Convex Split-Face Polygon Cuts

January 8, 2022

Tools

For more compact, finite summations:

$$S(x_1, ..., x_N) = \sum_{k=1}^{N} x_k \tag{1}$$

Introduction

There is a formula which counts the number of ways you can cut a convex polygon between all its faces. Given a convex polygon with N faces, this is

$$C = S(N-1, ..., 0) = \frac{N(N-1)}{2} \tag{2}$$

This captures the idea that from one face, you can cut to $N-1$ other faces and from the next you can cut to $N-2$ other faces while being sure not to double count.

What if the polygon's faces were further divided into some number of co-linear segments? How can we count the number of ways you can cut it?

Equations

Let there be a convex polygon with N faces. Each face numbered by k with $1 \leq k \leq N$ and each divided into x_k segments. Let $C(x_1, ..., x_N)$ be the number of cuts. For the simple case where each x is equal to 1, it's the familiar formula

$$C(1, ..., 1) = S(N-1, ..., 0) = \frac{N(N-1)}{2} \tag{3}$$

In the general case, the count function satisfies a couple relations. First, since the polygon in question has no set orientation and may be rotated, the arguments to the count function may also be rotated

$$C(x_1, ..., x_k) = C(x_2, ..., x_k, x_1) \tag{4}$$

Second, extending the cut process described for the simple case in the introduction, given a known count, dividing a face produces new cuts from each other faces' segments

$$C((x_1+1), ..., x_k) = S(x_2, ..., x_N) + C(x_1, ..., x_N) \tag{5}$$

This can be applied in a reductive manner by subtracting 1 from x_1 and repeatedly applying the rule until the first parameter to the remaining application of C is reduced to 1

$$\begin{aligned} C(x_1, ..., x_k) &= S(x_2, ..., x_N) + C((x_1-1), ..., x_N) \\ &= 2S(x_2, ..., x_N) + C((x_1-2), ..., x_N) \\ &= (x_1-1)S(x_2, ..., x_N) + C(1, x_2, ..., x_N) \end{aligned} \tag{6}$$

Using the two rules together, each parameter to C can be reduced to 1 and a fully reduced form can be derived

$$\begin{aligned} C(x_1, ..., x_k) &= (x_1-1)S(x_2, ..., x_N) \\ &+ (x_2-1)S(1, x_3, ..., x_N) \\ &+ C(1, 1, x_3, ..., x_N) \\ &= (x_1-1)S(x_2, ..., x_N) \\ &+ (x_2-1)S(1, x_3, ..., x_N) \\ &+ (x_3-1)S(1, 1, x_4, ..., x_N) \\ &+ C(1, 1, 1, x_4, ..., x_N) \\ &= \sum_{k=1}^{N}(x_k-1)S(1, ..., 1, x_{k+1}, ...x_n) + C(1, .., 1) \\ &= \sum_{k=1}^{N}(x_k-1)\left[(k-1) + S(x_{k+1}, ...x_n)\right] + C(1, .., 1) \end{aligned} \tag{7}$$

This derivation process is useful when calculating these counts on paper. The abbreviations can be expanded for the final result

$$C(x_1, ..., x_k) = \sum_{k=1}^{N}(x_k-1)\left[(k-1) + \sum_{j=k+1}^{N} x_j\right] + \frac{N(N-1)}{2} \tag{8}$$

Computationally, the nested summations do not necessarily mean the calculation will take $O(N^2)$ steps since the inner summation can be pre-calculated. Then, for each k, reduce the calculated value by x_k.