Numerical evaluation of the strongly triple-free set constant

Julien Cassaigne
IML Marseille
Julien.Cassaigne@iml.univ-mrs.fr

Paul Zimmermann
INRIA-Lorraine
Paul.Zimmermann@inria.fr

November 27, 1996

The strongly triple-free set constant is defined as follows:

$$C = \frac{1}{3}\sum_{k=1}^{\infty} p_{v_k}\left(\frac{1}{v_k} - \frac{1}{v_{k+1}}\right)$$

where v_k is the kth smallest element of the set $S = \{2^i 3^j\}$, i.e. $v_1 = 1$, $v_2 = 2$, $v_3 = 3$, $v_4 = 4$, $v_5 = 6$, $v_6 = 8$, $v_7 = 9$, $v_8 = 12$, $v_9 = 16$, $v_{10} = 18$, ... and p_n is the size of a largest subset of $S_n = \{k \in S, k \leq n\}$ such that $x \in S_n$ implies $2x$ is not in S_n and $3x$ is not in S_n. A graphical interpretation of p_n is as follows: put all the elements of S_n in adjacent columns, where column j has numbers of the form $2^i 3^j$. Then p_n is the size of the maximal independent set of non-adjacent vertices. Here is for $n = 16$ a maximal independent set consisting of $1, 4, 16, 6, 9$ (marked with x), whence $p_{16} = 5$.

```
x - 3 - x
|   |
2 - x
|   |
x - 12
|
8
|
x
```

We notice in the above graph that we marked all vertices of even parity (i.e. numbers $2^i 3^j$ such that $i + j$ is even). As we shall see below, we always obtain a maximal independent set in such a way, or by taking all vertices of odd parity:

Proposition 1 *Among the set of all vertices with $i + j$ even and the set of all vertices with $i + j$ odd, at least one is a maximal independent set.*

If we assume this result, then computing p_n is easy (here in MuPAD):

```
peven:=proc(n) local k; begin
  _plus(iquo(1+floor(ln(n/3^k)/ln(2))+((k+1) mod 2),2)$k=0..floor(ln(n)/ln(3)))
end_proc:

podd:=proc(n) local k; begin
  _plus(iquo(1+floor(ln(n/3^k)/ln(2))+(k mod 2),2)$k=0..floor(ln(n)/ln(3)))
end_proc:

p:=proc(n) begin max(podd(n),peven(n)) end_proc:

>> p(n)$n=1..25;

 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6
```

Then we compute v_k with the following code:

```
V:=[1]: pow3:=3:
v:=proc(n) option remember; local m; begin
   while nops(V)<n do
         m:=V[nops(V)]; # always a power of two #
         V:={op(V. map(V,_mult,2))};
         if pow3<=2*m then V:=V union {pow3}; pow3:=3*pow3 end_if;
         V:=sort([op(V)]);
   end_while;
   V[n]
end_proc:

>> v(k) $ k=1..20;

  1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 27, 32, 36, 48, 54, 64, 72, 81, 96
```

and to approximate C:

```
C:=proc(K) local k; begin
   _plus(p(v(k))*(1/v(k)-1/v(k+1))$k=1..K)/3
end_proc:

>> C(1000);

    552918015883794551135802950003311/901288191458500983405233137778688

>> C(2000);

1663831992128745608787594598713464378541141893 9/27121419161564558068894103\

990026016011617042432

>> C(4000);

5980061742728781798827917068519650967498558731778021207993546719869/974784\

484875042630024324409376616965930922797825482957889840401 6128
```

Error analysis. The error we make by truncating at index K is bounded by:

$$
\begin{aligned}
E_K &= \sum_{k=K+1}^{\infty} p_{v_k}\left(\frac{1}{v_k} - \frac{1}{v_{k+1}}\right) \\
&= \frac{p_{v_K}}{v_{K+1}} + \sum_{k=K+1}^{\infty} (p_{v_k} - p_{v_{k-1}})\frac{1}{v_k} \\
&\leq \frac{p_{v_K}}{v_{K+1}} + \sum_{k=K+1}^{\infty} \frac{1}{v_k}
\end{aligned}
$$

as $p_{v_k} - p_{v_{k-1}}$ is at most 1 (if one removes a vertex from a maximal independent set of k points, one obtains a independent set of $k-1$ points). By estimating the number of numbers $2^i 3^j$ less than x, we get that $\log v_k \simeq \sqrt{2 \log 2 \log 3k}$:

```
>> k:=4000: map(ln(v(k))=sqrt(2*ln(2)*ln(3)*k),float);
```

```
77.14988918 = 78.05126573
```

As p_{v_k} is the size of the maximal independent subset of S_{v_k}, which contains exactly k elements, we have trivially $p_{v_k} \leq k$. By Euler-Maclaurin summation using Maple, and bounding $1/v_{k+1}$ by $1/v_k$, we get:

$$E_k \leq (p_{v_k} + \frac{2k^{1/2}}{\sqrt{2 \log 2 \log 3}})\frac{1}{v_k}$$

which gives:

```
>> k:=1000: float((p(v(k))+2*k^(1/2)/sqrt(2*ln(2)*ln(3)))/v(k));

                              1.53e-14

>> k:=2000: float((p(v(k))+2*k^(1/2)/sqrt(2*ln(2)*ln(3)))/v(k));

                             2.8361e-21

>> k:=4000: float((p(v(k))+2*k^(1/2)/sqrt(2*ln(2)*ln(3)))/v(k));

                              6.564e-31
```

This corresponds to the values above: the difference between $C(1000)$ and $C(2000)$ is about $5\,10^{-15}$, and the difference between $C(2000)$ and $C(4000)$ is about $9\,10^{-22}$. Therefore we can conclude that at least 30 digits of $C(4000)$ are correct:

$$0.613475269202234416018041663 8$$

Proof of Proposition 1. Suppose that the graph has a vertices of even parity, and b vertices of odd parity, with $a \geq b$ (the other case is similar). We call *domino* a pair of adjacent vertices. If we can find a way to put b non-overlapping dominoes on the graph (with $a - b$ remaining vertices of even parity), then any independent set contains at most one vertex of each domino, plus some of the $a - b$ remaining vertices, hence at most a vertices, and as the set of vertices of even parity is independent and has a vertices, it is therefore a maximal independent set.

For instance, consider the graph with 19 vertices ($a = 10$, $b = 9$):

```
1  - 3  - 9  - 27 - 81
|    |    |    |
2  - 6  - 18 - 54
|    |    |
4  - 12 - 36
|    |    |
8  - 24 - 72
|    |
16 - 48
|
32
|
64
```

Let us first regularly place vertical dominoes:

```
1    3    9    27   81
|    |    |    |
2    6    18   54

4    12   36
|    |    |
8    24   72

16   48
|
32

64
```

We are now left with 3 isolated vertices: 81 and 64 have even parity, and 48 has odd parity. Let us select two isolated vertices of different parity, e.g. 48 and 81. They are connected by the following path of dominoes:

```
48   24 - 12   6  - 3    9  - 18   54 - 27   81
```

We can shift the dominoes in this path to make room for one more domino:

```
48 - 24   12 - 6    3  - 9    18 - 54   27 - 81
```

We then get the desired covering with 9 dominoes and one remaining vertex:

```
1    3  - 9    27 - 81
|
2    6    18 - 54
     |
4    12   36
|         |
8    24   72
     |
16   48
|
32

64
```

In general, after regularly placing the vertical dominoes, we get on the diagonal border a certain number of isolated vertices, of both parities, at most one in each column and in each even line. We pair them (one vertex of odd parity with one vertex of even parity) in such a way that paths connecting paired vertices do not intersect. This can be done by taking them in order, pairing a vertex with the last unpaired one of opposite parity, if any. Finally, to connect vertex x to vertex y, we move dominoes from the initial position

```
*   *   *   *   *   y
|   |   |   |   |
*   *   *   *   *

*
|
*

x
```

to

```
* - *   * - *   * - y

*   * - *   * - *
|
*

*
|
x
```

The only unpaired vertices will be the $a-b$ vertices of even parity in excess.

This works in fact for any grid graph having the property that (i,j) in the graph and $i>0$ implies $(i-1,j)$ in the graph, and (i,j) in the graph and $j>0$ implies $(i,j-1)$ in the graph (with a slight modification to take into account the case when we get several isolated vertices on the same line, which does not happen with the graphs initially considered).