The Finite Element Basis for Simplices in Arbitrary Dimensions

John Burkardt
Department of Scientific Computing
Florida State University
https://people.sc.fsu.edu/~jburkardt/presentations/fem_basis_2013_fsu.pdf

04 October 2013

1 Introduction

This informal technical report describes a procedure for defining and evaluating a finite element basis for simplices. The polynomial degree of the basis set and the spatial dimension of the simplex are both arbitrary.

The approach is a generalization of commonly-used families of Lagrange basis functions. These families typically are defined by selecting a certain set of special nodes inside the simplex, and then associating with each node a corresponding Lagrange polynomial. The resulting family has the property that each basis function is 1 when evaluated at its associated node, and 0 at all other nodes.

For a low spatial dimension and low polynomial degree, it is possible to construct such a family by inspection. However, as the spatial dimension or polynomial degree increases, it is impossible to generate a family of basis functions unless a systematic approach is available.

We begin the discussion with some technical details that are unnecessary to the specialist, who may proceed to the last two sections where the main material is presented.

2 The Beauty of Convex Combinations

It is a surprise that the intuitive property of convexity can be captured in a simple mathematical definition. We say that a shape, in 1D, 2D, 3D, or general dimensions, is convex, if and only if, for any pair of points x and y that belong to the shape, it is also true that the shape includes all points of the form

$$z = \lambda x + (1 - \lambda)y, \quad 0 \leq \lambda \leq 1 \tag{1}$$

Figure (**??**) illustrates a non-convex figure, and a pair of points for which the convexity test fails. The figure also illustrates the fact that Equation (**??**) defines a line segment.

This definition of convexity can be used in a surprising variety of situations. For instance, a programmer often comes across a situation in which values x_1 and x_2 are given, and it is desired to compute n values equally spaced between them. Depending on the situation, the endpoints might or might not be included. A straightforward way to carry out such a computation is as follows:

```
dx = ( x2 - x1 ) / n
x = x1
for i = 1 to n
  x = x + dx
```

but this actually does not compute n points; rather, it takes n steps, and computes $n + 1$ points. Maybe that's what we really wanted, but we will find it important, as we go, to be clear about some counting issues.

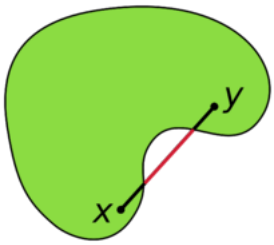

Figure 1: A 2D shape that fails the convexity test.

This incremental approach has several disadvantages. For instance, there is no direct formula given for the value of the i-th point, and there is the minor issue that, because of roundoff, in most cases the final point is unlikely to exactly equal x_2.

The clumsiness of the approach is more obvious if we are asked, once we have computed these points, to compute a new set of points that are midway between the ones we just computed. How many are there? *Probably n, right?* How do we get to the first one? *Start at x_1, but add a half step.* How do we proceed? *Now we just add full steps as before.* How would we compute just the 1000-th point? *I guess we take 999 steps!*

A more natural approach is to think about the fact that the i-th point between x_1 and x_2 should represent a kind of fraction of the distance between the endpoints. We might want to say something like

$$\frac{i}{n} \sim \frac{x}{x_2 - x_1} \quad ???$$

but that can't be right; we want the top of the fraction on the right to represent the distance from x_1:

$$\frac{i}{n} \sim \frac{x - x_1}{x_2 - x_1}$$

This is the right approach, but let's rearrange it as a formula that tells us the value of x we get when we choose i:

$$x = \frac{(n-i)x_1 + ix_2}{n} \tag{2}$$

or, in "computerese":

```
x = ( ( n - i ) * x1
  +   (     i ) * x2 )
  /   ( n     )
```

I promise, this is the same formula. But the form of the formula is extremely revealing. It suggests that the coefficients $(n-i)$ and i need to add up to n in order for things to work. It makes it clear that we are using these two coefficients to define a mixture of x_1 and x_2 constrained so that the proportions add up to 1. The value of n is essentially our "ruler" that defines n equal subintervals of $[x_1, x_2]$, and the $n+1$ equally spaced points that correspond to the fractions $\frac{0}{n}, \frac{1}{n}, ..., \frac{n}{n}$. Finally, we note that when $i = 0$, we compute x_1, and when $i = n$ we compute x_2.

If we are using FORTRAN or MATLAB, we may prefer our indices to run from 1 to n, and in C we might wish to go from 0 to $n-1$. Or we may want n to represent the number of intervals between the $n+1$ points we generate. All of these "minor details" can make it very difficult to write down the correct formula for a given case, even though the idea is intuitively simple.

So let's start by considering problems which can be described by drawing a certain number of equally spaced points which have consecutive indices. Suppose that x is one of these points, and z is another, and

that these points have indices i and k. Then, to compute the location of y, the j-th point, we simply write

$$y(j) = \frac{(k-j)x + (j-i)z}{k-i} \tag{3}$$

Now, let's pose some problems and see if we can determine the correct incarnation of Equation (**??**) that must be applied.

1. We want to compute 25 equally spaced points between 10 and 17. The first point is labeled 1 and the last point is labeled 25. What is a formula for the j-th point?

$$y(j) = \frac{(25-j)10 + (j-1)17}{25-1}$$

2. Our ODE solver just took 100 equal steps from t = 5.0 to t = 7.0 (so we have 101 solution values). Suppose we are interested in the midpoints of these 100 intervals. What is a formula? Suppose we label the first point 0 and the last point 100. The midpoints can be thought of as having fractional indices, so that the first midpoint has $j = \frac{1}{2}$ and the last is $j = 99\frac{1}{2}$. The formula for the j-th point is simply:

$$y(j) = \frac{(100-j)5.0 + (j-0)7.0}{100-0}$$

and in particular, $y(\frac{1}{2}) = 5.01$.

3. Suppose the interval [9,12] is divided into 75 equal subintervals, thus defining 76 points. We want to write a C loop, using index j, to compute the interior points, that is, the 74 points between, and not including, 9 and 12. We'd prefer to describe this problem in terms of the endpoints. In that case, 9 would implicitly have index -1, since C wants to count from 0, and 12 would implicitly have index 74. We want to execute a C loop that, for $j = 0$ to $j = 73$, evaluates the formula

$$y(j) = \frac{(74-j)9.0 + (j--1)12.0}{74--1} = \frac{(74-j)9.0 + (j+1)12.0}{75}$$

4. The interval $[\pi, 2\pi]$ has been divided into 16 intervals by 17 equally spaced points. We want to evaluate a function at each of these points, and at three points before and three points after the interval, using the same spacing. We are using MATLAB, and we are storing the values in an array, so the first point we compute must be labeled $j = 1$. OK, that means points 1, 2, and 3 are outside the interval, and point 4 is at 0.0. Thinking further, we see that 1.0 will have the index $j = 20$. Therefore, the formula for the points $j = 1$ through 23 is:

$$y(j) = \frac{(20-j)\pi + (j-4)2\pi}{20-4}$$

From these examples, I hope it is clear that, in order to determine the appropriate convex combination formula, you simply need to determine the indexing of your equally spaced points, and from that, the index and value of two points in your set. If you can correctly specify that information, you can write down a formula to generate all the points using your indexing scheme. Moreover, the indexing can involve negative values, fractions, or even real values.

Let us return to calling our two known values x_1 and x_2. We can regard our formula as computing a linear combination of the two values, using coefficients we can denote by ξ_1 and ξ_2, as follows:

$$\begin{aligned} \xi_1 &= \frac{x_2 - x}{x_2 - x_1} \\ \xi_2 &= \frac{x - x_1}{x_2 - x_1} \\ x &= \xi_1 x_1 + \xi_2 x_2 \end{aligned}$$

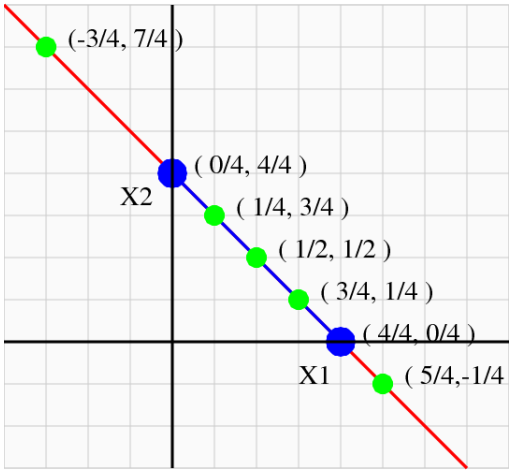

Figure 2: Line defined by combinations of x_1 and x_2.

These coefficients have some interesting properties. First, let us suppose that $x_1 < x_2$. It should be obvious that if ξ_1 is negative, x is to the left of x_1, while it is positive if x is to the right; the opposite behavior applies to ξ_2. And this means that x is inside the interval $[x_1, x_2]$ precisely if both ξ_1 and ξ_2 are non-negative, and strictly so if they are both positive.

We may think of ξ_1 as the *normalized signed distance* of the point x to the opposite reference point x_2. This distance is 0 if $x = x_2$, and is 1 if $x = x_1$, as though the line segment $[x_1, x_2]$ was being used as a reference length.

Suppose, now, that we have some function which passes through the points (x_1, f_1) and (x_2, f_2). Then the linear interpolant is easily constructed:

$$\begin{aligned} f(x) =& \xi_1 * f_1 + \xi_2 * f_2 \\ =& \frac{(x_2 - x) * f_1 + (x - x_1) * f_2}{x_2 - x_1} \end{aligned}$$

Is it right? The wonderful thing about a linear interpolant of this form is that, if it's right at two distinct arguments, it's right everywhere. And it's easy to see that $f(x_1) = f_1$ and $f(x_2) = f_2$ so we're done. Now notice that the formula for x in terms of ξ_1 and ξ_2 is, in fact, simply another example of linear interpolation, in which the functional values are the x-coordinates of the endpoints of the interval. This way of writing a linear interpolant is symmetric in the data, and makes its dependence on the data clear, unlike the high school formula $y(x) = a * x + b$ that requires the computation of a slope and intercept from the original data.

Now we seem to have started with a 1-dimensional problem involving x, and complicated it into a 2-dimensional problem involving (ξ_1, ξ_2). We can see in Figure (**??**) how the ξ coordinates parameterize the line, and how the interior of the line segment corresponds to both coordinates being positive.

Although we have been considering the ξ coordinates for a fixed value of x, it is worth considering the behavior of the convex coefficients as functions of x. Figure (**??**) displays the unsurprising plot of the convex coefficient functions associated with the interval $[0, 1]$.

In many cases, it is useful to restrict the convex coefficient functions to the interior of the interval used to define them, and to set them to zero outside of that. Figure (**??**) also displays the result of restricting the convex coefficient functions for $[0, 1]$ in this way.

Note that the pair of functions $\xi_1(x)$ and $\xi_2(x)$ constitute a *partition of unity* over the interval $[x_1, x_2]$, which seems a roundabout way of noting the obvious fact that $\xi_1(x) + \xi_2(x) = 1$.

What might be less obvious is that $\xi_1(x)$ and $\xi_2(x)$ are related to the piecewise-linear functions that can be used as a basis for the finite element method. To see this more clearly, we have to imagine that a larger

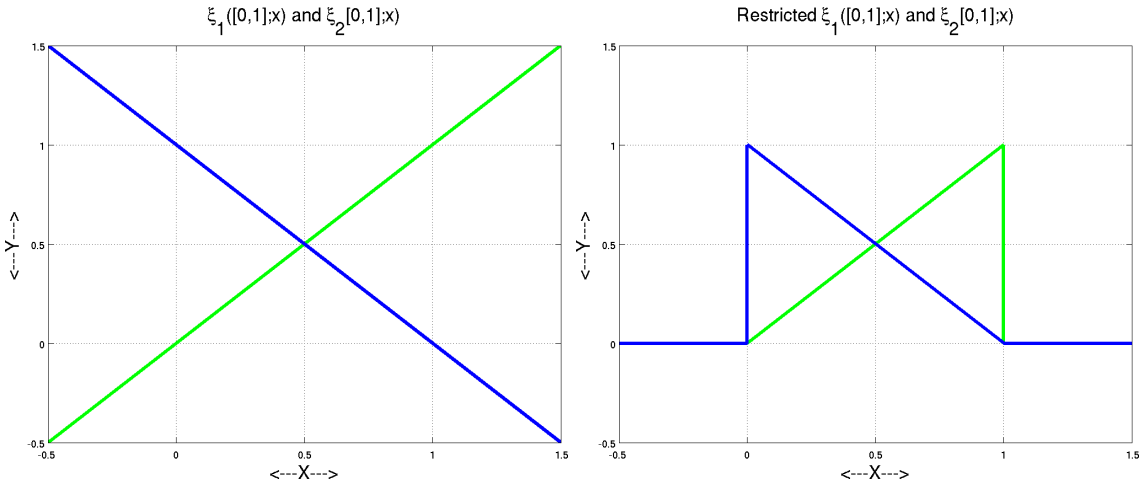

Figure 3: Unrestricted and restricted convex coefficient functions for [0,1].

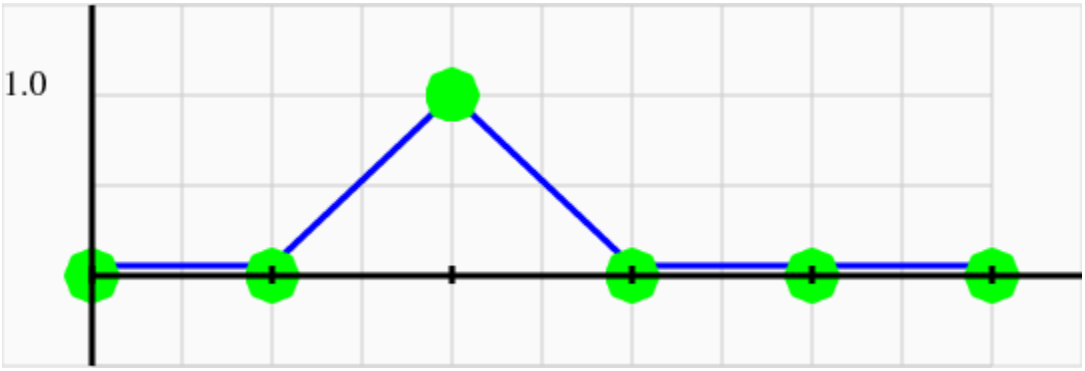

Figure 4: A finite element basis function from convex combination.

interval $[x_1, x_m]$ has been divided into subintervals $[x_1, x_2]$, ..., $[x_{m-1}, x_m]$ determined by a strictly increasing sequence of nodes x_1 through x_m. For each of these subintervals, we could define the functions $\xi_1(x)$ and $\xi_2(x)$. To indicate which subinterval we are using, we will use the notation $\xi_1(x_k, x_{k+1}; x)$ to indicate that, over the interval $[x_k, x_{k+1}]$, we wish to evaluate the $\xi_1(x)$ function that is 1 at x_k and 0 and x_{k+1}.

For each node x_k, we now construct an associated basis function $\phi_k(x)$ defined as follows:

$$\phi_k(x) = \begin{cases} \xi_2(x_{k-1}, x_k; x) & \text{if } x_{k-1} < x \leq x_k; \\ \xi_1(x_k, x_{k+1}; x) & \text{if } x_k \leq x < x_{k+1}; \\ 0 & \text{otherwise.} \end{cases}$$

In Figure (**??**) we suggest what finite element basis function $\phi_k(x)$ would look like, constructed using convex combinations from neighboring subintervals $[x_{k-1}, x_k]$ and $[x_k, x_{k+1}]$. Note also that the set of m basis functions $\{\phi_k(x) | 1 \leq k \leq m\}$ constitutes a partition of unity over $[x_1, x_m]$.

Now, consider the convex coefficients from a linear algebraic point of view. Given an interval $[x_1, x_2]$ and a point x whose convex coefficients are $\xi_1(x)$ and $\xi_2(x)$, there are two conditions that we know must hold:

$$\begin{aligned} x_1 * \xi_1 + x_2 * \xi_2 &= x \\ \xi_1 + \xi_2 &= 1 \end{aligned}$$

or, written in matrix form:

$$\begin{pmatrix} x_1 & x_2 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

The determinant of this matrix is $(x_1 - x_2)$, so it's invertible assuming only that $x_1 \neq x_2$, and the inverse matrix is

$$\frac{1}{x_1 - x_2} \begin{pmatrix} 1 & -x_2 \\ -1 & x_1 \end{pmatrix}$$

so to compute a formula for the ξ coordinates, we simply multiply by the inverse matrix, to get:

$$\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \frac{1}{x_1 - x_2} \begin{pmatrix} 1 & -x_2 \\ -1 & x_1 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x_2 - x}{x_2 - x_1} \\ \frac{x - x_1}{x_2 - x_1} \end{pmatrix}$$

Convex combinations, as discussed here, show up repeatedly in computational mathematics. For instance, if we have 1024 tasks, numbered 0 to 1023, to divide among 10 processors, numbered 0 through 9, then the i-th processor should start at the task indexed $lo(i)$ and stop just before the task $lo(i+1)$, where

```
lo(i) = floor ( ( ( 10 - i ) * 0
                +          i   * 1024 )
                 /   10      )
```

Normally, we would drop the first numerator since it multiplies 0, but keeping it helps us see that it's really a convex combination, and it also suggests how we would deal with a problem where the tasks were numbered 1492 to 1945, for instance.

Many orthogonal polynomials are defined in a way that involves convex combination. For instance, the Legendre polynomials, denoted by $p_i(x)$, evaluated at a point x, satisfy the relationship:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_{i+1}(x) &= \frac{(2i+1)xP_i(x) - iP_{i-1}(x)}{i+1} \end{aligned}$$

so that we have a sort of convex combination of xP_{i-1} and P_i.

So far, we have only considered convex combinations of two quantities, and have generated line segments. Can we extend this idea to three quanties? What sort of objects do we create then?

3 Convex Combinations as Triangles

The triangle is an elemental geometric shape. We think of the triangle as being defined by three points or "vertices". Depending on the situation, when we speak of a triangle, we might mean just those three points, or we might wish to include the lines that join them (called "edges"), or even all the points bounded by those lines as well.

It is this last idea that we wish to capture for this discussion, and we will need to formulate this mathematically. Therefore, we begin with three vertices, and try to describe a formula that can characterize the vertices, edges, and interior points of the triangle. In order to do this, it is useful to extend our notion of a convex combination to the case where three or more vertices are involved. Thus, we will say that the point p is a convex combination of the n vertices $V = \{v_i \| i = 1, \ldots n\}$, if p can be represented, using a coefficient vector $\vec{\xi}$, as

$$p = \sum_{i=1}^{n} \xi_i * v_i$$

with the conditions that every ξ_i is nonnegative, and $\sum_{i=1}^{n} \xi_i = 1$.

Now suppose we take as our set V the three vertices of the triangle. Then it is trivial to assert that any vertex is a convex combination of V with the property that exactly one coefficient ξ is nonzero. Granting this, we can see that any point on an edge is a convex combination of two vertices, in which case exactly two coefficients are nonzero. Since the triangle is convex, we already know that any point strictly in the interior is a convex combination of two points on the boundary, and it turns out that this will imply that all three coefficients will be nonzero. Thus, convexity allows us to characterize the vertices, edges, and interior points of the triangle in a natural and algebraic way.

Thus, any triangle T may be described as the set of all points p which are convex combinations of the vertices, that is

$$T(v_1, v_2, v_3) = \{p : p = \xi_1 v_1 + \xi_2 v_2 + \xi_3 v_3; \quad 0 \leq \xi_i; \quad \xi_1 + \xi_2 + \xi_3 \leq 1\}$$

The coefficients (ξ_1, ξ_2, ξ_3) form an alternative coordinate system for the plane. Assuming that the vertices of the triangle are not collinear, then there is a 1-1 correspondence between the Cartesian coordinates (x, y) and the triangular coordinates (ξ_1, ξ_2, ξ_3), often called the triangle's barycentric coordinates or area coordinates.

The rationale for the name "area coordinates" comes from the fact that we can determine the barycentric coordinates of any point p geometrically, If we let $A(a, b, c)$ represent the area of the triangle formed by the points a, b, c, then we have:

$$\begin{aligned}
\xi_1 &= \frac{A(p, v_2, v_3)}{A(v_1, v_2, v_3)} \\
\xi_2 &= \frac{A(v_1, p, v_3)}{A(v_1, v_2, v_3)} \\
\xi_3 &= \frac{A(v_1, v_2, p)}{A(v_1, v_2, v_3)}
\end{aligned}$$

In particular, note that the area of the triangle is

$$A(v_1, v_2, v_3) = \frac{1}{2}(x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))$$

This idea is illustrated in Figure (**??**). As long as p is inside the triangle, this formula is correct. If p is outside the triangle, then we know that at least one entry of ξ must be zero. It turns out that, if we use the convention that triangles with clockwise orientation have negative area, then the formula hold true even for those cases.

We need to know how to convert between the (x, y) and (ξ_1, ξ_2, ξ_3) coordinate systems. One direction is easy, of course. Let us denote the Cartesian coordinates of vertex v_1 by (x_1, y_1) and so on. Then, if we are given the ξ coordinates of a point, we can determine its (x, y) coordinates by using the first two of the following three formulas:

$$\begin{aligned}
x_1\xi_1 + x_2\xi_2 + x_3\xi_3 &= x \\
y_1\xi_1 + y_2\xi_2 + y_3\xi_3 &= x \\
\xi_1 + \xi_2 + \xi_3 &= 1
\end{aligned}$$

or, written in matrix form:

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

As long as the vertices of the triangle are not collinear, the system matrix is invertible, and we therefore can arrive at a formula for the ξ coordinates in terms of (x, y):

$$\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \frac{1}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \cdot \begin{pmatrix} x(y_2 - y_3) - (x_2 - x_3)y + x_2y_3 - x_3y_2 \\ x(y_3 - y_1) - (x_3 - x_1)y + x_3y_1 - x_1y_3 \\ x(y_1 - y_2) - (x_1 - x_2)y + x_1y_2 - x_2y_1 \end{pmatrix}$$

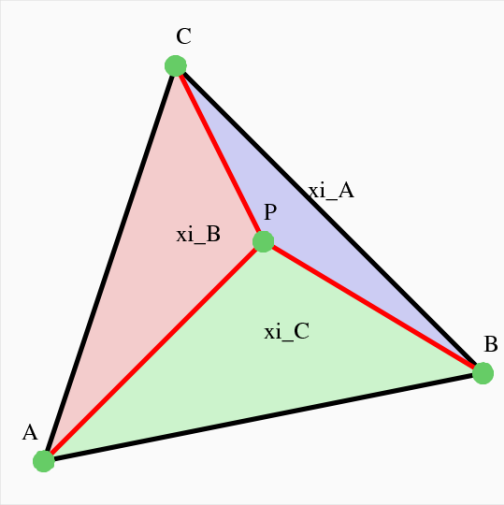

Figure 5: Barycentric coordinates are area ratios of subtriangles to the whole triangle.

For a given triangle, we may now consider the behavior of the coefficient functions xi. For instance, $\xi_1(x,y)$ is a linear function which is 1 at v_1 and 0 at v_2 and v_3. Hence, by linearity, it is zero at all points on the edge between v_2 and v_3. Corresponding statements hold for $\xi_2(x,y)$ and $\xi_3(x,y)$. Moreover, the three xi functions constitute a partition of unity over the triangle.

The linear interpolation property that we observed for the 2D case also carried over to the triangle. If f_1, f_2 and f_3 are values specified at the vertices, then the linear interpolant to this data is

$$f(x,y) = f_1\xi_1(x,y) + f_2\xi_2(x,y) + f_3\xi_3(x,y)$$

which is easily verified by checking the behavior of this function at the three vertices.

A fairly common occurrence in finite elements involves starting with a 2D region R, choosing an unstructured grid of points P that follow the boundary and sample the interior, and then using triples of points to define a triangulation of R. Now it is necessary to determine a basis of finite element functions, where each basis function $\phi_i(x,y)$ is associated with a particular point $p_i \in P$. The typical basis function can be defined by expressing its value at an arbitrary point (x,y). A few triangles in the triangulation will include p_i as a vertex. If (x,y) is interior to such a triangle, then the value of $phi_i(x,y)$ is the barycentric coordinate of (x,y) with respect to the vertex p_i. Otherwise, the function value is zero. This formulation gives a natural generalization of the "hat function" basis that arises when the finite element method is applied to a 1D interval that has been divided into subintervals. Figure (**??**) illustrates what such a basis function looks like.

4 The M-dimensional Simplex

We wish to generalize the idea of a triangle; a reasonable approach begins from the fact that the triangle is an object in two dimensions that is described by three vertices. If we look for a similar object in one dimension, we would use two vertices and thereby describe an interval. In three dimensions, four vertices will specify a tetrahedron. The pattern suggested by these three cases makes it possible to generalize the idea to an arbitrary spatial dimension.

The *M-dimensional simplex* S is defined by a set V of **M**+1 vertices v_i (points in **M**-dimensional space), and includes all points p which are convex combinations of those vertices:

$$S(V) = \{p : p = \sum_{i=1}^{M+1} \xi_i v_i; 0 \leq \xi_i; \sum_{i=1}^{M+1} \xi_i \leq 1\}$$

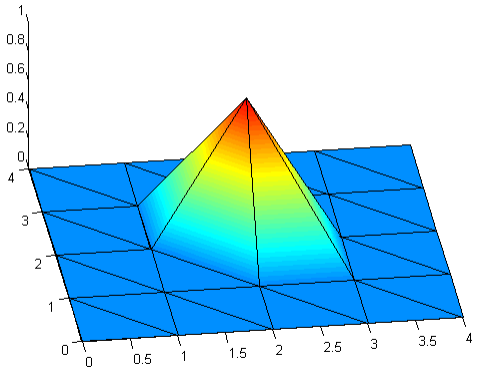

Figure 6: A finite element basis function associated with a particular node.

We will always assume that the simplex is not degenerate, or equivalently, that the vertices lie in what is called "general position". In the case of the triangle, this means we assume the three vertices do not lie on a line, which would result in a "flat" (and uninteresting!) triangle of zero area. In **M** dimensions, this requirement is equivalent to the condition that the **M**-dimensional volume of the simplex is not zero.

The "volume" (area) of a triangle with vertices $V = \{v_1, v_2, v_3\} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ is

$$\frac{1}{2!} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

and the volume of a tetrahedron is

$$\frac{1}{3!} \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix}$$

This formula generalizes, in the obvious way, to produce the volume of a simplex in **M** dimensions.

From the definition of the simplex, it should be clear that any point p contained in the simplex can be identified either by its Cartesian coordinates $\vec{x}$, but also by its barycentric coordinates $\vec{\xi}$ as long as we also know the vertices V of the simplex. We will find both coordinates systems useful, and so it will be convenient to determine the computational relationship between them.

5 The Reference Simplex

Let us suppose that we are working in an **M**-dimensional space, and we define a special simplex E by choosing as vertices the **M** unit Cartesian basis vectors $\vec{e_i}$, for i from 1 to M, as well as the origin $\vec{0}$. This object is known as the **M**-*dimensional reference simplex*.

The volume formula tells us that the reference simplex will have volume $\frac{1}{M!}$.

If we list the vertices of the reference simplex in the natural order, followed by the origin, then it turns out that for any point in the simplex, the Cartesian and barycentric coordinate systems are essentially identical.

For instance, in 2D, the correspondence $\vec{x} \iff \vec{\xi}$ for the reference triangle is simply

$$(x, y) \rightarrow (x, y, 1 - x - y) = (\xi_1, \xi_2, \xi_3)$$
$$(x, y) = (\xi_1, \xi_2) \leftarrow (\xi_1, \xi_2, \xi_3)$$

But now let us consider an arbitrary triangle T in 2D. Suppose we list its vertices as $V = (v_1, v_2, v_3)$ and associate these vertices, in this order, with the vertices of the reference triangle $(\vec{e_1}, \vec{e_2}, \vec{0})$. There is a unique linear mapping $\phi(v_1, v_2, v_3; \xi_1, \xi_2, \xi_3)$ which takes the vertices of the reference triangle to those of T, and its form is not hard to work out from the requirements that:

$$\begin{aligned} v_1 &= \phi(v_1, v_2, v_3; 1, 0, 0) \\ v_2 &= \phi(v_1, v_2, v_3; 0, 1, 0) \\ v_3 &= \phi(v_1, v_2, v_3; 0, 0, 0) \end{aligned}$$

However, this mapping also uniquely associates *every* point ξ in the reference triangle to a point p in T. Known as *the reference map*, it has the form:

$$\begin{aligned} p &= p(\xi_1, \xi_2, \xi_3) \\ &= \phi(v_1, v_2, v_3; \xi_1, \xi_2, \xi_3) \\ &= \xi_1 v_1 + \xi_2 v_2 + \xi_3 v_3 \\ &= V * \xi \end{aligned}$$

Assuming T is nondegenerate, this mapping is also invertible and so there is a corresponding function $\phi^{-1}(v_1, v_2, v_3; x, y)$ which maps points p in T to points ξ in the reference triangle.

Now the formula for ϕ is equivalent to the barycentric coordinates of the point p. This means that the barycentric coordinates of a point p are, at the same time, the coordinates of a representative point ξ in the reference triangle, and the coefficients that combine with the vertices of T to produce the point p.

These results extend to $\mathbf{M}$ dimensions, where there is a mapping $\phi(\mathbf{v}; \xi)$ from the reference simplex E to an arbitrary simplex S; if the simplex is nondegenerate, there is an inverse mapping as well.

6 Simplices in the Finite Element Method

An $\mathbf{M}$-dimensional simplex is in some ways the natural geometrical element into which to decompose an arbitrary geometric object. Because of its flexibility and economy, it is very frequently used in finite element analysis.

Of course, most geometries of interest correspond to physical objects, and hence are likely to be 1-, 2- or 3-dimensional. However, there are always situations in which higher-dimensional geometries are of interest, and the main point of our discussion will be that formulas that are obvious in 2D can be extended not merely to the more involved case of 3D, but can be automatically generalized to the $\mathbf{M}$-dimensional case.

A finite element analysis can be thought of as beginning with the definition of a geometric region to be studied. The region is then subdivided into numerous polygonal or polyhedral subregions, called *elements*. The most common choices for the element are rectangles or triangles and their higher dimensional generalizations. Our attention will focus on the case when a triangle, tetrahedron, or the general $\mathbf{M}$-simplex is the element being used.

The finite element analysis takes advantage of the ideas of the reference element and the reference mapping. Although the calculation is intended to deal with the entire region, the majority of the computation occurs while considering only the reference simplex. This is accomplished by considering in turn each one of the elements that cover the region, and determining the inverse mapping that establishes its correspondence with the reference element. After computations are completed on the reference element, the forward reference mapping is used to transfer the results back to the element.

Thus, a prime feature of the finite element method is that complicated geometries can be handled, but that the calculations occur on a single, simple region, which in our case will be a simplex.

7 The Lagrange Polynomial Basis For Rectangles

We are not interested in using rectangles, but rather triangular elements and their generalizations. However, the rectangular case is a useful starting point, since the procedure for triangles begins in the same way, but then requires some special steps.

The finite element calculations require the definition of a set of basis functions $\psi_j(\xi)$, defined over the reference element. Generally, the basis functions are polynomials; moreover, it is usually the case that a nested family of such basis functions is available, indexed by the parameter d, with each succeeding member of the family being able to achieve a higher order of polynomial approximation.

When the element being used is a rectangle, or its higher-dimensional generalizations, the orderly definition of these polynomials is easy - the d-th member of the nested family is a set of polynomials for which no variable has an exponent exceeding d. Such a basis can be formed from $(d+1)^M$ monomials, but we are free to form an equivalent basis of that size in a form that is more convenient for the finite element method. For rectangles, we start by choosing $d+1$ equally spaced points along each unit coordinate axis, and forming the Cartesian product of all possible pairs (x_i, y_j). We call this the set of *grid points*. To define our d-th basis family, we identify an arbitrary element of that family by associating it with one of the grid points. Then the (i, j) element of the set, written $\psi_{i,j}(x, y)$ is a polynomial with the properties that

- for every monomial term in $\psi_{i,j}(x, y)$, the exponents of x and y are between 0 and d;
- $\psi_{i,j}(x_i, y_j) = 1$
- $\psi_{i,j}(x_k, y_l) = 0$ if (x_k, y_l) is any grid point other than (x_i, y_j).

Because of the use of the grid points, it should be clear that a formula for the (i, j) element of the d-th family of basis function is

$$\psi_{i,j}(x, y) = \frac{\prod_{k=0;k\neq i}^{d}(x - x_k)\prod_{l=0;l\neq j}^{d}(y - y_l)}{\prod_{k=0;k\neq i}^{d}(x_i - x_k)\prod_{l=0;l\neq j}^{d}(y_j - y_l)}$$

A set of polynomials and points with the property that each polynomial is identified with exactly one point, at which it attains the value 1, while being zero at all other points, is known as a *Lagrange polynomial basis*, and has an extensive history and wide application, particularly in the area of interpolation.

For rectangular elements in higher dimensions, the procedures outlined here are readily extensible, requiring nothing but the most obvious modifications.

8 The Lagrange Polynomial Basis For Triangles

In the case of triangles or general simplex elements, the natural way to nest a family of basis functions imposes the requirement that each member of the d-th family have *total degree* no more than d. This means that it is now the sum of the exponents of x and y that must be no more than d, rather than applying that limitation to each exponent separately.

To find these functions for the case of a triangle, in dimension $M = 2$, we again resort to choosing $d + 1$ equally spaced points along each coordinate direction. We form the Cartesian product as before, but now many of the pairs will not be acceptable because they lie outside the triangle. We end up with $\binom{M+d}{M} = \frac{(d+1)(d+2)}{2}$ acceptable grid points compared to the $(d+1)^2$ points in the case of the corresponding calculation for a rectangle.

The rationale for producing the basis functions from the grid points is the same as in the rectangular case. Each basis function is to be associated with a grid point; it is 1 there and 0 at the other grid points; it has a total degree of d. The problem now is that coming up with formulas for these functions is significantly less obvious than it was for the rectangular case.

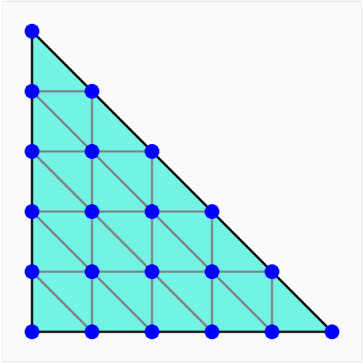

Figure 7: The reference triangle, with degree-5 gridlines.

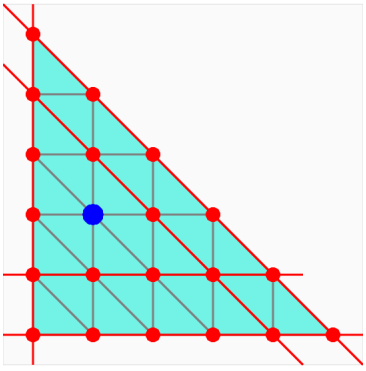

Figure 8: A node (in blue) and the 5 grid lines (in red) that each contribute a linear factor

However, we will now find it convenient to index the nodes with a sort of *barycentric index*, that corresponds with the barycentric coordinate system. That is, given the value $\mathbf{d}$, which represents the index of the polynomial basis family we are generating, we assign indices i and j to the nodes in the obvious way, but then add an auxiliary index $k = d - i - j$. Thus, the barycentric indices are nonnegative, and add to d. Moreover, a node whose barycentric indices are (i, j, k) has barycentric coordinates $(\frac{i}{d}, \frac{j}{d}, \frac{k}{d})$.

With this understanding, we can now write a formula for the polynomial basis function associated with the node whose barycentric index is (i, j, k):

$$L(i,j,k)(x,y) = \prod_{p=0}^{i-1}(x - \frac{p}{d}) \prod_{p=0}^{j-1}(y - \frac{p}{d}) \prod_{p=0}^{k-1}((1 - x - y) - \frac{p}{d})$$

but then we have to normalize by computing the value of this polynomial at the associated node.

$$\psi(i,j,k)(x,y) = \frac{L(i,j,k)(x,y)}{L(i,j,k)(x_{i,j,k}, y_{i,j,k})}$$

9 The Lagrange Polynomial Basis For Tetrahedrons

The procedure outlined in the previous section for triangles was framed in terms of scaled barycentric coordinates. To extend these results to dimension $M = 3$, where the simplex is the tetrahedron, it is enough to note that the same scaled barycentric coordinate system can be defined, once we have added another coordinate index to account for the higher spatial dimension.

Thus, in 3D, the scaled barycentric coordinate system for the tetrahedron involves a vector whose typical entries are represented by (i, j, k, l) and a scaled factor d. If a point is contained in the tetrahedron, its scaled barycentric coordinates must be nonnegative and their sum must lie between 0 and d. Moreover, there will be $\binom{M+d}{M}$ nodes, namely, the tetrahedral points whose scaled barycentric coordinates are all integers, and the same number of Lagrange basis functions.

We index a given Lagrange basis function using the scaled barycentric coordinates of its associated node. Thus, a typical Lagrange basis function could be represented as $L(i, j, k, l)(x, y, z)$.

For a given index (i, j, k, l), we can evaluate the Lagrange basis function as follows:

$$L(i,j,k,l)(x,y,z) = \prod_{p=0}^{i-1}(x - \frac{p}{d}) \prod_{p=0}^{j-1}(y - \frac{p}{d}) \prod_{p=0}^{k-1}(z - \frac{p}{d}) \prod_{p=0}^{l-1}((1 - x - y - z) - \frac{p}{d})$$

but then we have to normalize by dividing by the value of this polynomial at the associated node.

10 The Lagrange Polynomial Basis For Simplices

One of the primary advantages of the approach presented here is that we can now see how to generate the basis functions for an arbitrary degree d, and in a reference simplex of arbitrary dimension M.

To begin with, we know that the number of basis functions to be generated will be $\binom{M+d}{M}$. We choose $d + 1$ equally spaced points along each of the $\mathbf{M}$ coordinate axes to generate the grid points. We define the $M + 1$-dimensional barycentric grid index $\mathbf{i}$ in the corresponding way. For a point $\mathbf{x}$, we evaluate the Lagrange basis function by:

$$L(\mathbf{i};\mathbf{x}) = \prod_{p=0}^{\mathbf{i}_1-1}(\mathbf{x}_1 - \frac{p}{d}) \prod_{p=0}^{\mathbf{i}_2-1}(\mathbf{x}_2 - \frac{p}{d}) \ldots \prod_{p=0}^{\mathbf{i}_M-1}(\mathbf{x}_M - \frac{p}{d}) \prod_{p=0}^{\mathbf{i}_{M+1}-1}((1 - \mathbf{x}_1 - \mathbf{x}_2 \ldots - \mathbf{x}_M) - \frac{p}{d})$$

or, if we extend the vector $\mathbf{x}$ by an $\mathbf{M}$+1-th component, so that $\mathbf{x}_{M+1} = 1 - \sum_{i=1}^{M} \mathbf{x}_i$, we have

$$L(\mathbf{i};\mathbf{x}) = \prod_{j=1}^{M+1} \prod_{p=0}^{\mathbf{i}_j - 1} (\mathbf{x}_j - \frac{p}{d})$$

and we normalize by dividing by the value of this polynomial at the associated node to get $\phi(\mathbf{i};\mathbf{x})$.

11 MATLAB Code

The web page http://people.sc.fsu.edu/~jburkardt/m_src/fem_basis/fem_basis.html makes available MATLAB implementations of the various versions of the finite element basis function formulas. (Versions in C, C++, FORTRAN77 and FORTRAN90 are also available.) Here is the text for the function which handles the case of an $\mathbf{M}$-dimensional simplex.

```
function l = fem_basis_md ( m, i, x )

%*****************************************************************************80
%
%% FEM_BASIS_MD evaluates an arbitrary M-dimensional basis function.
%
%  Licensing:
%
%    This code is distributed under the GNU LGPL license.
%
%  Modified:
%
%    08 January 2011
%
%  Author:
%
%    John Burkardt
%
%  Parameters:
%
%    Input, integer M, the spatial dimension.
%
%    Input, integer I(M+1), the integer barycentric
%    coordinates of the basis function, 0 <= I(1:M+1).
%    The polynomial degree D = sum(I(1:M+1)).
%
%    Input, real X(M), the evaluation point.
%
%    Output, real L, the value at X of the basis function designated by I.
%


%
%  Augment the X vector.
%
  x(m+1) = 1.0 - sum ( x(1:m) );
%
```

```
%  Determine the degree.
%
  d = sum ( i(1:m+1) );

  l = 1.0;

  for q = 1 : m + 1
   for p = 0 : i(q) - 1
      l = l * ( d * x(q) - p ) / ( i(q) - p );
    end
  end

  return
end
```