Finite Element Analysis

Josh Greenberg

University of Connecticut

4/22/2021

Textbook

The Finite Element Method: Theory, Implementation, and Applications
Larson, Mats G., Bengzon, Fredrik

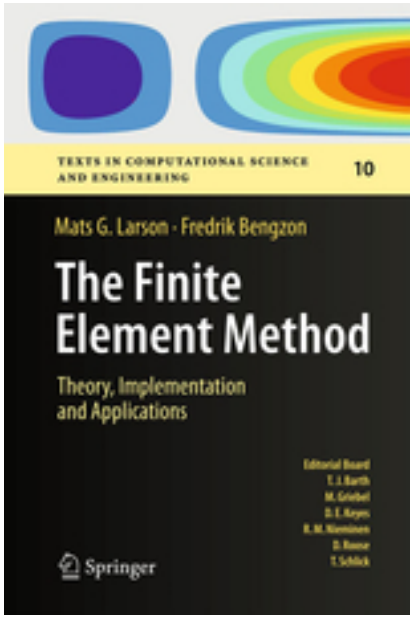
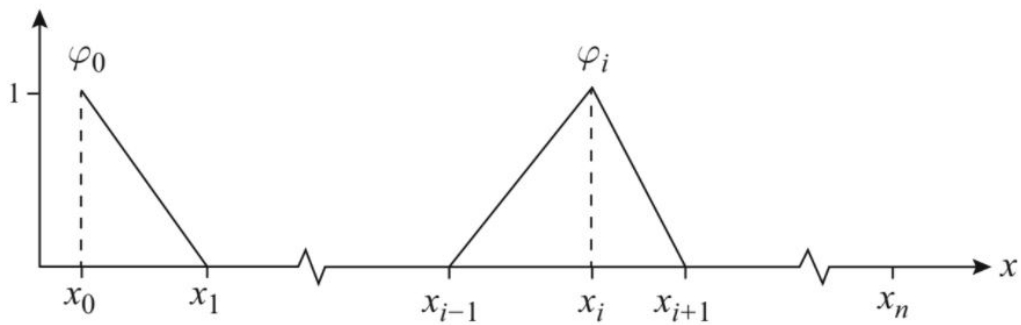


Table of Contents

Basis of an abstract vector space

We are estimating solutions of differential equations on a finite interval, or a finite region of R^2. The estimation will consist of piecewise-continuous linear functions. We need a basis to represent those functions.

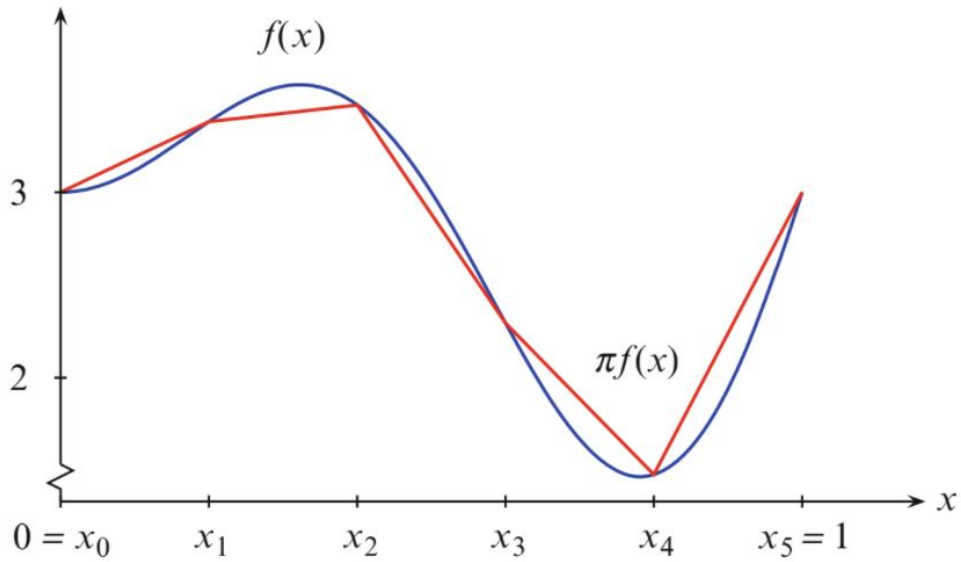
$$\varphi_i = \begin{cases} (x - x_{i-1})/h_i, & \text{if } x \in I_i \\ (x_{i+1} - x)/h_{i+1}, & \text{if } x \in I_{i+1} \\ 0, & \text{otherwise} \end{cases}$$



Estimating Functions

The simplest way to estimate a function is to use a point on the function and place a line segment on that point that connects to the next point of the function.

Fig. 1.4 The function $f(x) = 2x \sin(2\pi x) + 3$ and its continuous piecewise linear interpolant $\pi f(x)$ on a uniform mesh of $I = [0, 1]$ with six nodes x_i, $i = 0, 1, \ldots, 5$

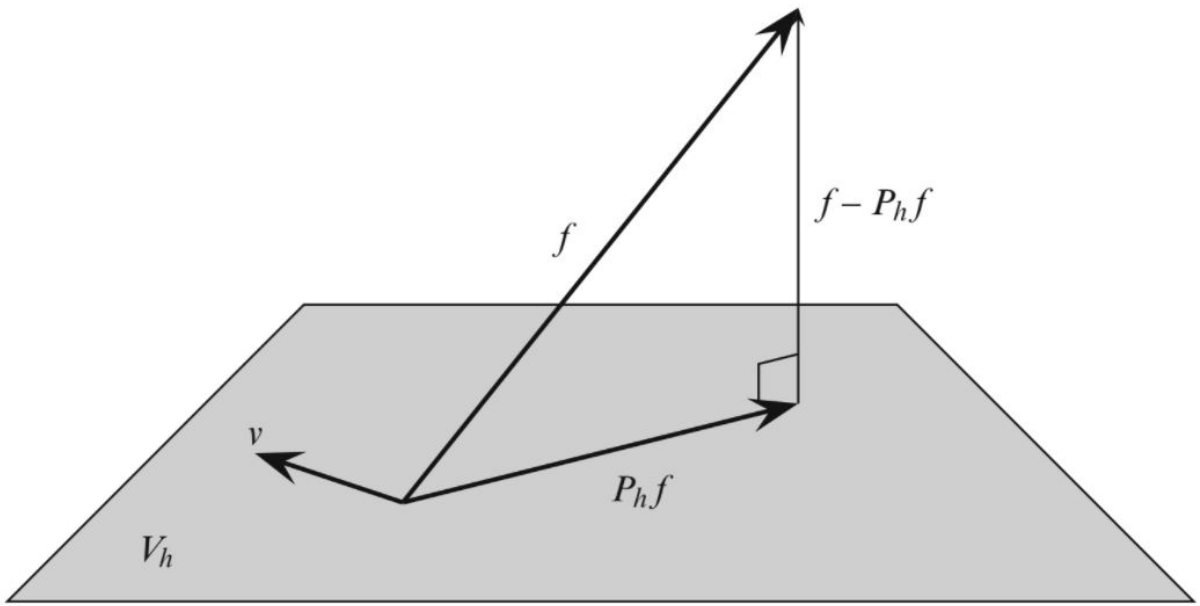


L-2 Projection

The L-2 projection method is better, but it is much harder to construct

It is also much more grounded in linear algebra

1.3 L^2-Projection



L-2 Projection Cont

Notice that $f - P_h f$ is orthogonal to the entire span of V_h (i.e. the span of the hat functions)

Therefore, the dot product with any linear combination of hat functions is zero

Since we are working with functions, we need the integral form of the dot product

$\int_I (f - P_h f)\phi_i dx = 0,\ i = 0, 1, 2, ..., n$

This condition is how we will construct the L-2 projection

The most important derivation

The equation in the previous slide implies that
$\int_I f\phi_i dx = \int_I P_h f \phi_i dx$

$P_h f$ is the projection, so it is a linear combination of the hat functions.

Therefore, it follows that $P_h f = \sum_{j=0}^{n} \alpha_j \phi_j$

We interchange the sum and the integral to get,
$\int_I f\phi_i dx = \sum_{j=0}^{n} \alpha_j \int_I \phi_j \phi_i dx, \quad i = 0, 1, ..., n$

The most important derivation cont.

Let the matrix $M_{ij} = \int_I \phi_j \phi_i dx, \quad i, j = 0, 1, ..., n$

Let the vector $b = \int_I f \phi_i dx \quad i = 0, 1, ..., n$

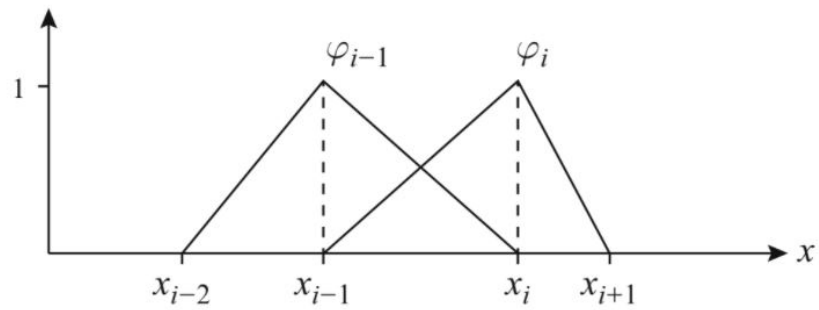
It follows that the equation in the previous slide is equivalent to the matrix equation in $(n + 1)$ dimensions,
$M\alpha = b$

Back to the L-2 Projection

Now that we have the matrix equation, we need to extract patterns to implement this into MATLab

M isn't actually so complicated

Fig. 1.9 Illustration of the hat functions φ_{i-1} and φ_i and their support



The picture shows that only the adjacent hat functions have a nonzero contribution to the integral of their product over the interval I
Therefore, the only nonzero entries of M are M_{ii}, $M_{i,(i+1)}$, $M_{(i+1),i}$

Back to the L-2 Projection cont

$$M_{ii} = \int_I \phi_i \phi_i dx, \quad = \int_{x_{i-1}}^{x_i} \phi_i \phi_i dx + \int_{x_i}^{x_{i+1}} \phi_i \phi_i dx$$

These integrals can be estimated using whichever technique (rectangles, trapezoids, etc) works best for the given situation.

If we use Simpson's rule, the integral works out to be $\frac{h_i}{3} + \frac{h_{i+1}}{3}$ $i = 1, ..., n-1$

For $i = 0$ and $i = n$, the integral turns out to be $\frac{h_1}{3}$ and $\frac{h_n}{3}$, respectively

Back to the L-2 Projection cont

For $M_{i+1,i}$, and $M_{i,i+1}$ the integral only runs over one sub interval.

Repeating the procedure in the last slide, we get that $M_{i+1,i}$ and $M_{i,i+1}$ both equal $h_{i+1}/6, \quad i = 0, ..., n$

If all this is confusing, the final picture is this matrix:

$$M = \begin{bmatrix} \frac{h_1}{3} & \frac{h_1}{6} & & & & & \\ \frac{h_1}{6} & \frac{h_1}{3} + \frac{h_2}{3} & \frac{h_2}{6} & & & & \\ & \frac{h_2}{6} & \frac{h_2}{3} + \frac{h_3}{3} & \frac{h_3}{6} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} + \frac{h_n}{3} & \frac{h_n}{6} & \\ & & & & \frac{h_n}{6} & \frac{h_n}{3} & \end{bmatrix}$$

Back to the L-2 Projection cont

Now we tackle how to assemble this in MATLab. We extracted the pattern from the matrix. Now we run this algorithm:

```
function M = MassAssembler1D(x)
n = length(x)-1;    (number of subintervals)
M = zeros(n+1,n+1);    (allocate mass matrix)
for i = 1:n    (loop over subintervals)
h = x(i+1) - x(i);    (interval length)
M(i,i) = M(i,i) + h/3;    (add h/3 to M(i,i))
M(i,i+1) = M(i,i+1) + h/6;
M(i+1,i) = M(i+1,i) + h/6;
M(i+1,i+1) = M(i+1,i+1) + h/3;
end
```

Back to the L-2 Projection cont

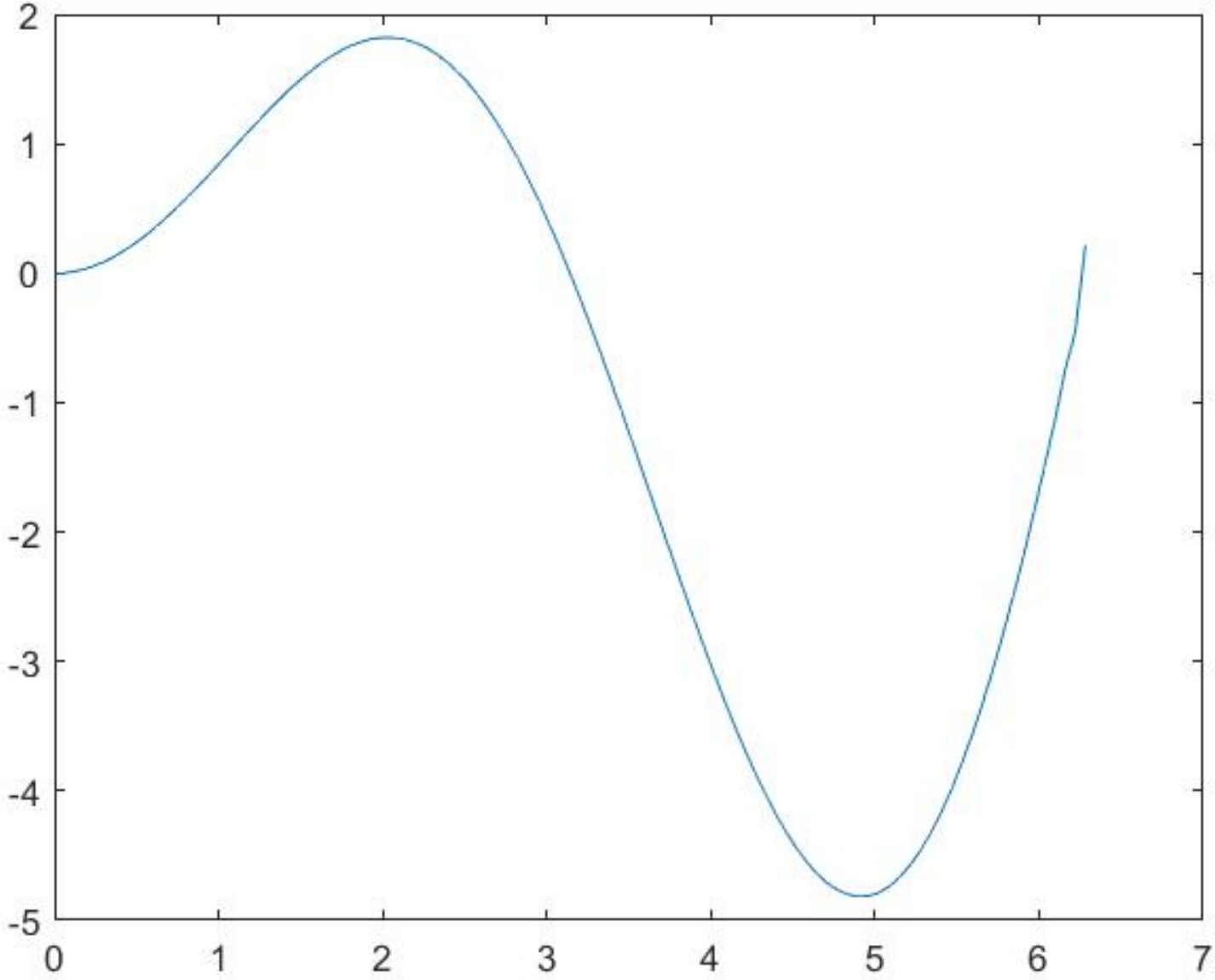
We will use a similar technique to compute the vector
$b = \int_I f \phi_i dx \quad i = 0, 1, ..., n$
In this case, f is the function we are trying to project and estimate, so it is known.
We estimate the integrals using whatever technique we wish, and we extract a similar pattern to implement into another for-loop into MATLab.
As an example,

Back to the L-2 Projection cont

Finally, we use this algorithm to put it all together and generate the L-2 Projection:

```
function L2Projector1D()
n = 100;      number of subintervals
h = 2*pi/n;      mesh size
x = 0:h:2*pi;      mesh
M = MassAssembler1D(x);      assemble mass
b = LoadAssembler1D(x,y.*sin(y));      assemble load
Pf = M/b;      solve linear system
plot(x,Pf)      plot L^2 projection
```

Back to the L-2 Projection cont



Solving a Differential Equation

Consider the second order ODE $-u'' = f$,
$x \in I = [0, L], \quad u(0) = u(L) = 0 \quad f$ is a given function

If we are going to construct the solution out of the hat functions, then we need to convert this from a differential equation into an integral equation.

We introduce the variational formulation:

Solving a Differential Equation

We start by multiplying the ODE by a test function v which vanished at the endpoints, and then integrating

Therefore we have:
$\int_0^L fvdx = -\int_0^L u''vdx = \int_0^L u'v'dx - u'(L)v(L) + u'(0)v(0)$

The two nonintegrable terms vanish, and we are left with,
$\int_0^L fvdx = \int_0^L u'v'dx$, which looks almost identical to what we had before.

Solving a Differential Equation

Remember that v is any function that vanishes at $x = 0, L$. The good news is that there are plenty of piecewise linear functions with that property, and the phi-hat functions just so happen to be in that group, as long as we eliminate the hat functions on the ends.

Therefore, we have:
$\int_0^L f\phi_i dx = \int_0^L u'\phi_i' dx \quad i = 1, 2, ..., n-1$

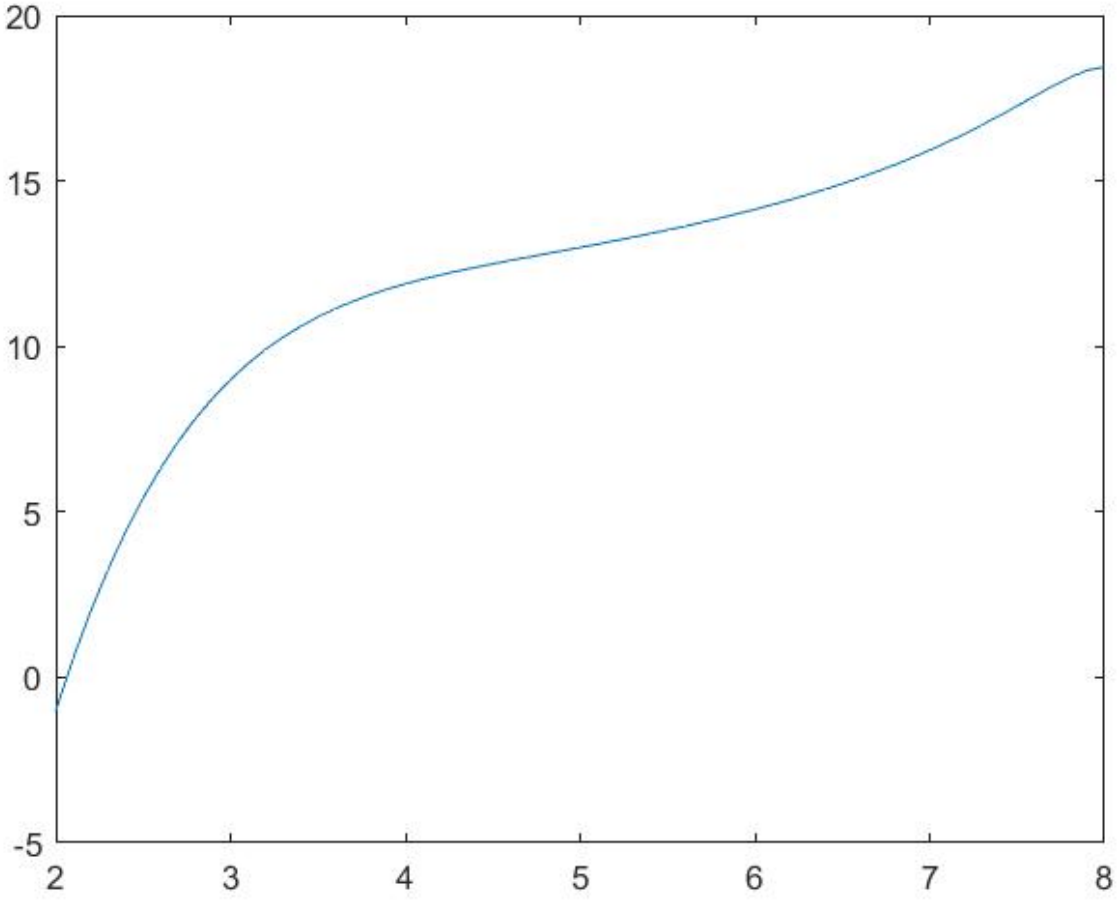
Solving a Differential Equation

u is the solution to the differential equation, so we are trying to construct it from the hat functions. We replace u and u' with u_h and u_h' to signify that it is being constructed from the hat functions.

That implies that $u_h = \sum_{j=0}^{n} \alpha_j \phi_j$ and we can repeat the exact same procedure as before to assemble the appropriate matrix and vector.

Solving a Differential Equation

$$-(0.5 + 0.7x)\,T'' = 0.3x^2,$$
$$2 < x < 8, \qquad T(2) = -1,\ T'(8) = 0$$

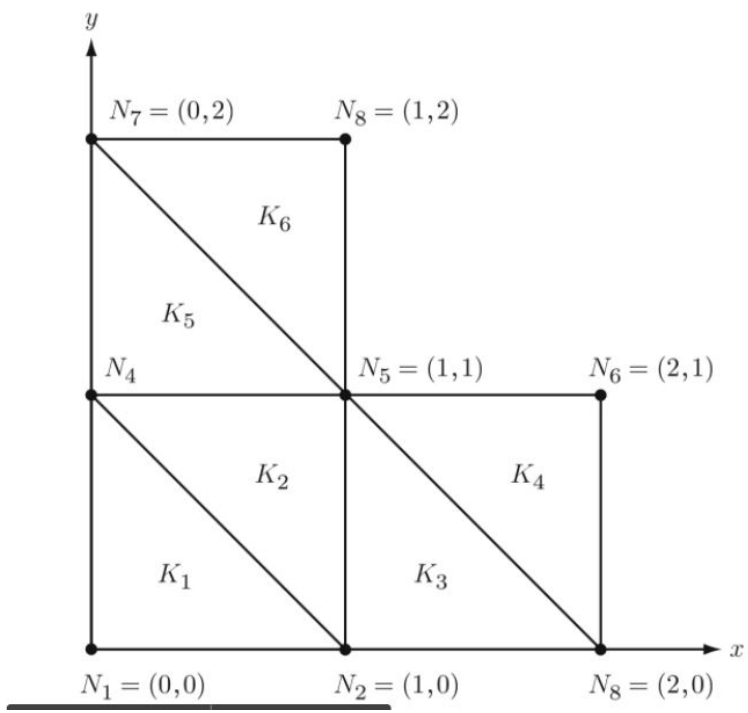


The 2 Dimensional Case

For 2 dimensions, we need to figure out how to partition the x-y plane

The answer is triangles:

Fig. 3.2 A *triangle* mesh of the L-shaped domain



The 2 Dimensional Case

$$P = \begin{bmatrix} 0.0 & 1.0 & 2.0 & 0.0 & 1.0 & 2.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 1.0 & 1.0 & 2.0 & 2.0 \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 2 & 5 & 3 & 4 & 5 \\ 2 & 5 & 2 & 6 & 5 & 8 \\ 4 & 4 & 8 & 5 & 7 & 7 \end{bmatrix}$$

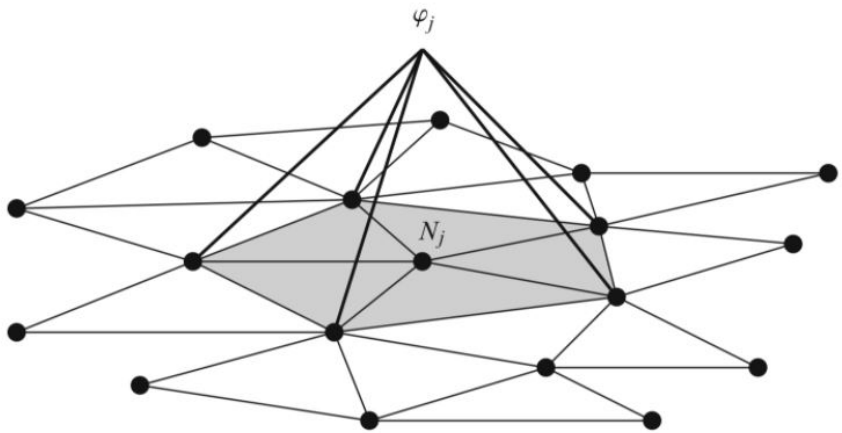


Fig. 3.4 A two-dimensional hat function φ_j on a general *triangle* mesh

The 2 Dimensional Case

4.2.1 Poisson's Equation

Let us consider Poisson's equation: find u such that

$$-\Delta u = f, \quad \text{in } \Omega \tag{4.5a}$$

$$u = 0, \quad \text{on } \partial\Omega \tag{4.5b}$$

where $\Delta = \partial^2/\partial x_1^2 + \partial^2/\partial x_2^2$ is the Laplace operator, and f is a given function in, say, $L^2(\Omega)$.

To derive a variational formulation of Poisson's equation (4.5) we multiply $f = -\Delta u$ by a function v, which is assumed to vanish on the boundary, and integrate using Green's formula.

$$\int_\Omega f v dx = -\int_\Omega \Delta u v \, dx \tag{4.6}$$

$$= \int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} n \cdot \nabla u v ds \tag{4.7}$$

$$= \int_\Omega \nabla u \cdot \nabla v \, dx \tag{4.8}$$

$$\int_\Omega \nabla u_h \cdot \nabla \varphi_i \, dx = \int_\Omega f \varphi_i \, dx, \quad i = 1, 2, \ldots, n_i \tag{4.14}$$

Then, since u_h belongs to $V_{h,0}$ it can be written as the linear combination

$$u_h = \sum_{j=1}^{n_i} \xi_j \varphi_j$$

Time Dependence

Let us consider the model Heat equation

$$\dot{u} - u'' = f, \quad x \in I = [0, L], \quad t \in J = (0, T] \tag{5.22a}$$

$$u(0, t) = u(L, t) = 0 \tag{5.22b}$$

$$u(x, 0) = u_0(x) \tag{5.22c}$$

where $u = u(x, t)$ is the sought solution, $f = f(x, t)$ a given source function, and $u_0(x)$ a given initial condition.

Multiplying $f = \dot{u} - u''$ by a test function $v = v(x, t)$ and integrating by parts we have

$$\int_0^L f v \, dx = \int_0^L \dot{u} v \, dx - \int_0^L u'' v \, dx \tag{5.23}$$

$$= \int_0^L \dot{u} v \, dx - u'(L)v(L) + u'(0)v(0) + \int_0^L u' v' \, dx \tag{5.24}$$

$$= \int_0^L \dot{u} v \, dx + \int_0^L u' v' \, dx \tag{5.25}$$

Time Dependence

Next, we seek a solution u_h to (5.29), expressed for every fixed t, as a linear combination of hat functions $\varphi_j(x)$, $j = 1, 2, \ldots, n-1$, and time-dependent coefficients $\xi_j(t)$. That is, we make the ansatz

$$u_h(x,t) = \sum_{j=1}^{n-1} \xi_j(t)\varphi_j(x) \tag{5.30}$$

and seek to determine the vector

$$\xi(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \vdots \\ \xi_{n-1}(t) \end{bmatrix} = \begin{bmatrix} u_h(x_1,t) \\ u_h(x_2,t) \\ \vdots \\ u_h(x_{n-1},t) \end{bmatrix} \tag{5.31}$$

$$M\dot{\xi}(t) + A\xi(t) = b(t), \quad t \in J \qquad \dot{\xi}(t) + \bar{A}\xi(t) = \bar{b}(t) \qquad \xi(t) = \xi(0)e^{-\bar{A}t} + \int_0^t e^{-\bar{A}(t-s)}\bar{b}(s)\,ds$$

Time Dependence