Overview of the “Barrier Approach” to lower the upper bound of the *de Bruijn-Newman* constant.

D.H.J. Polymath

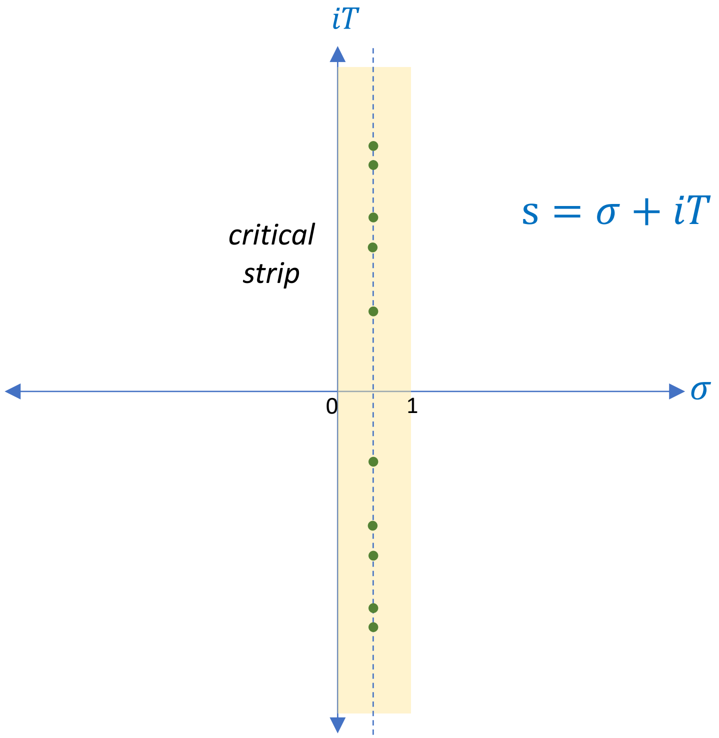
July 2018

V1.6

High level storyline

- The basic De Bruijn-Newman idea and the function $H_t(x+iy)$.
- How to effectively bound a good estimate for H_t?
- Some observations on the zeros of H_t.
- How could zeros of H_t be 'blocked' to lower the Λ upper bound.
- The key ideas behind the “Barrier approach”.
- How to ensure no zeros have passed the Barrier?
- How to show that H_t doesn’t vanish from the Barrier to N_b ?
- Optimizations used to improve computation speed.
- Numerical results showing $\Lambda < 0.22$ (and $\Lambda < 0.13$ conditionally on RH < 10^18).
- Software used, optimizations, and detailed results available.

The idea behind the De Bruijn-Newman constant

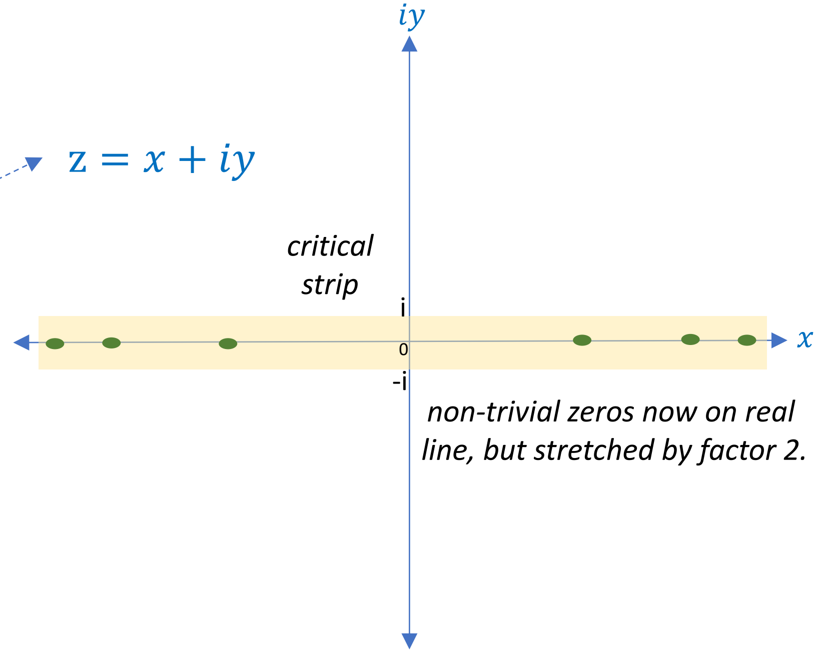


$s = \sigma + iT$

$$\xi(s) := \frac{s(s-1)}{2}\pi^{-s/2}\Gamma\left(\frac{s}{2}\right)\zeta(s)$$

$$H_0(z) := \frac{1}{8}\xi\left(\frac{1}{2} + \frac{iz}{2}\right).$$

$z = x + iy$



non-trivial zeros now on real line, but stretched by factor 2.

Fourier transform

$$H_0(z) = \int_0^\infty \Phi(u)\cos(zu)\,du$$

New family of entire functions by De Bruijn

$$\Phi(u) := \sum_{n=1}^{\infty}(2\pi^2 n^4 e^{9u} - 3\pi n^2 e^{5u})\exp(-\pi n^2 e^{4u}).$$

$$H_t(z) := \int_0^\infty e^{tu^2}\Phi(u)\cos(zu)\,du.$$

First step: develop an effectively bounded estimate.

There exists a unique *De Bruijn – Newman constant* Λ, for which it is true that all zeros of t $\geq$ Λ are real:

1950 → de Bruijn : $\Lambda \leq 0.5$

.

.

2008 → Kee/Kim/Lee : $\Lambda < 0.5$

2018 → Rodgers/Tao : $\Lambda \geq 0$

$\Lambda = 0 \Longrightarrow$ RH

Could the DBN-upper bound be reduced further?

Estimating and effectively bounding $H_t(x+iy)$

Main estimate

$$A(x+iy) := M_t\left(\frac{1-y+ix}{2}\right)\sum_{n=1}^{N}\frac{b_n^t}{n^{\frac{1-y+ix}{2}+\frac{t}{2}\alpha\left(\frac{1-y+ix}{2}\right)}}$$

$$B(x+iy) := M_t\left(\frac{1+y-ix}{2}\right)\sum_{n=1}^{N}\frac{b_n^t}{n^{\frac{1+y-ix}{2}+\frac{t}{2}\alpha\left(\frac{1+y-ix}{2}\right)}}$$

Optionally: a more effective C-term is available

Designed for:

$$0 < t \le \frac{1}{2};\quad 0 \le y \le 1;\quad x \ge 200.$$

Error terms

$$E_A(x+iy) := \left|M_t\left(\frac{1-y+ix}{2}\right)\right|\sum_{n=1}^{N}\frac{b_n^t}{n^{\frac{1-y}{2}+\frac{t}{2}\mathrm{Re}\,\alpha\left(\frac{1-y+ix}{2}\right)}}\varepsilon_{t,n}\left(\frac{1-y+ix}{2}\right)$$

$$E_B(x+iy) := \left|M_t\left(\frac{1+y+ix}{2}\right)\right|\sum_{n=1}^{N}\frac{b_n^t}{n^{\frac{1+y}{2}+\frac{t}{2}\mathrm{Re}\,\alpha\left(\frac{1+y+ix}{2}\right)}}\varepsilon_{t,n}\left(\frac{1+y+ix}{2}\right)$$

$$E_{C,0}(x+iy) := \exp\left(\frac{t\pi^2}{64}\right)|M_0(iT')|\left(1+\tilde{\varepsilon}\left(\frac{1-y+ix}{2}\right)+\tilde{\varepsilon}\left(\frac{1+y+ix}{2}\right)\right).$$

$$\boxed{H_t(x+iy) = A(x+iy) + B(x+iy) + O_{\le}(E_A(x+iy) + E_B(x+iy) + E_{C,0}(x+iy))}$$

Normalize by B_0 and bound effectively

$$\frac{H_t(x+iy)}{B_0(x+iy)} = \sum_{n=1}^{N}\frac{b_n^t}{n^{s_*}} + \gamma\sum_{n=1}^{N}n^y\frac{b_n^t}{n^{\overline{s_*}+\kappa}} + O_{\le}(e_A + e_B + e_{C,0})$$

Main estimate lower bound

(triangle) $|f_t(x+iy)| \ge 1 - |\gamma| - \sum_{n=2}^{N}\frac{b_n^t}{n^{\sigma}}(1+|\gamma|n^{y-\mathrm{Re}(\kappa)}),$

(lemma) $|f_t(x+iy)| \ge \left(\left|\sum_{n=1}^{N}\frac{b_n^t}{n^{s_*}}\right| - \left|\sum_{n=1}^{N}\frac{|\gamma|b_n^t n^y}{n^{s_*}}\right|\right)_+ - |\gamma|\sum_{n=1}^{N}\frac{b_n^t(n^{|\kappa|}-1)}{n^{\sigma-y}}.$

$$\sum_{d=1}^{D}\frac{\lambda_d}{d^s} = \prod_{p\le P}\left(1-\frac{b_p^t}{p^s}\right)$$

Choice of 'Euler mollifiers'

Hence,

Error upper bounds

$$e_A + e_B \le \sum_{n=1}^{N}(1+|\gamma|N^{|\kappa|}n^y)\frac{b_n^t}{n^{\mathrm{Re}(s_*)}}\left(\exp\left(\frac{\frac{t^2}{16}\log^2\frac{x}{4\pi n^2}+0.626}{x-6.66}\right)-1\right)$$

$$e_{C,0} \le \left(\frac{x}{4\pi}\right)^{-\frac{1+y}{4}}\exp\left(-\frac{t}{16}\log^2\frac{x}{4\pi}+\frac{1.24\times(3^y+3^{-y})}{N-0.125}+\frac{3|\log\frac{x}{4\pi}+i\frac{\pi}{2}|+10.44}{x-8.52}\right)$$

If Lower bound $\ge$ Upper bound then $H_t(x+iy) \neq 0$

Actual trajectories of real and complex zeros of $H_t(x+iy)$

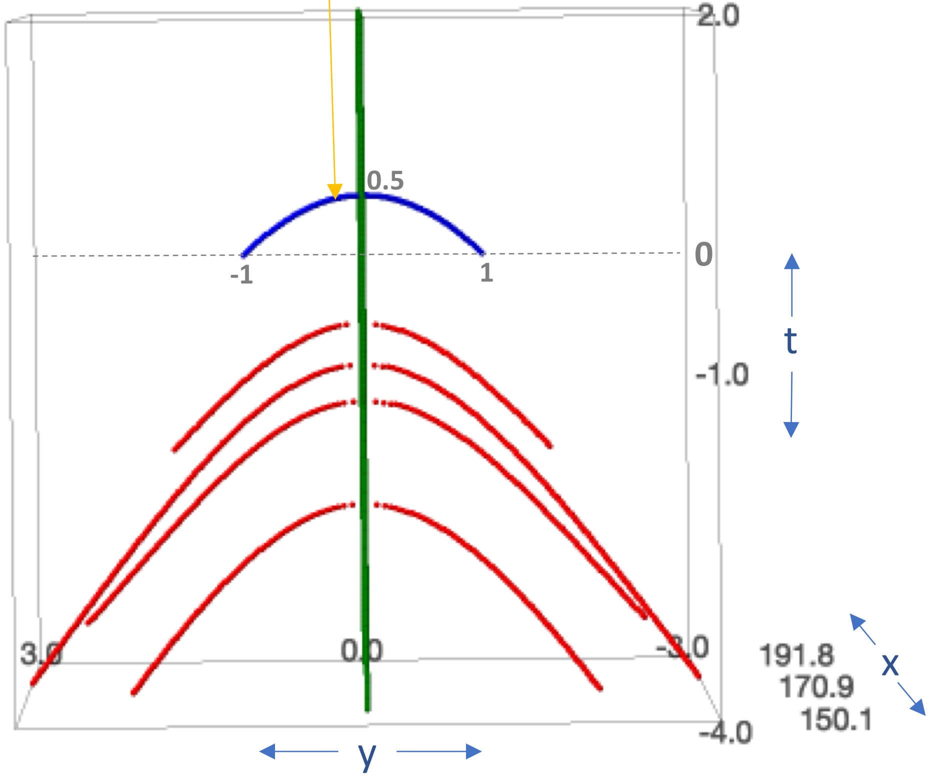
The complex parts of zeros attract each other and the real parts repel each other. From isolating the imaginary "force", it can be derived that all complex zeroes will be forced into the real axis in a finite time leading to the bound:

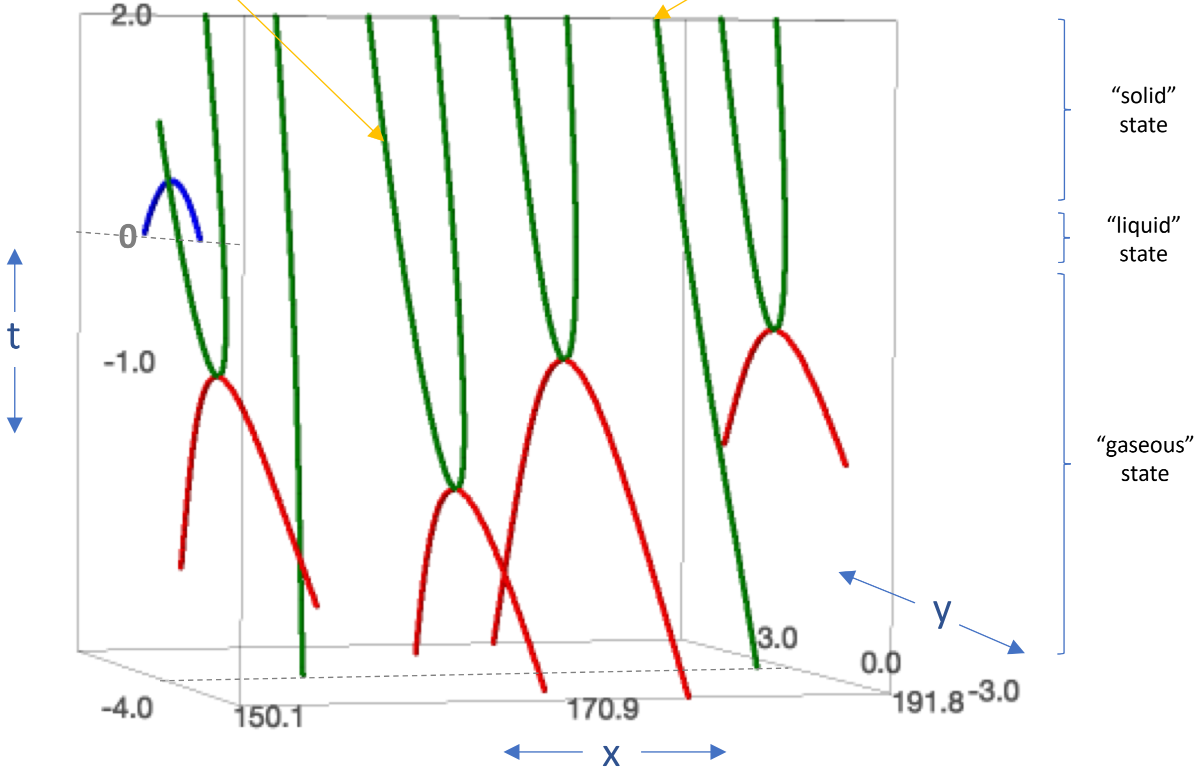
$$\Lambda \leq t + \frac{1}{2}\sigma_{max}^2(t)$$

Zeroes get denser as one moves away from the origin, so there are more zeros to the right of x_n then to the left, hence their trajectories typically "lean" leftwards.

Once a zero becomes real, it stays real forever and ends up roughly equally spaced with:

$$z_{j+1}(t) - z_j(t) = (1+o(1))\frac{4\pi}{\log z_j(t)}$$

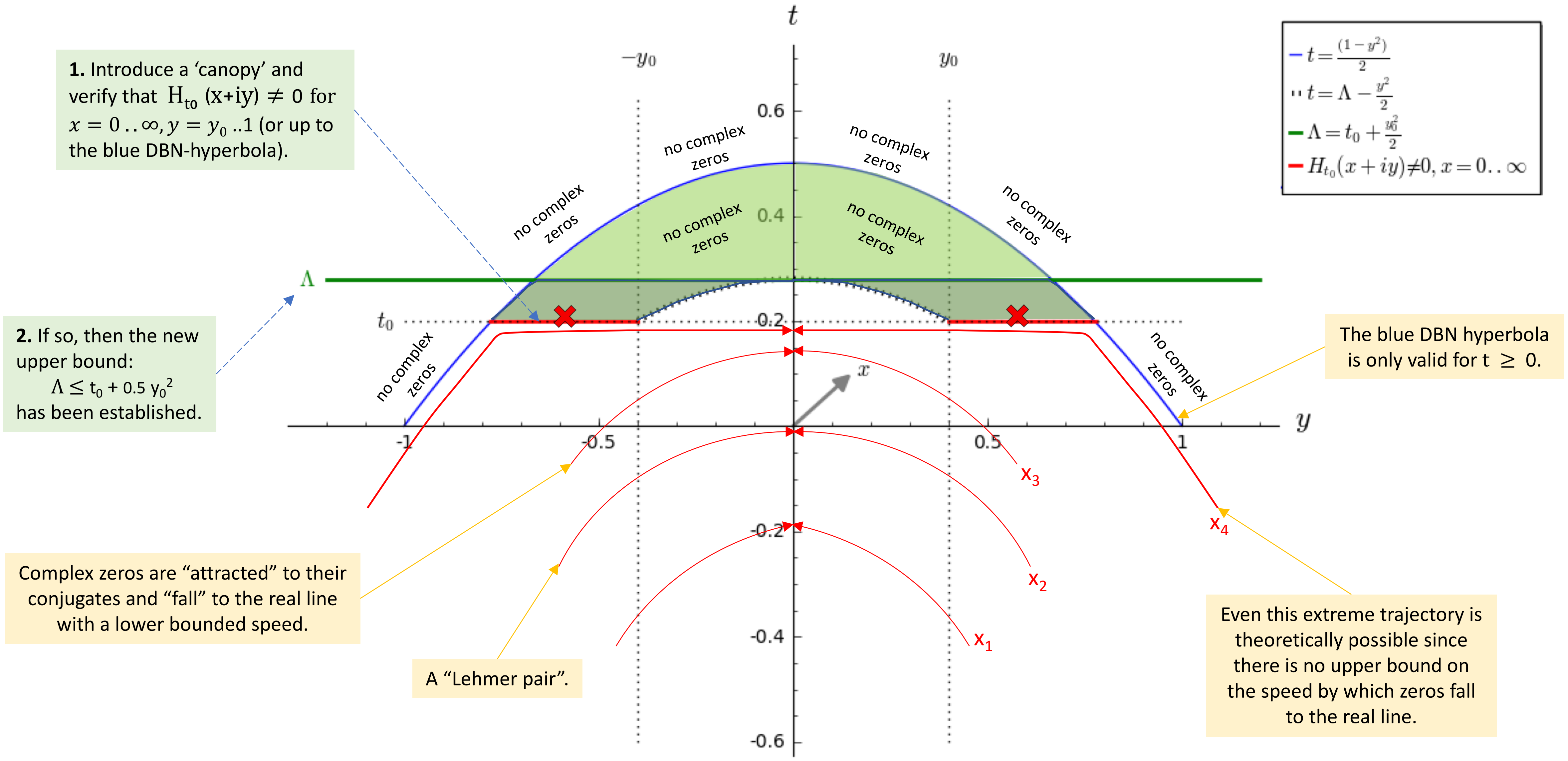




trajectory of a complex zero

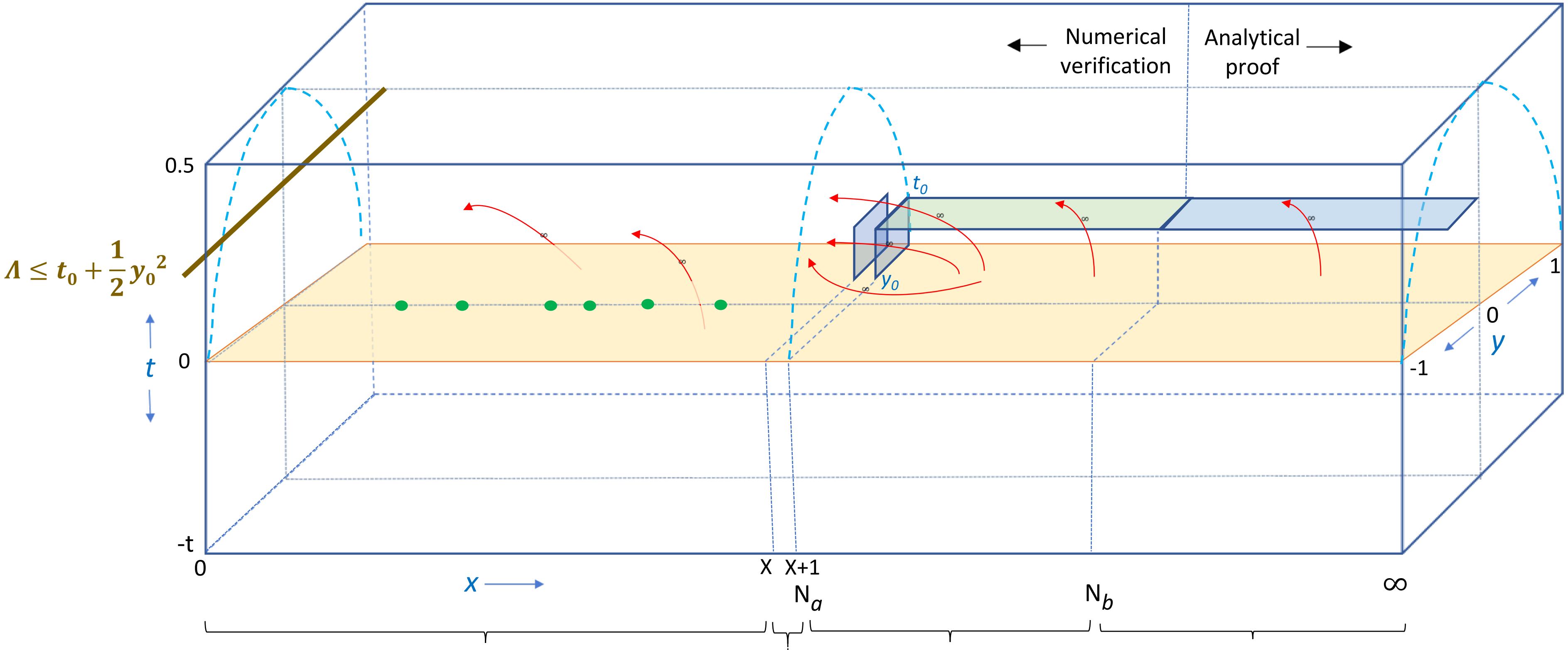
trajectory of a real zero

The De Bruijn – Newman Λ: introducing a ‘canopy’ the complex zeroes can’t cross



Possible trajectory of a complex zero $\left(H_t\,(x+i|y|_{>0}) = 0\right)$

“ Barrier” approach to assure $H_t\,(x+iy) \neq 0$ for a certain $y>y_0$, t_0.



1. Area where the RH has been verified e.g. 6x10[10] certain, 10[13] to be confirmed. Or assume that it has been verified up to X.

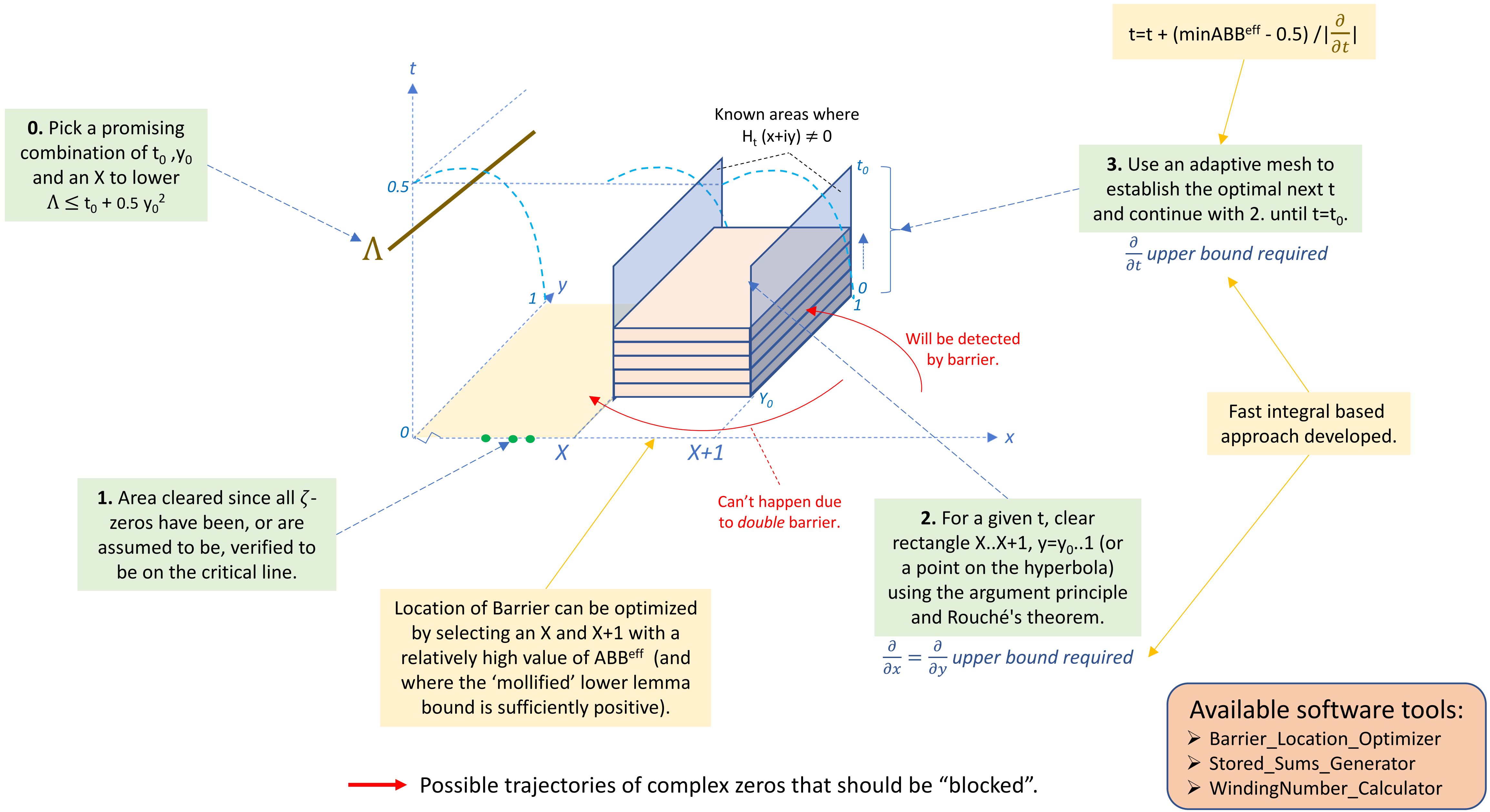
2. Verify $H_t\,(x+iy) \neq 0$ in the Barrier area $x=X..X+1$, $y=y_0..1$, $t=0..t_0$

3. Verify $H_{t0}\,(x+iy_0) \neq 0$ i.e. Lemma lower bound > Error upper bound.

4. Analytical proof that $H_{t0}\,(x+iy_0) \neq 0$

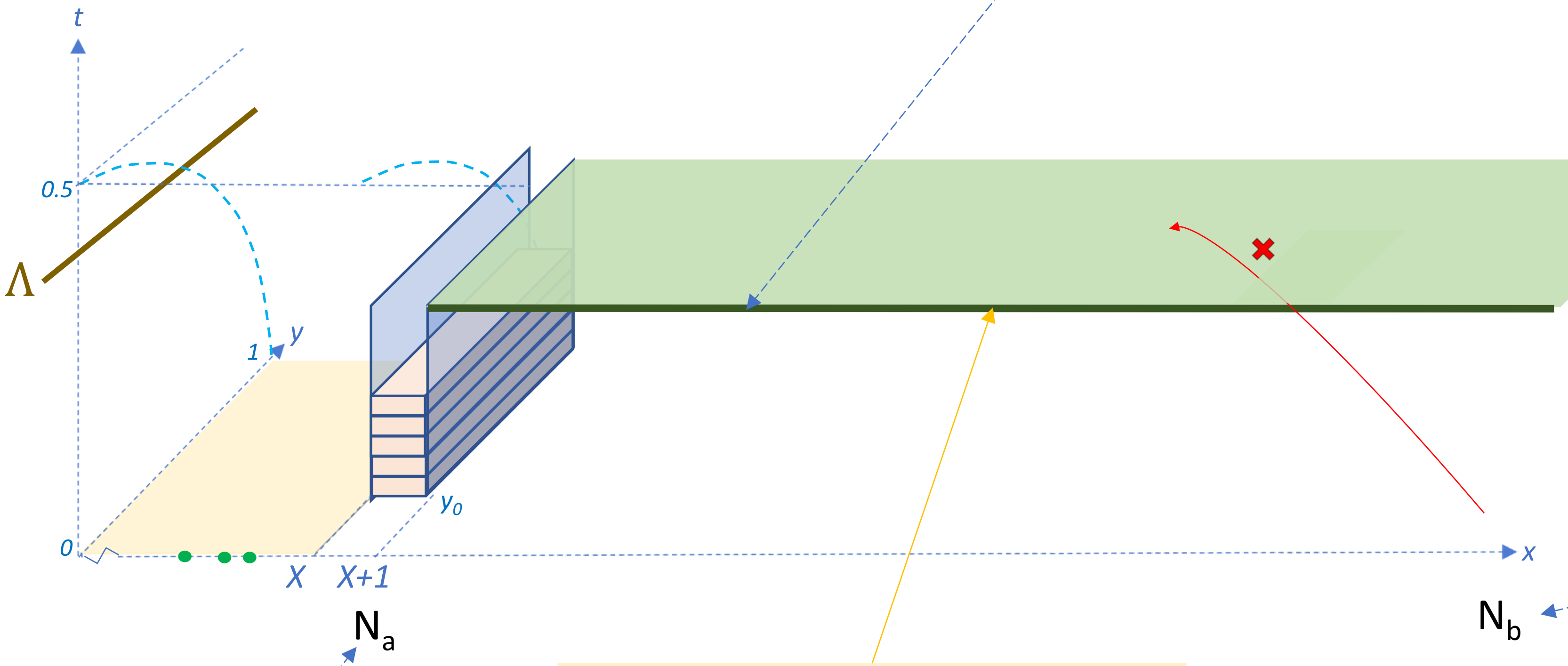
Possible trajectories of complex zeros that should be “blocked”.

“Barrier” approach: how to clear the barrier?



“Barrier” approach: how to verify the area from the barrier up to N_b?

3. Only the lower *Lemma* bound for the **line** y_0, t_0 needs to be verified to stay above the error bounds, since the Lemma bound monotonically increases for y going to 1.



2. A fast Approximate *Triangle* bound is used to establish the Nb point after which analytical proof takes over (currently ‘unmollified’, but prime 2 available).

1. Select a ‘mollifier’ that makes the *Lemma* bound sufficiently positive.

A fast “Sawtooth” mechanism has been developed, that only calculates the required incremental *Lemma* Bound terms and only requires a full calculation when the incremental bound passes a user defined threshold.

Available software tools:

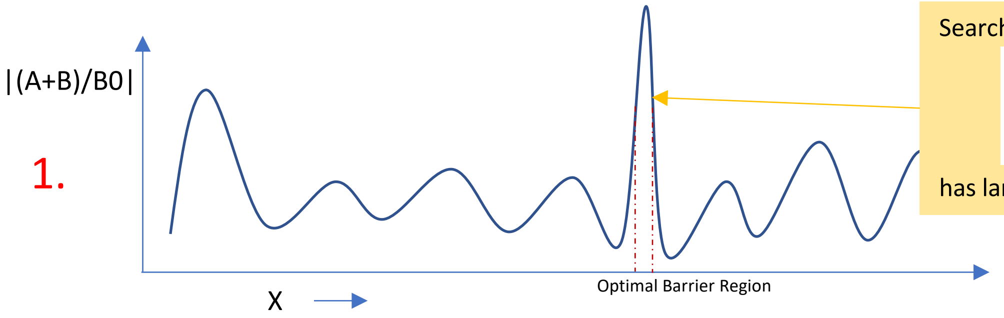
- N_a_Lemmabound_calculator
- N_b_Location_Finder
- LemmaBound_Sawtooth_calculator

Possible trajectory of complex zeros that should be “blocked”.

Optimizations made to improve computation speed

Barrier Optimizations

1.

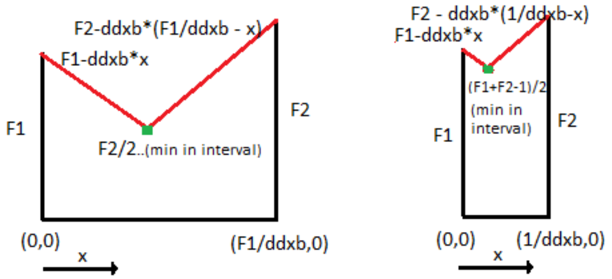


Search for X interval where:

$$\prod_{p}^{P} \frac{1}{1-\frac{1}{p^{1-iX/2}}}$$

has large relative magnitude

2.

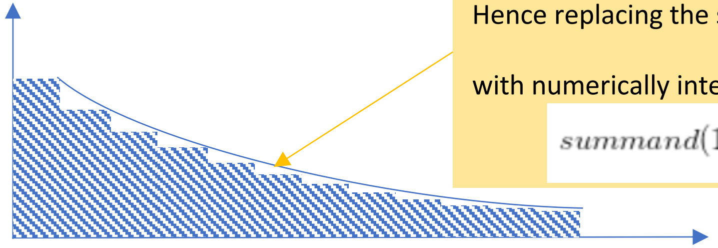


A larger t step:

$$t_{next} = t_{curr} + \frac{\min(mesh|(A+B)/B0|) - 0.5}{ddtbound}$$

is allowed due to a conservatively chosen mesh gap 1/*ddzbound*.

3.



The *ddzbound* and *ddtbound* are monotonous in X, y, t.

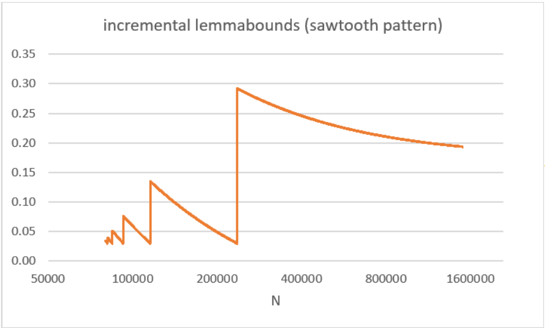
Hence replacing the slow $\sum_{n=1}^{N} summand(n)$,

with numerically integrated fast conservative:

$$summand(1) + \int_{1}^{N} summand(n)dn$$

Euler Bounds Optimizations

1.

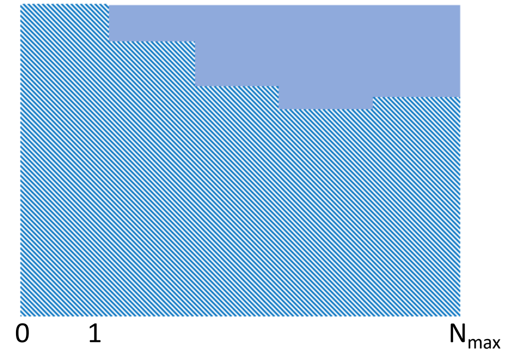


Using the incremental Lemmabound computations, as in section 8 of the writeup. When the incremental bounds reach a threshold (>> error bound), the full bound is recalculated.

2.

Fast conservative error bound for $[N_{min}, N_{max}]$.
N_{min} could also be reset at the beginning of a new sawtooth.

$$3N_{max}\max(e_B summand(n=1, X=X_{N_{min}}), e_B summand(n=N_{max}, X=X_{N_{min}})) + e_{C,0}(X=X_{N_{min}})$$



The Barrier model - progress made

X-location where min_mesh|(A+B)/B0| starts between 4 and 5 at t=0 and ends near 1.5 at $t=t_0$.

Selected with the LemmaBound utility

Selected with the TriangleBound utility

x	Barrier offset	RH verified?	t_0	y_0	Λ	Winding number	mollifier primes	Lemma bound value	Na	mollifier primes	Triangle bound value	Nb	
6.00E+10	155019	yes [1)]	0.20	0.20	0.22	0	(2,3,5,7)	0.067	69098	(0)	0.077	1.7E+06	✓
2.00E+11	187700	tentative	0.19	0.20	0.21	0	(2,3,5,7)	0.095	126156	(0)	0.121	4.0E+06	✓
1.00E+12	46880	tentative	0.18	0.20	0.20	0	(2,3,5)	0.135	282094	(0)	0.112	7.0E+06	✓
5.00E+12	194858	tentative [2)]	0.17	0.20	0.19	0	(2,3,5)	0.180	630783	(0)	0.116	1.5E+07	✓
1.00E+13	9995	not yet	0.16	0.20	0.18	0	(2,3,5)	0.109	892062	(0)	0.091	3.0E+07	✓
1.00E+14	2659	not yet	0.15	0.20	0.17	0	(2,3,5)	0.195	2820947	(0)	0.076	7.0E+07	✓
1.00E+15	21104	not yet	0.14	0.20	0.16	0	(2,3,5)	0.251	8920620	(0)	0.073	2.0E+08	✓
1.00E+16	172302	not yet	0.13	0.20	0.15	0	(2,3,5)	0.278	28209479	(0)	0.077	7.0E+08	✓
1.00E+17	31656	not yet	0.12	0.20	0.14	0	(2,3,5)	0.279	89206205	(0)	0.080	3.0E+09	✓
1.00E+18	44592	not yet	0.11	0.20	0.13	0	(2,3)	0.207	282094791	(0)	0.103	2.0E+10	✓
1.00E+19	12010	not yet	0.10	0.20	0.12	tbd	(2,3)	0.128	892062059	(2)	0.109	2.0E+09	
1.00E+20	37221	not yet	0.09	0.20	0.11	tbd	(2,3,5)	0.037	2820947918	(2)	0.094	1.3E+10	

1) Platt et al, 2011
2) Gourdon-Demichel, 2004

Software used and useful links

All software was developed in two languages and results between them were reconciled:

- Symbolic math programming language - **pari/gp** (https://pari.math.u-bordeaux.fr)
  - Short development time
  - Relatively fast
- Arithmetic Balls C-based library - **Arb** (http://arblib.org)
  - Longer development time
  - Very fast (up to 20 x pari/gp)

All software and results are free to use (under the LGPL-terms) and can be found here:

https://github.com/km-git-acc/dbn_upper_bound