

Computational Geometry Lab: BARYCENTRIC COORDINATES IN TRIANGLES

John Burkardt
Information Technology Department
Virginia Tech

http://people.sc.fsu.edu/~jburkardt/presentations/cg_lab_barycentric_triangles.pdf

August 28, 2018

1 Introduction to this Lab

In this lab we investigate a natural coordinate system for triangles called the **barycentric** coordinate system. The barycentric coordinate system gives us a kind of standard chart, which applies to all triangles, and allows us to determine the positions of points independently of the particular geometry of a given triangle.

Our path to this coordinate system will identify a special **reference triangle** for which the standard (x, y) coordinate system is essentially identical to the barycentric coordinate system.

We begin with some simple geometric questions, including determining whether a point lies to the left or right of a line. If we can answer that question, we can also determine if a point lies inside or outside a triangle. Once we are comfortable determining the (signed) distances of a point from each side of the triangle, we will discover that the relative distances can be used to form the barycentric coordinate system.

This barycentric coordinate system, in turn, will provide the correspondence or mapping that allows us to establish a 1-1 and onto relationship between two triangles.

The barycentric coordinate system is useful when defining the basis functions used in the finite element method, which is the subject of a separate lab.

2 Signed Distance From a Line

In order to understand the geometry of triangles, we are going to need to take a small detour into the simple world of points and lines. Our immediate interest is in finding a formula for the distance between a point and a line; when we have worked out this result, we will be able to turn it into a very useful fact about triangles.

As with triangles, we need to agree on a system for representing points and lines. It's easy to agree that a typical point called **P** can be thought of as a list of its x and y components:

$$\mathbf{P} = \{ x, y \}$$

and it's reasonable to define a line **L** in terms of two distinct points **Pa** and **Pb** that lie on that line:

$$\mathbf{L} = \{ \mathbf{P}_a, \mathbf{P}_b \} = \{ \{x_a, y_a\}, \{x_b, y_b\} \}.$$

Now if we are given any line **L**, it is easy to construct a vector **D** which corresponds to the direction of the line, simply by subtracting the starting point from the endpoint:

$$xD = xb - xa$$

$$yD = yb - ya$$

Perhaps more important is the fact that we can also construct a vector \mathbf{N} which is perpendicular or **normal** to the line:

$$\begin{aligned}xN &= ya - yb \\ yN &= xb - xa\end{aligned}$$

We can now easily verify, by arithmetic, that the two direction vectors are orthogonal, by computing the dot product of the two vectors:

$$D \cdot N = xDxN + yDyN = (xb - xa)(ya - yb) + (yb - ya)(xb - xa) = 0$$

If a vector is normal to a line, then so is any (nonzero) multiple of it. To try to standardize things, let us compute another vector \mathbf{n} , the result of dividing \mathbf{N} by its length. We call the result a *unit normal vector* :

$$\begin{aligned}xn &= \frac{ya - yb}{\sqrt{(ya - yb)^2 + (xb - xa)^2}} \\ yn &= \frac{xb - xa}{\sqrt{(ya - yb)^2 + (xb - xa)^2}}\end{aligned}$$

The unit normal vector \mathbf{n} to a line \mathbf{L} has the very useful property that the signed distance from any point \mathbf{P} to the line \mathbf{L} is a simple vector dot product involving \mathbf{n} and the difference between the point \mathbf{P} and any point on the line, such as \mathbf{Pa} :

$$\text{Signed Distance}(P, L) = n \cdot (P - Pa)$$

Note that the signed distance is *zero* if \mathbf{P} is actually on the line, negative if \mathbf{P} is “to the right” of the line, and positive if \mathbf{P} is “to the left” of the line. Here, being to the right of the line is understood by imagining that we stand on point \mathbf{Pa} and face in the direction of point \mathbf{Pb} . We are looking along the line \mathbf{L} , in the direction specified by its two defining points. Points to our right have negative distance to the line, and points to our left have positive distance.

We have come up with a formula that gives us both a distance and an orientation. We will find that we can do a lot with this idea!

3 Program #1: Point/Line Orientation

Write a program which:

- reads the definition of a line \mathbf{L} ;
- reads the definition of a point \mathbf{P} ;
- computes $D = \text{Signed Distance}(P, L)$;
- prints \mathbf{P} and \mathbf{D} .

To test your program, use the line

$\mathbf{L1} = \{\{1,3\}, \{4,9\}\}$

As test points \mathbf{P} , use the sequence of values $\{3,3\}, \{3,4\}, \{3,5\}, \dots \{3,10\}$.

Repeat your tests for the lines $\mathbf{L2}$ and $\mathbf{L3}$:

$\mathbf{L2} = \{\{4,9\}, \{1,3\}\}$

$\mathbf{L3} = \{\{0,1\}, \{10,21\}\}$

Lines $\mathbf{L1}$, $\mathbf{L2}$ and $\mathbf{L3}$ all “look” like the same line, but $\mathbf{L2}$ is going in the opposite direction, which affects its orientation and hence the value of the signed distance. Do your results agree with this statement?

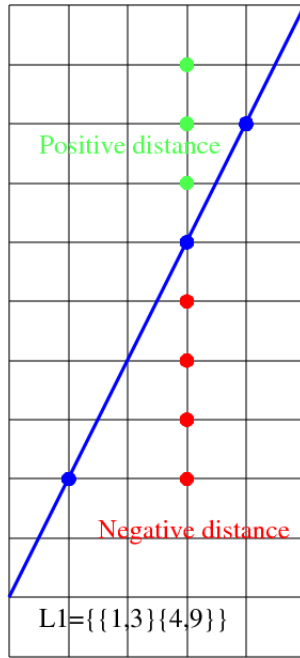


Figure 1: The Point-Line Orientation Test.

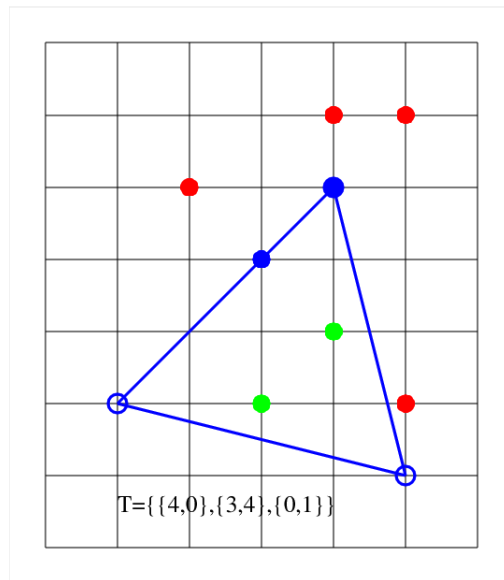


Figure 2: The Point-Triangle Orientation Test for Triangle Tex4.

4 Does a Point Lie Inside a Triangle?

Because the formula for the signed distance from a point to a line is positive exactly when a point is “to the left” of a line, we can use this formula to find out something interesting about triangles.

Suppose we ask whether a point \mathbf{P} is inside or outside a given triangle $\mathbf{T}=\{\mathbf{a},\mathbf{b},\mathbf{c}\}$. To answer the question, imagine walking along the perimeter of the triangle. Suppose you walk from vertex \mathbf{a} to vertex \mathbf{b} , or as we might say, along side \mathbf{Sab} . Suppose that as you do this, you can say the point \mathbf{P} is to the left of the line $\{\mathbf{a},\mathbf{b}\}$. Suppose that as we continue our walk, from \mathbf{b} to \mathbf{c} , and then from \mathbf{c} back to \mathbf{a} , we also see that the point is to the left of the lines $\{\mathbf{b},\mathbf{c}\}$ and $\{\mathbf{c},\mathbf{a}\}$.

Remember that we assume the vertices of our triangle are listed in counterclockwise order. Therefore, if the point is always to our left as we walk around the triangle, it must actually be **inside the triangle**. On the other hand, if it is to the right at any time, then it must lie outside. We can replace the image of walking around the triangle by our formula which measures signed distances. If all the signed distances are positive, the point is inside; if any one is negative, it is outside. If all the signed distances are nonnegative, but one is zero, the point lies along a side of the triangle. If two distances are zero, the point is actually a vertex.

Thus we have discovered a numeric formula which tells us something interesting about a geometric object.

5 Program #2: Point/Triangle Inclusion

Write a program which

- reads the definition of a triangle $\mathbf{T}=\{\mathbf{a},\mathbf{b},\mathbf{c}\}$;
- reads a point \mathbf{P} ;
- computes the signed distances from \mathbf{P} to the lines $\{\mathbf{a},\mathbf{b}\}$, $\{\mathbf{b},\mathbf{c}\}$ and $\{\mathbf{c},\mathbf{a}\}$;
- prints \mathbf{P} , the signed distances, and the word “inside” or “outside”, depending on whether \mathbf{P} is inside \mathbf{T} or not.

Use the triangle $\mathbf{Tex4}$ defined by $\{\{4,0\},\{3,4\},\{0,1\}\}$. Use the points (1,4), (2,1), (2,3), (3,2), (3,4), (3,5), (4,1) and (4,5) to see if your program can detect inside/outside/on. Be prepared to explain your test for when the point is inside the triangle.

What happens when the point is on an edge, or is a vertex?

What happens if you reverse the order of the vertices of $\mathbf{Tex4}$?

6 The Reference Triangle

We will find it convenient to agree on a single standard triangle called the *reference triangle*. The actual definition of this triangle is simply $\mathbf{Tref}=\{\{1,0\},\{0,1\},\{0,0\}\}$.

Consider figure 3, in which we have displayed an image of the reference triangle. This triangle can be defined as the intersection of three half planes:

$$\begin{aligned}0 &\leq x \\ 0 &\leq y \\ x + y &\leq 1\end{aligned}$$

Suppose that we define a new variable $z = 1 - x - y$. While x and y are Cartesian coordinates, z will just be an artificial coordinate for us. However, if we include z , our triangle definition has the nicely symmetric

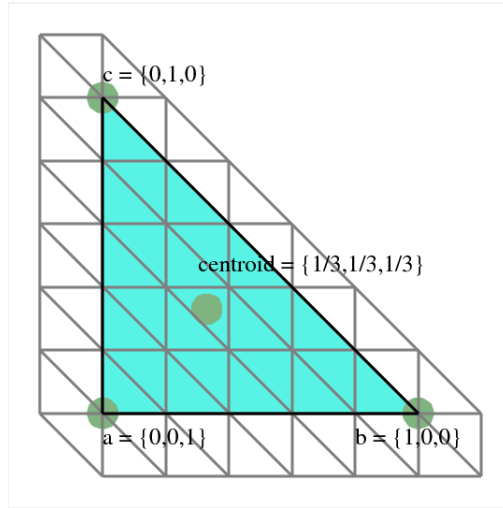


Figure 3: The Reference Triangle with Barycentric Gridlines

form:

$$\begin{aligned} 0 &\leq x \\ 0 &\leq y \\ 0 &\leq z \end{aligned}$$

Moreover, this new variable z means that we now have an interesting coordinate system inside the triangle. The vertical and horizontal gray lines in the figure already mark contour levels for x and y coordinates. Now the diagonal lines do the same thing for z .

Notice that x is 0 on the left boundary, attaining its maximum value at the node a , and that in fact x marks the distance from the left boundary. Of course y works the same way for the lower boundary and the node b . By adding the variable z , we can use the same interpretation for the diagonal boundary and the node c .

We can, if we wish, label points in the triangle with (x, y, z) coordinates. By doing so, we emphasize the symmetries in the triangle. Notice that a vertex has two zero coordinates, points along the boundary have one zero coordinate, and points in the interior have all coordinates strictly positive. The centroid has the coordinates $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Moreover, the definition of the triangle can be rephrased to state that the triangle is exactly the set of points for which all three values of the coordinates (x, y, z) are nonnegative.

This coordinate system is interesting for the reference triangle, but the remarkable thing is that we can essentially use this same coordinate system on *any* triangle, by imagining a linear transformation from the reference triangle to that triangle. A linear transformation maps lines to lines, so the grid lines on our reference triangle will map to corresponding grid lines in a general triangle. Figure 4 shows a large yellow triangle and the reference triangle together, and you can see how the coordinate system is preserved.

7 Barycentric Coordinates

In the reference triangle, it was easy to see what the meanings of the x and y coordinates were, and how z could be usefully added. But in a general triangle, is there a way to define this coordinate system that doesn't rely on us to come up with a mapping from the reference triangle?

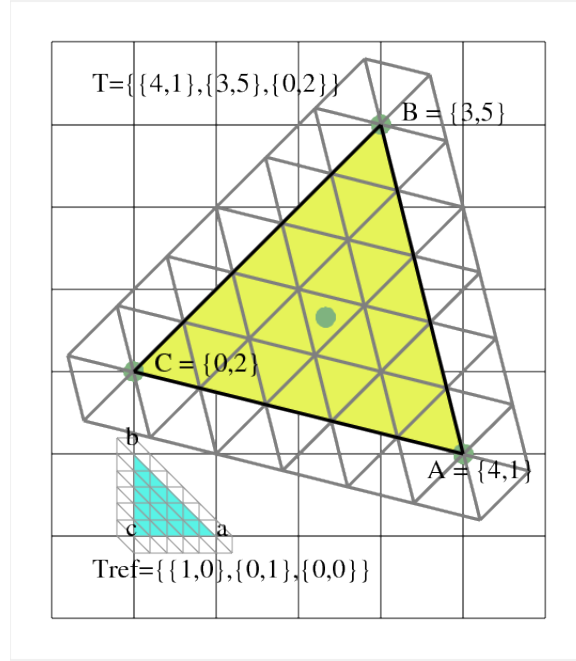


Figure 4: Barycentric Gridlines on a General Triangle

The key to making the handy coordinate system in the reference triangle into a universal coordinate system is to find some general property that remains true for other triangles. The contour lines should be a clue. Instead of thinking about \mathbf{x} as a Cartesian coordinate, think about it as the distance from the vertex \mathbf{a} to the opposite side $\{\mathbf{b},\mathbf{c}\}$. This distance will be 1 at \mathbf{a} and decrease linearly until the side is reached. This is *almost* the right idea, but then if we looked at a triangle that was doubled in size, we'd still want the new \mathbf{a} vertex to have an “ \mathbf{x} ” coordinate of 1.

What would work well would be to use *relative distances* instead of absolute ones. So in measuring the “ \mathbf{x} ” coordinate of a point, we'd compute the distance to side $\{\mathbf{b},\mathbf{c}\}$ as before, but then we'd divide it by the distance of vertex \mathbf{a} to side $\{\mathbf{b},\mathbf{c}\}$. Since \mathbf{a} is the point in the triangle that is as far as possible from $\{\mathbf{b},\mathbf{c}\}$, points inside the triangle will generally have a relative distance between 0 and 1.

To get all three coordinates, of course, which correspond to our (x,y,z) coordinate system in the reference triangle, we'd measure a point's relative distance to the sides opposite each vertex in turn.

Since we're not in the reference triangle anymore, we can't really call this set of coordinates (x,y,z) . Instead, we will use the names (ξ_a, ξ_b, ξ_c) , where the subscript reminds us that the first coordinate, is related to vertex \mathbf{a} . This system of coordinates is known as the *barycentric coordinate system*.

We know how to compute the distance between a point and a line, so we can write down formulas for the barycentric coordinates of a point with respect to a triangle. These are valid whether the point is inside or outside the triangle:

$$\xi_a = \frac{\text{SignedDistance}(P, \{b, c\})}{\text{SignedDistance}(a, \{b, c\})}$$

$$\xi_b = \frac{\text{SignedDistance}(P, \{c, a\})}{\text{SignedDistance}(b, \{c, a\})}$$

$$\xi_c = \frac{\text{SignedDistance}(P, \{a, b\})}{\text{SignedDistance}(c, \{a, b\})}$$

8 Program #3: Barycentric Coordinates

Write a program which:

- reads the definition of a triangle **t**;
- reads a point **P**;
- computes and prints (ξ_a, ξ_b, ξ_c) .

Use the triangle **Tex4**. Test the points (1,4), (2,3), (3,2), (3,4), (3,5) and (4,5). Compare these results with your “inside/outside/on” results. How do the barycentric coordinates tell your point is inside, on an edge, at a vertex, or outside the triangle.

For a triangle **t**, the centroid **t.cent** is the average of the vertices. The centroid of **Tex4** is (7/3,5/3). Use this point as input to your program. What are the barycentric coordinates of this point?