The Composites are a Hyperbolic Fractal

Robert M. Akscyn

The University of Waikato

Faculty of Computing & Mathematical Sciences

rakscyn@gmail.com

8 June 2012

An exact, massively-parallelizable formulation for the prime counting function is

$$\pi(n) = n - 1 - \overbrace{\sum_{2 \le p \le \sqrt{n}} R(n/p, p)}^{Composites} \tag{1}$$

where $R(h,p)$ counts p-Rough integers not greater than h (i.e., integers whose least prime is not less than p) for which we have the induction-provable recurrence relation, with base case $R(h,2) = h - 1$,

$$R(h, p_i) = R(h, p_{i-1}) - R(h/p_{i-1}, p_{i-1}) - \underbrace{R(p_i - 1, p_{i-1})}_{=1} \tag{2}$$

for which inverting the recursive descent of the recurrence into an iterative ascent gives

$$R(h,p) = \begin{cases} 0, & \text{if } h < p \\ \underbrace{(h-1)}_{Base\ \ Case} - \sum_{2 \le q < p} \left(R(\frac{h}{q}, q) + 1 \right), & \text{otherwise} \end{cases} \tag{3}$$

illustrating the composites are a hyperbolic fractal.

In Java terms, computing the count of composites (to date replicating known exact values for all orders of magnitude to 10^{14}) consists primarily of

```
for (int i=1; P[i] <= Math.sqrt(n); i++) Composites += R(n/P[i],i);
```

where $P[i]$ is the ith prime, and the iterated function system for $R(h,p)$ in (3) above is:

```
public static long R(long h, int a) {
   if (h < P[a]) return 0;
   long R = h-1;
   for (int b=1; b<a; b++) R -= R(h/P[b],b) + 1;
   return R;
}
```