GENERATING UNIFORM RANDOM POINTS

IN A REGULAR n SIDED POLYGON.

MYRON HLYNKA AND DEBORAH LOACH

Dept. of Mathematics and Statistics,

University of Windsor, Windsor, Ontario, Canada N9B 3P4

ABSTRACT. Consider a regular polygon with n sides. We present a method to generate points uniformly distributed inside the polygon without using inclusion/exclusion.

Keywords. polygon, uniform random generator

Acknowledgment. This work was supported by grants from NSERC (Natural Science and Engineering Research Council of Canada).

1. INTRODUCTION:

There are many reasons for wanting to generate random points inside a polygon. The most common polygon in which random points are generated is the square. This is especially easy and can be used to study various properties of random points in the square. For example, we might want to know the distribution of the distance between two randomly chosen points in a square in order to approximate how long it might take a mobile helicopter ambulance in a random position to respond to an emergency in a different random position or to find the expected connection length between two random points on a computer chip.

Typeset by $\mathcal{AMS}$-TEX

As $n \to \infty$, a regular n-gon approaches a circle. It is easy to generate uniform points in a circle using inclusion/exclusion methods. It is also possible to generate uniform random points inside a circle directly without using inclusion/exclusion. One obvious (but incorrect) method to generate uniform random values inside a circle of radius 1 is to generate polar coordinates (r, θ) where r is a value from a uniform distribution on $(0, 1)$ and θ is a random value from a uniform distribution on $(0, 2\pi)$. The correct method gives higher weights to the values of r which are farther from the center. One should generate a random value θ from a uniform distribution on $(0, 2\pi)$ and generate a value $r = \sqrt{u}$ where u is a random value r from a uniform distribution on $(0, 1)$ (McCrae, 2005).

Excluding the square, it is not so easy to generate uniform random points inside a regular polygon such as a triangle, a pentagon, a hexagon. It is certainly possible using inclusion/exclusion but this method could involve some (perhaps minor) complications in programming. Our method, although it involves some complex expressions, is easy to implement.

2. METHODOLOGY:

Consider a regular polygon with n sides (a regular n-gon). Orient the n-gon so that one of the vertices is situated on the positive X axis at $(a, 0)$ and so that the center of the circle that circumscribes the polygon is centered at (0,0). This is always possible.

An n-gon can be divided up into n congruent isosceles triangles by drawing lines from (0,0) to the vertices of the n-gon. See Figure 1. Consider the triangle with vertices O, A, B where $O = (0, 0)$, $A = (a, 0)$ and B is the first vertex on the polygon which is encountered when moving counterclockwise from the vertex A. Let the sides of the triangle opposite O, A, B be o, a, b. The angle AOB has $2\pi/n$ radians. The angle OAB has $\dfrac{(n-2)\pi}{2n}$ radians. Let θ be an angle at O measured counterclockwise from the line segment AO, with $0 \leq \theta \leq 2\pi/n$. The line from O with angle θ will intersect AB at some point D whose distance from O is denoted by $R(\theta)$. See Figure 2. For the triangle DAO, we apply the Law of Sines. Angle AOD has θ radians. Angle OAD has $\frac{(n-2)\pi}{2n}$ radians. Hence angle ODA has $\pi - \theta - \frac{(n-2)\pi}{2n}$ radians. Apply the Law of Sines to get

$$\frac{\sin\left(\pi - \theta - \frac{(n-2)\pi}{2n}\right)}{a} = \frac{\sin\left(\frac{(n-2)\pi}{2n}\right)}{R(\theta)}.$$

Then

$$R(\theta) = \frac{a\sin\left(\frac{(n-2)\pi}{2n}\right)}{\sin\left(\pi - \theta - \frac{(n-2)\pi}{2n}\right)} = \frac{a\sin\left(\frac{(n-2)\pi}{2n}\right)}{\sin\left(\theta + \frac{(n-2)\pi}{2n}\right)}.$$

We intend to use the inverse c.d.f. method to generate random values (Ross, 2002). First we adjust $R(\theta)$ so that $\frac{R(\theta)}{k}$ is a probability density function on $0 \le \theta \le \frac{(n-2)\pi}{2n}$. Here k must be

$$\begin{aligned} k &= \int_0^{2\pi/n} R(\theta)d(\theta) = \int_0^{2\pi/n} \frac{a\sin(\frac{(n-2)\pi}{2n})}{\sin(\theta + \frac{(n-2)\pi}{2n})} d(\theta) \\ &= a\sin(\frac{(n-2)\pi}{2n}) ln(|\csc(\theta + \frac{(n-2)\pi}{2n}) - \cot(\theta + \frac{(n-2)\pi}{2n})|)|_0^{2\pi/n} \end{aligned}$$

Next observe that

$$\begin{aligned} \csc x - \cot x &= \frac{1}{\sin x} - \frac{\cos x}{\sin x} = \frac{1-\cos x}{\sin x} \\ &= \frac{1-\cos\left(2(\frac{x}{2})\right)}{2}\frac{2}{\sin x} = \sin^2(\frac{x}{2})\frac{2}{\sin x} \\ &= \sin^2(\frac{x}{2})\frac{2}{2\sin(\frac{x}{2})\cos(\frac{x}{2})} \\ &= \frac{\sin(\frac{x}{2})}{\cos(\frac{x}{2})} = \tan(\frac{x}{2}) \end{aligned}$$

This simplificiation is indeed fortunate, since if such a simplication did not exist, we might not be able to find a closed expression that we can use when applying the inverse cdf method to generate random values.

Thus, $k = a\sin\left(\frac{(n-2)\pi}{2n}\right) ln|\tan\left(\frac{\theta+\frac{(n-2)\pi}{2n}}{2}\right)||_0^{2\pi/n}$.

If $0 \leq \theta \leq \frac{2\pi}{n}$, then $\frac{(n-2)\pi}{2n} \leq \theta + \frac{(n-2)\pi}{2n} \leq \frac{2\pi}{n} + \frac{(n-2)\pi}{2n}$ and

$$\frac{(n-2)\pi}{4n} \leq \frac{\theta+\frac{(n-2)\pi}{2n}}{2} \leq \frac{\pi}{n} + \frac{(n-2)}{n}\frac{\pi}{4} \leq \frac{\pi}{3} + \frac{1}{3}\frac{\pi}{4} < \frac{\pi}{2}$$

for $n = 3, 4, 5, \ldots$

Thus, for our region of interest, we can ignore the absolute value, and we obtain

$$ln|\tan\left(\frac{\theta+\frac{(n-2)\pi}{2n}}{2}\right)| = ln(tan\left(\frac{\theta+\frac{(n-2)\pi}{2n}}{2}\right)).$$

So

$$\begin{aligned} k &= a\sin\left(\frac{(n-2)\pi}{2n}\right)\ln(\tan\left(\frac{\frac{2\pi}{n}+\frac{(n-2)\pi}{2n}}{2}\right)) - \ln(\tan\left(\frac{\frac{(n-2)\pi}{2n}}{2}\right)) \\ &= a\sin\left(\frac{(n-2)\pi}{2n}\right)\ln(\tan\left(\frac{(n+2)\pi}{4n}\right)) - \ln(\tan\left(\frac{(n-2)\pi}{4n}\right)). \end{aligned}$$

This makes

$$f(\theta) = \frac{1}{k}R(\theta) \text{ for } 0 \leq \theta \leq \frac{2\pi}{n}$$

a pdf. Next we compute the cdf $F(\theta)$.

$$\begin{aligned} F(\theta) &= \int_0^\theta f(\tau)d\tau = \frac{1}{k}\int_0^\theta R(\tau)d\tau \\ &= \frac{1}{k}a\sin\left(\frac{(n-2)\pi}{2n}\right)\ln(\tan\left(\frac{\tau+\frac{(n-2)\pi}{2n}}{2}\right))|_{\tau=0}^{\tau=\theta} \\ &= \frac{1}{k}a\sin\left(\frac{(n-2)\pi}{2n}\right)(\ln(\tan\left(\frac{\theta+\frac{(n-2)\pi}{2n}}{2}\right)) - \ln(\tan\left(\frac{(n-2)\pi}{4n}\right))) \end{aligned}$$

To simulate θ, we solve, $u_1 = F(\theta)$ for θ, where $u_1 \equiv$ Unif(0,1). Therefore,

$$\frac{ku_1}{a\sin\left(\frac{(n-2)\pi}{2n}\right)} + \ln(\tan\left(\frac{(n-2)\pi}{4n}\right)) = \ln(\tan\left(\frac{\theta + \frac{(n-2)\pi}{2n}}{2}\right)).$$

Hence

$$\theta = \frac{-(n-2)\pi}{2n} + 2\arctan\left(\exp\frac{ku_1}{a\sin\left(\frac{(n-2)\pi}{2n}\right)} + \ln(\tan\left(\frac{(n-2)\pi}{4n}\right))\right).$$

This generates the θ part of the random point (r, θ).

Given the value of θ, we next generate the value r. First we compute $R(\theta)$ for the given value of θ. Then we generate the value of r in the same manner that is used to generate uniform random values in the interior of a circle. In other words, we give more weight to values that are farther away from the origin. Since the maximum possible value of r for the given value is $R(\theta)$, we adjust the values of r accordingly. Thus we get $r = R(\theta)\sqrt{u_2}$, where $u_2 \equiv$ is Unif(0,1). The pair of values thus determined gives (r, θ), where $0 \leq \theta \leq 2\pi/n$.

To generate values inside the the entire n-gon, we generate a third random value u_3 which is Unif(0,1). Then letting $[x]$ be the greatest integer function of x, we find $u = [nu_3]$ is uniform in $\{0, 1, ..., n-1\}$. Then, $(r, \theta + u\frac{2\pi}{n})$ gives polar coordinates of a uniform random point over the entire n-gon.

3. CONCLUSION:

We have given a method to generate points inside a regular polygon. The method allows for easy programming and only requires a good uniform(0,1) generator. The method is not particularly efficient since it requires three uniform(0,1) values to generate a single two dimensional point. The method could be modified (by defining a c.d.f. over all the whole interval $(0, 2\pi)$) to convert every pair of uniform(0,1) values into a point inside the polygon, but only at the cost of addition effort in programming. But no inclusion/exclusion is required, which would require effort more in programming. The method might be adjusted to non-regular polygons but only with extra work, which may not be warranted.

REFERENCES:

1. A. McCrae website. 2005.

http://home.wlu.edu/ mcraea/GeometricProbabilityFolder/ApplicationsConvexSets/

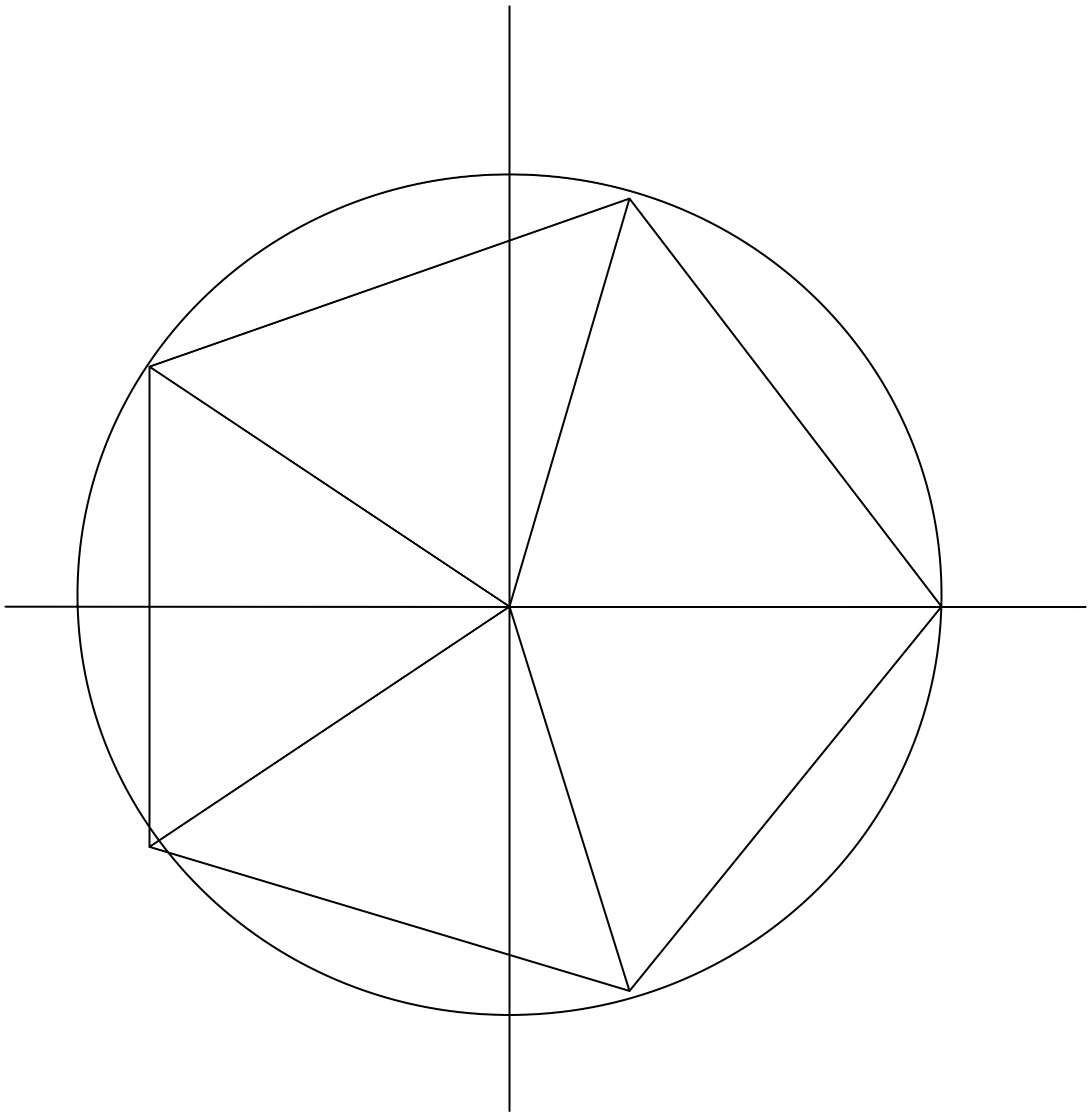
Problem16/ Problem16.html

2. H. Perros. 2003. Computer Simulation Techniques: The definitive introduction! by Harry Perros, Computer Science Department, NC State University Raleigh, NC. 2003.

http://www.csc.ncsu.edu/faculty/perros//simulation.pdf

3. S. Ross, Simulation. Academic Press. 2002.

Figure 1. Regular Pentagon



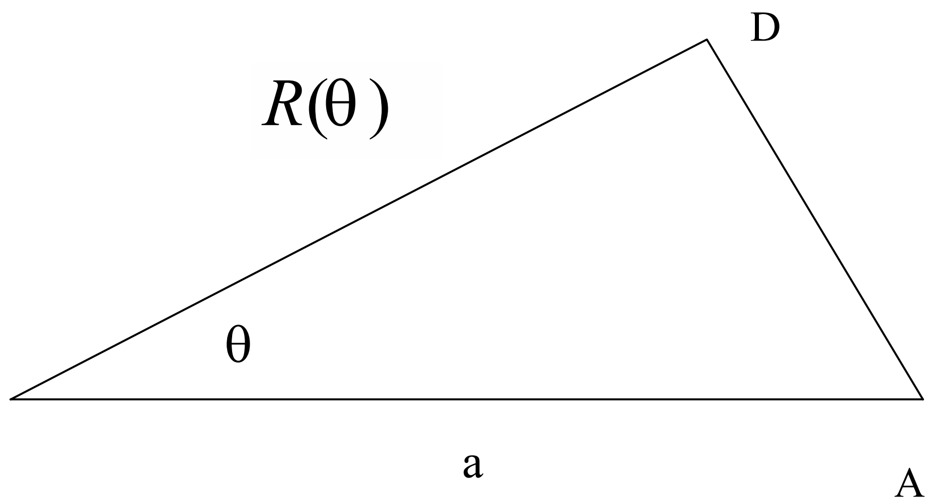


Figure 2: Triangular Section