A General Algorithm For Testing Polygonal Candidates

Niklas Lindblad

April 3, 2007

Abstract

This paper describes a general formula for reversing polygonal numbers and how it can be applied to test any integer for s-gonal properties.

1 The Definition of a Polygonal Number

Mathematically the common formula for the polygonal number $P_{s,n}$ where s is the number of sides in the polygon and n declares that it is the n^{th} s-gonal number can be written as:

$$P_{s,n} = \frac{(s-2)n^2-(s-4)n}{2}$$

2 Rewriting as an Equation

In order to form a reverse function that will provide n for every $P_{s,n}$ the goal is to first rewrite the general formula for polygonal numbers as an equation:

$$P_{s,n} = \frac{(s-2)n^2-(s-4)n}{2} \rightarrow$$

$$2P_{s,n} = (s-2)n^2 - (s-4)n \rightarrow$$

$$(s-2)n^2 - (s-4)n - 2P_{s,n} = 0 \rightarrow$$

$$(s-2)(n^2 - \frac{(s-4)}{(s-2)}n - \frac{2}{(s-2)}P_{s,n}) = 0$$

By solving the above second degree equation using the general solving formula the solution is:

$$n = \frac{\frac{(s-4)}{(s-2}}{2} \pm \sqrt{\frac{\frac{(s-4)}{(s-2)}}{2}^2 + \frac{2}{(s-2)}P_{s,n}}$$

Due to the nature of series only the solution n_1 is usable since a position within a series cannot be negative:

$$n_1 = \frac{\frac{(s-4)}{(s-2}}{2} + \sqrt{\frac{\frac{(s-4)}{(s-2)}}{2}^2 + \frac{2}{(s-2)}P_{s,n}} \rightarrow$$

Furthermore simplified:

$$n_1 = \frac{s-4}{2s-4} + \sqrt{\frac{(s-4)^2}{(2s-4)^2} + \frac{2P_{s,n}}{s-2}}$$

3 Conditional Testing

By calling the function $r(P_{s,n})$ a reverse function of $P_{s,n}$ the function will give n for the s-gonal number $P_{s,n}$ if:

$n \in Z$ and $n > 0$

If the above conditions are met then the following condition will be true:

$P_{s,n} = P_{s,r(P_{s,n})}$

3.1 Source Code Example

The following functions (written in the programing language Python) will take advantage of the previously mentioned reverse function.

3.1.1 The Reverse Function

This is simply a source code version of $r(P_{s,n})$ (truncated to preserve space):

```
def r(s,p):
        n = (((s-4.0)/(s-2.0))/2)\
        +sqrt((((s-4.0)/(s-2.0))/2)**2\
        +(2.0/(s-2.0))*p)
        return n
```

3.1.2 Conditional Function

The following function will check whether the provided number $P_{s,n}$ really is a s-gonal number or not:

```
def is_sgonal(s,p):
        n = int(((s-4.0)/(2.0*s-4.0))\
        +sqrt(((s-4.0)**2/(2*s-4.0)**2)\
        +((2.0*p)/(s-2.0))))
        q = ((s-2.0)*n**2-(s-4)*n)/2
        return p == q
```