Computational Geometry Lab:
FEM BASIS FUNCTIONS IN TRIANGULATIONS

John Burkardt
Information Technology Department
Virginia Tech
http://people.sc.fsu.edu/∼jburkardt/presentations/cg_lab_fem_basis_triangulation.pdf

March 25, 2024

1 Introduction

This lab continues the topic of *Computational Geometry*. Having studied triangles and how triangles are used to create triangulations of a region, we will now turn to the use of triangulations in the finite element method.

The finite element method is a procedure for approximating and solving partial differential equations. Part of the finite element method involves constructing the triangulation, a topic which is discussed in other labs. Once the triangulation is available, the finite element method uses this mesh to represent functions $f(x, y)$. The representation is *discrete*, that is, it depends on just a finite number of values, but the resulting function is defined over the entire triangulated region; with some restrictions, it can be evaluated, plotted, differentiated or integrated.

If you have ever used a finite difference method to solve differential equations, you will understand an important distinction between these two methods. The finite difference method works with values of a function at given points, but it does not try to "fill in the gaps" between the tabulated points. In contrast, the finite element method may only have exact knowledge of a function at specified points, but it builds a "model" of the function over the entire problem domain.

The key to this model building is the set of **finite element basis functions**. It is the purpose of this lab to understand how these basis functions are defined, evaluated and used to create the finite element functions.

2 Overview

Our discussion of this problem has already involved several steps. Although the arithmetic details can seem complicated, each of the steps completes a part of the solution to our overall problem. In previous labs, we have already seen discussions of the geometric properties of triangles, the representation of triangulations as a collection of points, and triples of node indices, the construction of an interpolation function on a single reference triangle, and interpolation on a general triangle.

Now it is time to take the next steps. We begin by looking at a triangulation, selecting an arbitrary node P_i, constructing a function $\phi_i(x, y)$ over the triangulation, which has the value 1 at P_i, but is zero at all other nodes. We will show what a typical such function looks like, as defined over the triangulation.

We will thus create a set of basis functions which completely solve our problem of constructing a general interpolating function.

We will show how to evaluate this interpolating function

3 Moving to the Triangulation

Now let's consider our full interpolation problem, which was posed on a triangulation. How do we propose to handle this? Suppose we are given a point (x, y), so that we have to compute the value $f(x, y)$ of our interpolation function. How do we proceed?

First, of course, we must determine the triangle that contains (x, y). A few special cases need to be considered. If (x, y) occurs in *no* triangle, then it falls outside the triangulated region, and we will simply return a zero value or an error condition. After all, we were only asked to interpolate over the triangulated region. If (x, y) occurs in *several* triangles, then we can simply choose one of the triangles. Of course, we must verify that the result will be the same no matter which triangle we choose - in other words, our definition of $f(x, y)$ is continuous.

Once we have located the triangle $\mathbf{T_i}$ containing (x, y), then we need to retrieve the vertices {a,b,c} of the triangle. Actually, we will need to retrieve the *indices* of these vertices as they appear in the list of nodes, because that will allow us to retrieve the corresponding function values.

In other words, triangle $\mathbf{T_i} = \{i_a,\, i_b,\, i_c\}$. Then we can use these indices as keys to both the NODE_XY array and the NODE_VALUE arrays.

Now let us return to the question of continuity. Suppose (x, y) occurs in multiple triangles.

If the point is a vertex in one triangle, it must be a vertex in all the triangles, because we do not allow "hanging nodes" in our triangulations. Therefore, (x, y) is not just a vertex of the triangle, but also a node of the triangulation, and has a corresponding node index i. The definition of $f(x, y)$ at a vertex with node index i is f_i, so our result is the same no matter which triangle we choose.

If (x, y) occurs in multiple triangles, but is not a vertex, then it must be a point shared by exactly two triangles with a common edge. But, as we have seen, on any edge of the triangle, the only two nonzero basis functions are those associated with the endpoints of the edge. The value of the function is the linear interpolant of the values at the two endpoints. No matter which triangle we choose, the endpoint values will be the same, and hence the value of the linear interpolant at (x, y) will be the same.

Thus, the function $f(x, y)$ is well defined. It is continuous because it is a linear function over each triangle, where any triangles have a common point, the function definitions coincide. Therefore, the function $f(x, y)$ is a (continuous) piecewise linear function over the triangulated region.

4 The Support of One Basis Function

5 Program #6: Finite Element Functions

Write a program which accepts three triangle vertices **Va, Vb, Vc** a set of three values associate with the vertices, **Wa, Wb, Wc** and a point **P**.

For the given point **P**, generate the barycentric coordinates $(\xi_a(P), \xi_b(P), \xi_c(P)$. Evaluate $f(P)$, the linear function which has the values **Wa, Wb, Wc** at the points **Va, Vb, Vc**.

Some simple checks include the following:

- setting **Wa, Wb, Wc** to (1,0,0) should mean $f(P) = \xi_a(P)$;
- setting **P = Va** should result in $f(P) = Va$;
- setting **P = (Va+Vb+Vc)/3** should result in $f(P) = (Wa + Wb + Wc)/3$;