Automated Point Spacing

Curvature Detection

In order to refine an edge grid based on curvature present on a curve, curvature must be defined. A classical way to calculate curvature is to evaluate derivatives on the curve. However, derivative calculations are relatively expensive operations and will not be used here. Instead, only the discrete edge grid will be used to determine the severity of the curvature in a region on the curve. A local measure, *Curvature Ratio*, is on intervals on the edge grid and used to refine intervals where curvature is relatively high. Consider Figure 1.

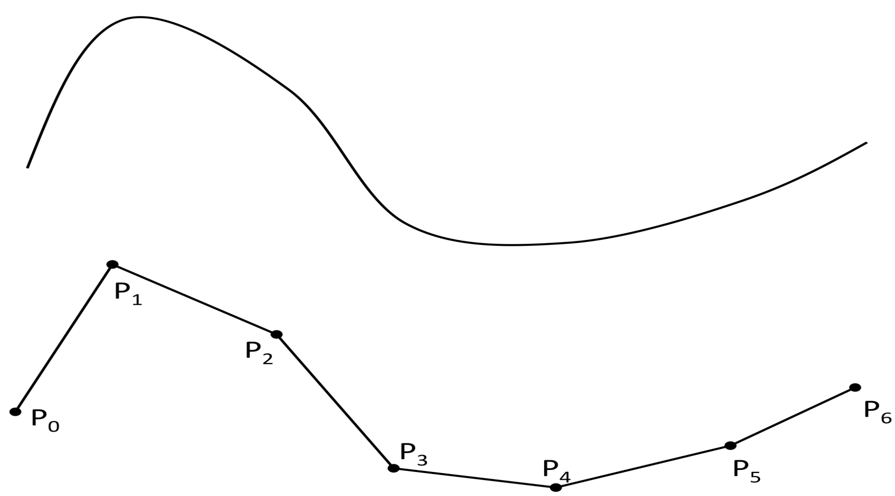


Figure 1 Curve and its corresponding discrete edge grid

In this figure, a curve and its corresponding, discrete edge grid can be seen. This discrete edge grid has been generated with a relatively uniform parameterization, i.e. this edge grid points are nearly evenly distributed along the curve. This discrete edge grid can be a starting point for the refinement process. Each interval on the discrete edge grid (P_0-P_1, P_1-P_2, etc…) is examined and the local measure, *Curvature Ratio*, is defined for each interval. In the figure above, the uniform parameterization is a relatively good approximation of the curve with only seven points. However, below in Figure 2, is a coarser uniform parameterization of the same curve with only four points. This discrete edge grid is not a good approximation of the curve.

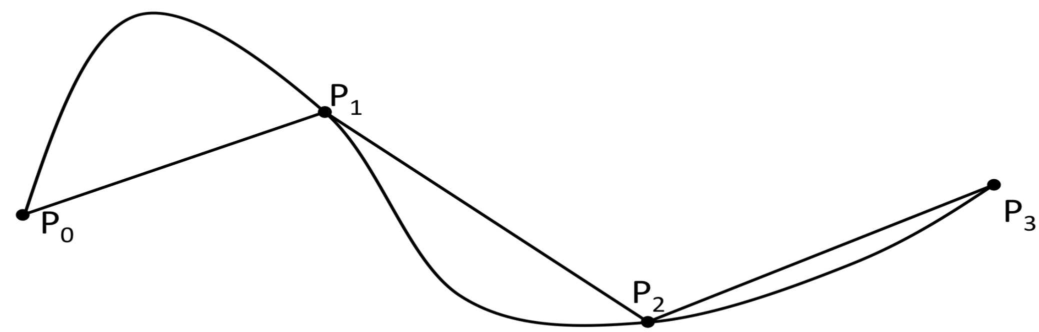


Figure 2 Curve with coarse, uniform parameterization

The process of refining this discrete edge grid would begin by calculating the *Curvature Ratio* for each interval. This is done by first calculating the length, L_i, of the interval, e.g. distance from P_0 to P_1. Then, a candidate point is inserted on the curve at the halfway point in the parameterization between P_0 and P_1. The distance, L_c, between the candidate point and the line connecting the points defining the interval is then calculated. The ratio of L_c to L_i is the *Curvature Ratio*. This can be seen below in Figure 3, which shows a close-up of the interval P_0-P_1, and the candidate point, P_c.

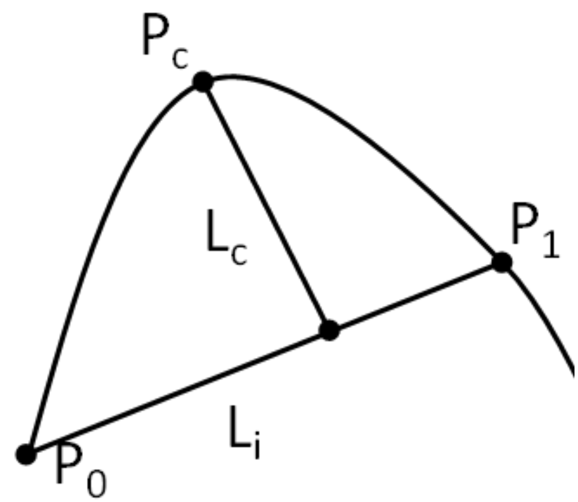


Figure 3 Close-up of interval on discrete edge grid

The ratio of L_c to L_i is compared to the user-defined value of the *Curvature Ratio*. The candidate point, P_c, is kept if the interval's local value of *Curvature Ratio* is larger than the user-defined value. By repeating this calculation for each interval the edge grid is refined until no intervals have a local value of *Curvature Ratio* larger than the user-defined value. This process is visualized in the following figures.

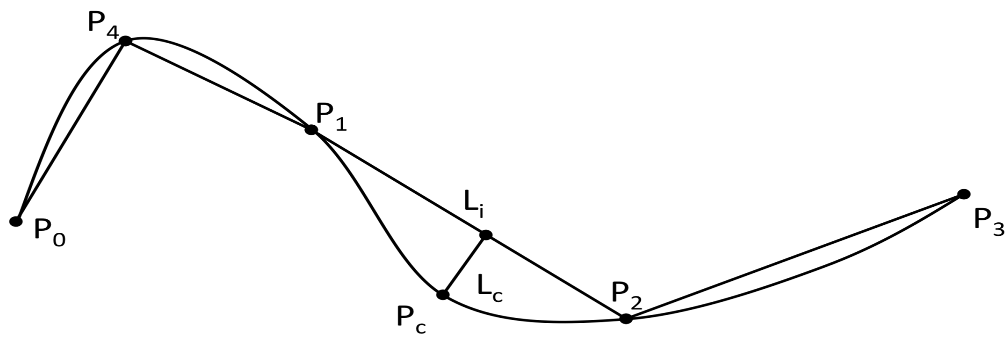


Figure 4 Edge Grid Refinement, Second Interval

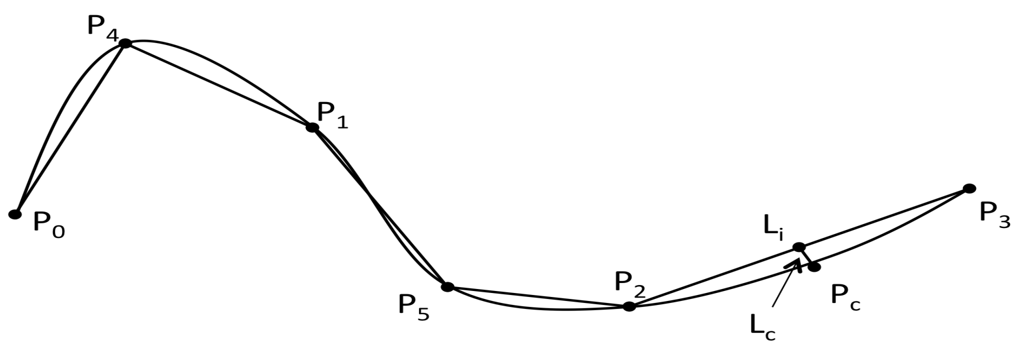


Figure 5 Edge Grid Refinement, Third Interval

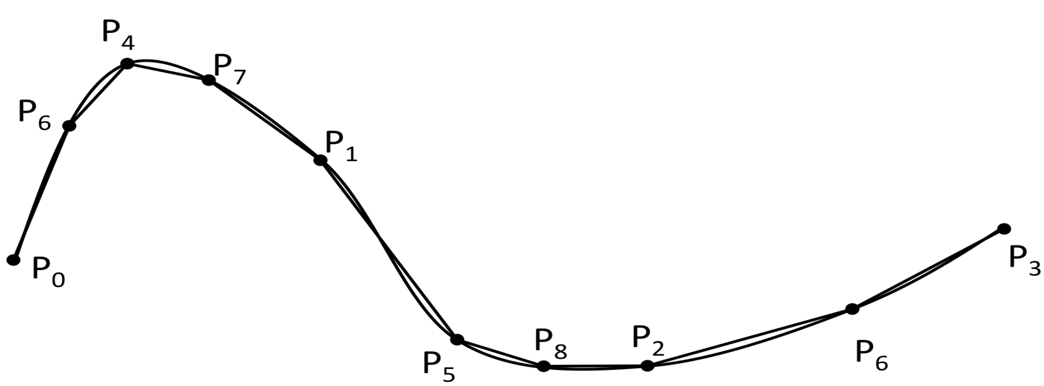


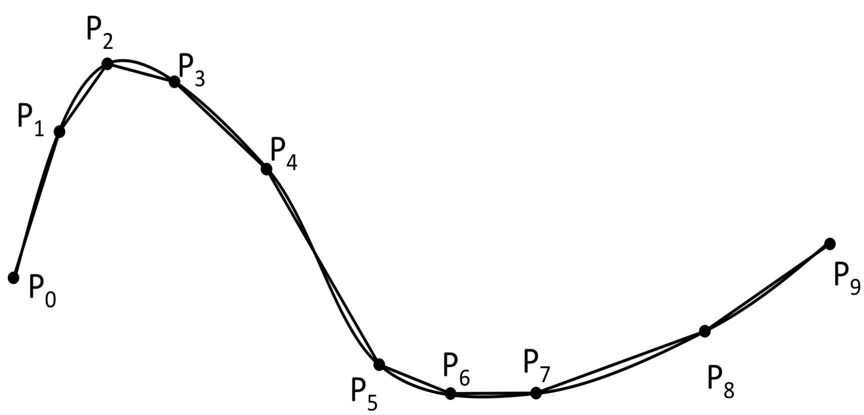
Figure 6 Edge Grid Refinement, Finished

In the figures above, Figure 4, Figure 5, and Figure 6, the iterative process of edge grid refinement by using a local measure, *Curvature Ratio*, can be seen. The edge grid that is generated through this process is known as the *ideal edge-grid*. This process of edge refinement is not optimal point placement, which would be expensive; it is a locally-justified edge-refinement scheme that is both robust and computationally cheap. The use of a ratio instead of a global value means that this edge grid refinement process can be used on grids where the geometry has largely varying scale without the need for adjusting the user-defined value for *Curvature Ratio*.

Automatic Edge-Grid Refinement

Now that the local measure, *Curvature Ratio*, has been defined and the process of generating the *ideal edge-grid* has been detailed, a brief explanation of how this fits in with edge grid generation in SolidMesh will follow. In SolidMesh, edge grids are generated using point-spacing values on the end points defining a curve. With those values set, the edge grid generation is a one-dimensional, grid generation process. Point-spacing sources can be added to the interior of a curve to locally refine the edge grid at that location. These two options, end-point spacing and interior point-spacing sources, are both used in order to automatically arrive at an edge grid that is based on curvature.

First, the *ideal edge-grid* is calculated using the user-defined *Curvature Ratio*. The point-spacing values present at the end-points of the *ideal edge-grid*, Figure 7 (top), are used as the end-point spacing values for the curve's edge grid. They are determined by calculating the length of the first interval attached to the end points of the discrete edge grid. With these values, an *initial edge-grid* is then generated, Figure 7(bottom).



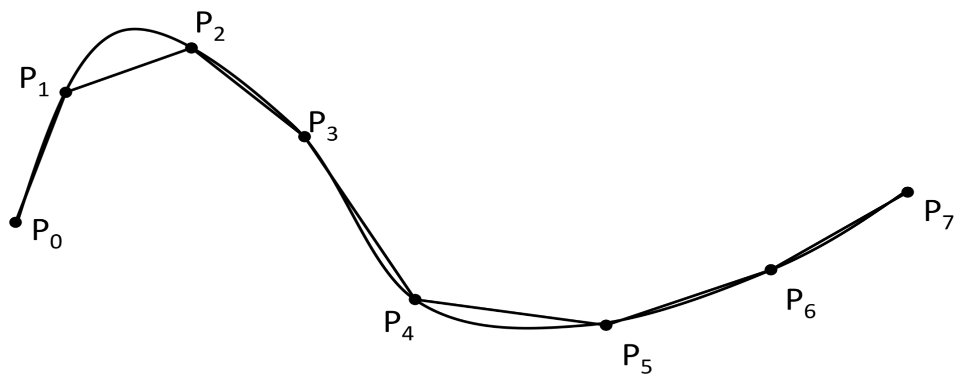


Figure 7 Ideal Grid (top), Initial Grid (bottom)

In this example, the end point spacing at P_0 is too large to capture the curvature between P_1 and P_2 in the *initial edge-grid*. In order to more accurately represent that curvature, the edge grid should be refined. In order to do this, a second process is performed that compares the point spacing values present in the *ideal edge-grid* to the ones present in the *initial edge-grid*. Each point in the *initial edge-grid* has two attached edges. The smaller of the two attached edges is taken to be the point-spacing value at that point (*current point*). Next, the location of the *current point* in the *initial edge-grid* is found on the *ideal edge-grid* by determining which interval (*containing-interval)* the point is located. The length of *containing-interval* is then compared to the point-spacing value of the *current point*. The ratio, *Deviation Factor*, of the point-spacing value of the *current point* and the length of the *containing-interval* is then calculated. The *Deviation Factor* is a user-defined value and if the value of the local *Deviation Factor* exceeds the user-defined value, a point-spacing source is added at the *current node* with the point-spacing value equal to the length of *containing-interval*. The edge-grid would be recomputed at this point with the edge-grid being refined at *P1*. This effect can be seen in Figure 8.

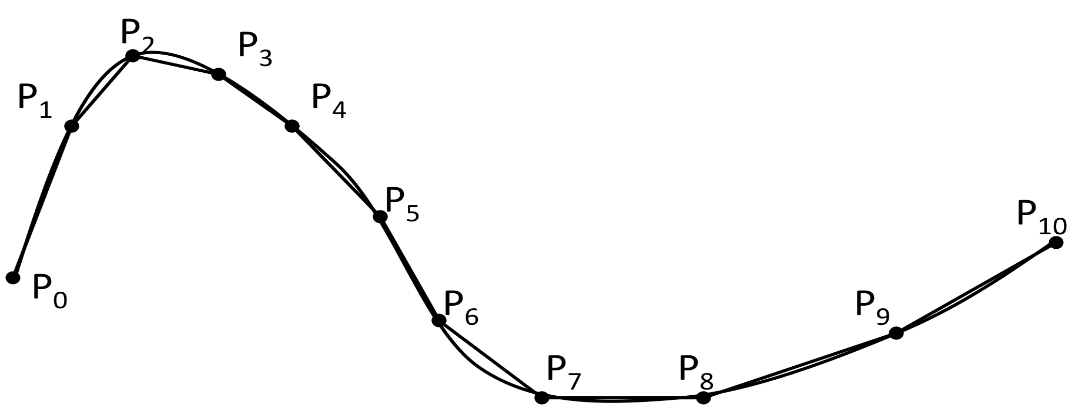


Figure 8 Edge Grid After Point-Spacing Source Insertion

The point-spacing value of the curve has been altered at P_1 and has subsequently altered the entire edge-grid. The new edge-grid will be called the *evolving edge*-grid. The process of comparing the point-spacing values in the *evolving edge-grid* to the values present in the *ideal edge-grid* is repeated until no points on the *evolving edge-grid* have point-spacing values that deviate from the *ideal edge-grid* by a factor of *Deviation Factor*. In the above example, the point-spacing value at P_7 is larger than is present in the *ideal edge-grid* and could therefore be refined—depending on the user-defined value of *Deviation Factor*.

The two ways of altering an edge grid, changing the end-point spacing values and inserting point-spacing sources on the interior of a curve, are both used to arrive at an edge grid that is based on curvature. The *Curvature Ratio* is used to calculate the *ideal edge-grid*. The *ideal edge-grid* is then used to set the end-point point-spacing values on the curve. An *initial edge-grid* is then calculated and refined by comparing the ratio of point spacing values in the *initial edge-grid* to the ones present in the *ideal edge-grid*. The *initial edge-grid* is then refined where the ratio exceeds the user-defined *Deviation Factor* to arrive at a *final edge-grid*. The combination of *Curvature Ratio* and *Deviation Factor* gives the user the option of deciding how close to the *ideal edge-grid* they want the *final edge-grid* to be. For example, if the user did not want any point-spacing sources inserted into any of the edge grids, then setting the *Deviation Factor* to a large number would accomplish this. The edge grids generated from this choice would be generated only from the point-spacing values on the end-points of the curves. This algorithm is detailed below.

1. Loop over edge grids:
    a. Create Uniform Parameterized Edge Grid, *uniform edge-grid*
    b. Define *uniform edge-grid as first edge-grid*
    c. Use *Curvature Ratio* to locally refine *first edge-grid* into *ideal edge-grid,*
        i. Loop over intervals in *first edge-grid*
            1. Insert candidate point
            2. Construct Local *Curvature Ratio* and compare to user-defined *Curvature Rato*
                a. Keep candidate point if Local *Curvature Ratio* violates user-defined *Curvature Ratio*
                b. Reject candidate point if Local *Curvature Ratio* does not violate user-defined *Curvature Ratio*
        ii. Repeat (c.i.) if any points have been inserted
    d. Define locally-refined *first edge-grid as ideal edge-grid*
    e. Use end-point point-spacing values from *ideal edge-grid* to generate *initial edge-grid*
    f. Define *evolving edge-grid* as *initial edge-grid*
    g. Loop over intervals in *evolving edge-grid*

i. Find interval with largest *Deviation Factor*

1. Compare Local *Deviation Factor* to user-defined *Deviation Factor*

a. Insert point-spacing source that matches point-spacing value on the *ideal edge-grid* for that interval

ii. Repeat (g.i.) if any point-spacing sources have been added

Two further user-defined values are used as bounds to this process of edge-grid refinement. These are *Minimum Point-Spacing* and *Maximum Point-Spacing* values. These values are global and are used for each edge grid. No end-point point-spacing value will be set at a value outside of the range specified by the *Minimum Point-Spacing* and *Maximum Point-Spacing* values. Also, no point-spacing sources will be inserted with values that are outside of the user-specified range. A total of four user-defined values, *Curvature Ratio*, *Deviation Factor*, *Minimum Point-Spacing*, and *Maximum Point-Spacing*, give the user a large amount of control over a highly automated edge grid generation process.

User Interface

Below is the control panel for the Automated Point Spacing Tool. The four fields alter four fundamental parts of the point spacing algorithm. The "Min Spacing" and "Max Spacing" buttons control the global minimum and maximum point spacing. For the example shown below, no point spacing will be below the minimum point spacing, 0.1 and no point spacing will be above the maximum point spacing, 10. The "Curvature Ratio" field alters how many points get placed in areas of high curvature. To increase resolution in highly curved regions, lower the value. To decrease resolution in highly curved regions, increase the value. This value is a ratio that is calculated for each curve and is therefore a local tolerance but the value in the field will be used globally. The "Max Deviation" button alters the number of resolution sources that get added on the interior of a curve. To add more points sources on a curve, lower the value. To add more point sources on a curve, decrease the value. This field gives the user the ability to alter the *interior* point spacing applied to curves instead of limiting the changes of resolution to the end points.

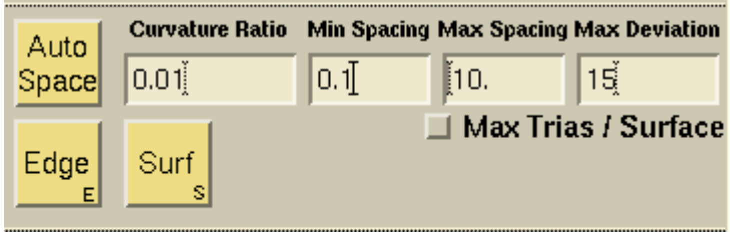


Figure 9 Panel for Automated Point Spacing Tool