small, low-degree elements are best near singularities, but larger, higher-degree elements are best in smooth regions of the domain. Very high degree elements lead to numerical difficulties, however, so linear, quadratic, and cubic elements are probably the most popular.

5 Constraining and Solving

Once the mesh and basis functions are chosen, a continuous problem can be constrained and solved. As described in an earlier chapter, common methods for constraining the approximation functions to lie in the chosen function space are collocation methods and the Galerkin method. Both methods transform a continuous problem of the form $\mathcal{F}u = g$ into a system of equations of the form $\mathcal{F}_i(u_1, \ldots, u_n) = g_i$, for $i = 1, \ldots, n$.

If the initial equations (differential or integral) are linear, then the resulting system of equations will be linear – a very important special case. In this case the discretized problem can be written in the form $\mathbf{Fu} = \mathbf{g}$, where $\mathbf{F}$ is a square matrix and $\mathbf{g}$ and $\mathbf{u}$ are vectors. There are many methods for solving linear systems of equations [Golub-Van Loan89]. For a general $n \times n$ system, Gaussian elimination takes time $O(n^3)$, but for some matrices, iterative methods such as Gauss-Seidel, Successive Overrelaxation, or the Conjugate Gradient Method yield adequate accuracy in time closer to $O(n^2)$. The exact convergence rate of these problems is dependent on the eigenvalues of the matrix $\mathbf{F}$, which in turn are dependent on the physics of the problem one is solving and the choice of mesh and basis functions. Extremely thin triangles (having very small or very large angles) can increase the condition number of the matrix, increasing the error.

Although detailed properties of the matrix $\mathbf{F}$ are application-specific, a few generalities can be made. Most partial differential equations involve only local phenomena: the value of a few functions and their derivatives at a single point, with no influence from distant points. Consequently, the matrix $\mathbf{F}$ will be sparse, and the only nonzero entries in row i of matrix $\mathbf{F}$ will be the degrees of freedom associated with node i or the set of neighbors that are used to estimate the derivatives at i. For a mesh of limited degree vertices (no vertices with more than $m_{\max}$ emanating edges), the number of neighbors of a node will be bounded, independent of the mesh size h or the number of elements. The fraction of nonzero entries in the matrix will then be $O(1/n)$. When the mesh has regular topology, such as a triangulated rectangular grid, the matrix $\mathbf{F}$ is often banded, allowing special optimized solvers to be used. Even when the topology is unstructured and the matrix is not banded, more general sparse system solvers can be used that save both memory and time relative to dense system solvers. Integral equations, on the other hand, have dense matrices (few nonzeros), in general.

One of the fastest known solution methods is the multigrid method, which can yield solutions in time $O(n)$ for some regular problems. Many of the most dramatic speedups that have been achieved on specific finite element problems in recent years have exploited physical properties of specific simulation problems.

References

[Bartels et al. 87] Richard Bartels, John Beatty, and Brian Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann Publishers, San Mateo, CA, 1987.

[Becker et al. 81] Eric B. Becker, Graham F. Cary, and J. Tinsley Oden. *Finite Elements: An Introduction*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[Bern-Eppstein92] Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. Technical report, Xerox PARC, March 1992. CSL-92-1. Also appeared in "Computing in Euclidean Geometry", F. K. Hwang and D.-Z. Du, eds., World Scientific, 1992.

[Farin88] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, Boston, 1988.

[Golub-Van Loan89] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 1989.

[Strang-Fix73] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[Zienkiewicz-Taylor89] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, Vol. 1.* McGraw-Hill, London, fourth edition, 1989.