How to split a terapolynomial ?

Nicolae Mihalache
nicolae.mihalache@u-pec.fr

Univ. Paris-Est Créteil
LAMA
UMR 8050

François Vigneron
francois.vigneron@univ-reims.fr

Univ. Reims Champagne-Ardenne
LMR
UMR 9008

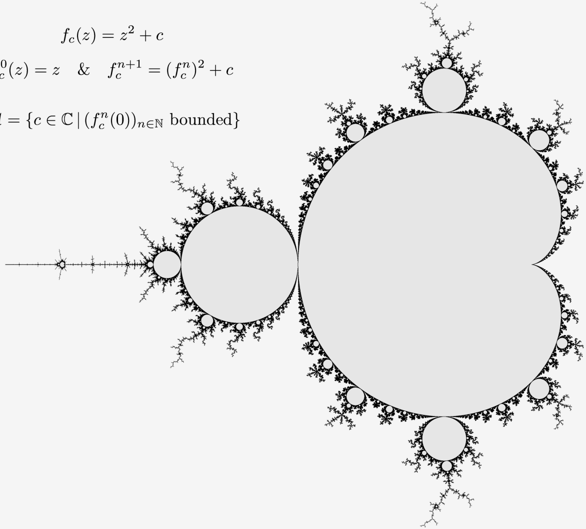
JCAD 2021 – 13-15 DEC.

Mandelbrot set $\mathcal{M}$

$$f_c(z) = z^2 + c$$

$$f_c^0(z) = z \quad \& \quad f_c^{n+1} = (f_c^n)^2 + c$$

$$\mathcal{M} = \{c \in \mathbb{C} \,|\, (f_c^n(0))_{n\in\mathbb{N}} \text{ bounded}\}$$



Polynomials of special interest

$$p_0(c) = 0 \qquad p_{k+1}(c) = p_k(c)^2 + c \qquad \deg p_k = 2^{k-1}$$

- **Hyperbolic centers**:

$$C_k \, : \, \text{roots of } p_k \qquad \text{(reduced without roots of divisors)}$$

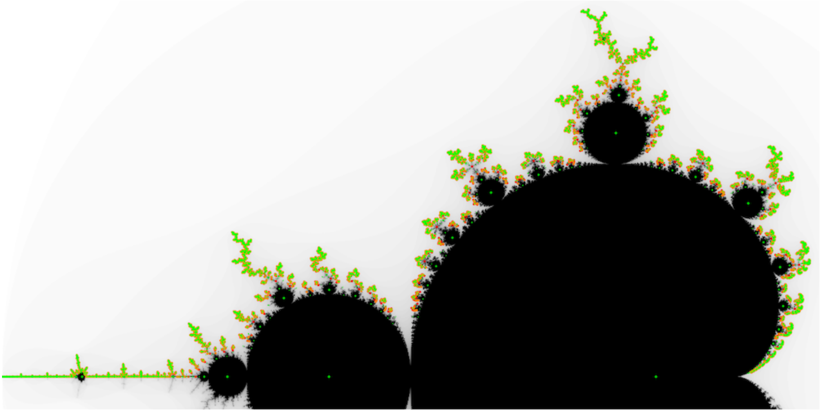
$\rightarrow$ C_k parametrizes all the orbits of period k that contain 0.
$\rightarrow$ Centers of the "hyperbolic" components of $\mathrm{Int}(\mathcal{M})$.

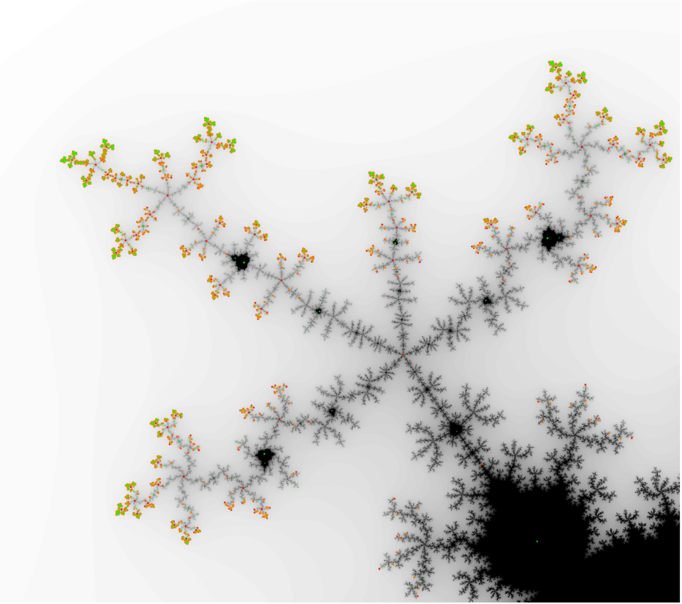
- **Misiurewicz-Thurston points**:

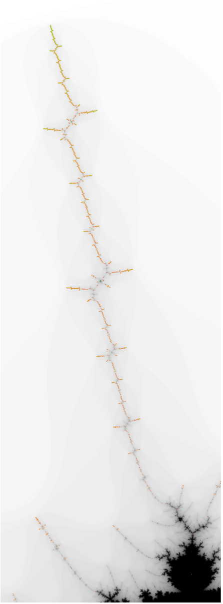
$$M_{m,k} \, : \, \text{roots of } p_{m+k} - p_k \qquad \text{(reduced)}$$

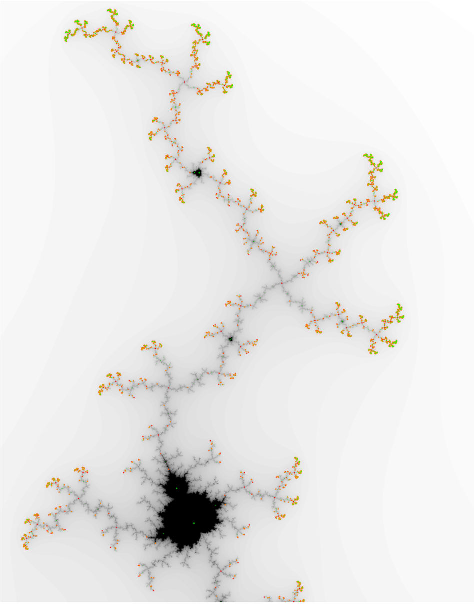
$\rightarrow$ $M_{m,k}$ describes pre-periodic orbits (m jumps before k-periodic).
$\rightarrow$ Tips & "branching nodes" in $\partial\mathcal{M}$.

$$C_1 \cup C_2 \cup \ldots C_{18} \quad \text{and} \quad \bigcup_{m+k\leq 20} M_{m,k}$$









The challenges of splitting a tera-polynomial

Raw search time (in days-core, dc or years-core, yc) *in $\Im m\ z \geq 0$

Order N	Hyperbolic C_N		ms/new root*	Misiurewicz $M_{m+k=N}$		ms/new root*
31	1 073 741 823	2.6 dc	0.41	29 683 757 317	237 dc	2.8 [1.9 to 9.9]
32	2 147 450 880	5.7 dc	0.44	61 514 965 058	1.5 yc	3.2 [3.1 to 11.0]
33	4 294 966 269	11.9 dc	0.46	127 324 766 178	4.3 yc	4.4 [2.8 to 17.3]
34	8 589 869 055	24.1 dc	0.47	263 239 398 474	11.7 yc	5.6 [2.9 to 18.2]
35	17 179 869 105	50.8 dc	0.49	543 658 403 426	+13.8 yc	
36	34 359 605 280	106 dc	0.51			
37	68 719 476 735	227 dc	0.55			
38	137 438 691 327	1.4 yc	0.61			
39	274 877 902 845	2.9 yc	0.65			
40	549 755 289 480	6.4 yc	0.71			
41	**1 099 511 627 775**	14.3 yc	0.80			
		26.2 yc	0.73		+31.9 yc	

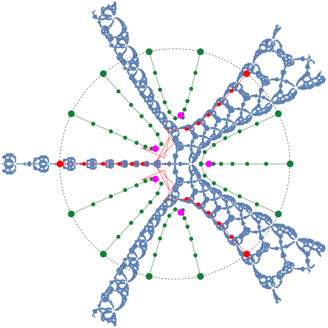
HPC Center ROMEO Centre de Calcul Regional

C_{41} computed (dec. 2020)

$M_{m+k=35}$ in progress.

Finding the roots

Newton's method: iterate $N_P(z) = z - \frac{P(z)}{P'(z)}$ from $O(d \log d)$ points[1]



→ Universal & guarantied.

→ Extremely small Newton steps !

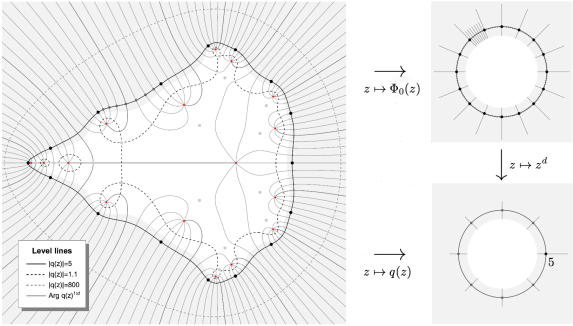
$$N_{z^d}(z) = z\left(1 - \frac{1}{d}\right)$$

$\simeq d \log 2$ steps from $|z| = 2$ to $|z| \simeq 1$.

State of the art: $O(d^2 \log d)$ steps algorithm,
practical only up to degree $d \sim 10^6$!

[1] J. Hubbard, D. Schleicher, S. Sutherland (2001).

A good "encircling" map (Riemann map of $\mathcal{M}^c$) provides a parametrization of the level lines that "treats all roots fairly":
$\rightarrow$ more points around root clusters ($\sim$self-refining mesh).

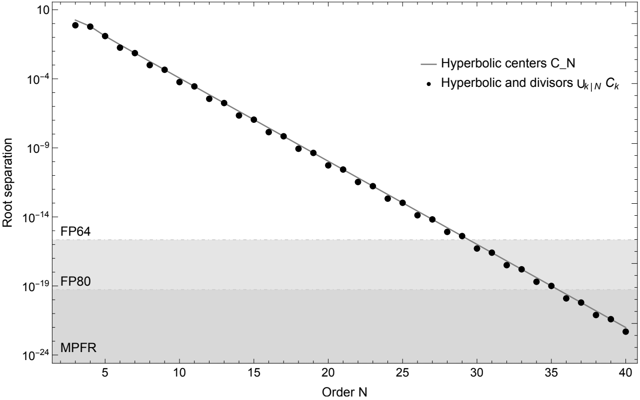


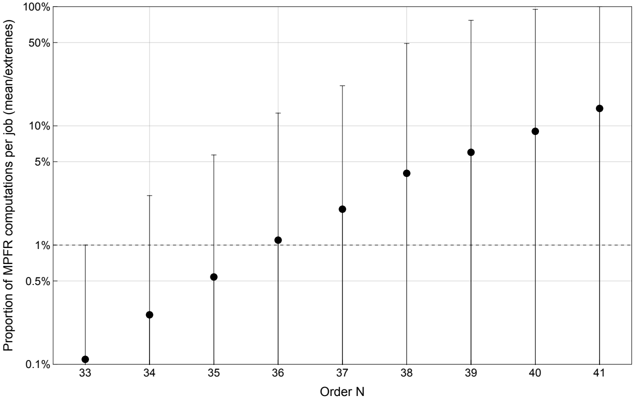
*	Level line from 16384 seeds	48 Newton steps/root
*	Search / Refinement	8 steps in FP80 + 3 steps in FP128

$\Rightarrow O(d)$ algorithm (= same cost as checking an oracle) !

Express the roots

C_k is inaccessible to FP64 (`double`) arithmetic for $k \geq 25$.

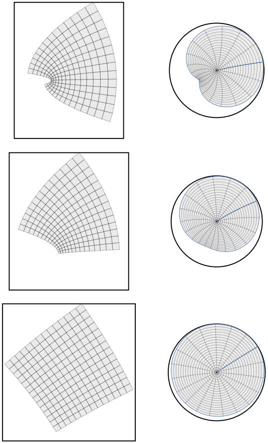




Most computations are done with FP80 (`long double`); when necessary, we switch **on the fly** to FP128 (`MPFR` library).

Certify the roots

Each root is **certified** (as a zero + refinable) using disk arithmetic.



Disk arithmetic is superior to interval arithmetic !

$$f(z) = z^2 \qquad z_0 = e^{3i\pi/16}$$

Images of

$$z_0 + [\pm r \pm ir]$$
$$z_0 + \mathrm{D}(0, r)$$

for $r = 0.8$ (top), 0.5 (middle), 0.1 (bottom).

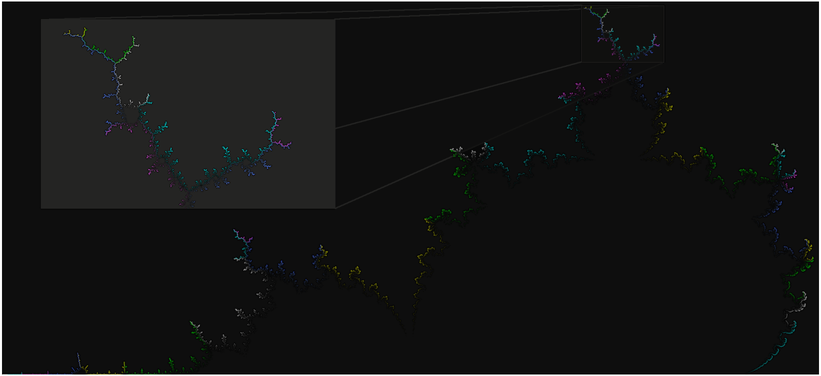
Loss after n iterates from $|z| = 1$:

$$re^{0.2365n} \quad \text{(box)}$$

$$r(1+r/2)^n \simeq r\left(1 + \frac{nr}{2}\right) \quad \text{(disk)}$$

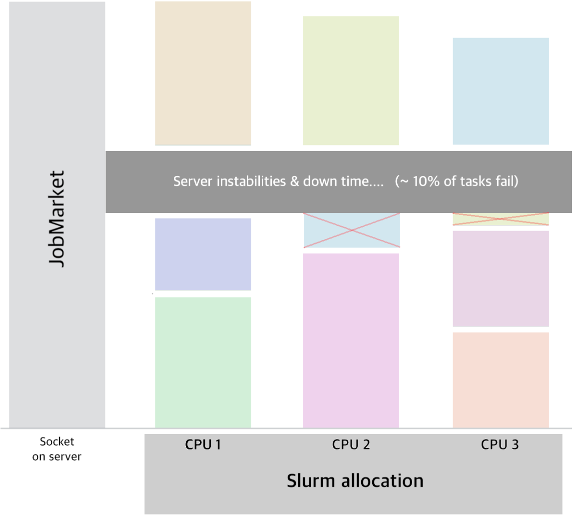
Parallelize the tasks

The search for C_{32} is split into 32 parallel tasks



- Each task produces a list of $\sim$35 millions roots.
- Up to 16384 tasks for the tera-polynomial !
- Mostly a local search (but overlap with distant tasks)

JobMarket: bookkeeping and task allocation ("vertical" parallelism)



Manage the data : sort / union of the roots

Data: Hyperbolic **17TB** (curated), Misiurewicz **10TB** (current).

Peak disk usage (during curation): **200%**/dataset; 150% of total.

$\rightarrow$ Search efficiency of each search task:

- 25% of search paths provide a new root.
- 73% of the paths produce duplicates.
- $\leq$2% of the searches are trashed (non convergent, big jumps,...).

$\rightarrow$ Find and prune duplicates

- Within each task : RAM only, no disk wasted.
- Across tasks : high IO+RAM job.

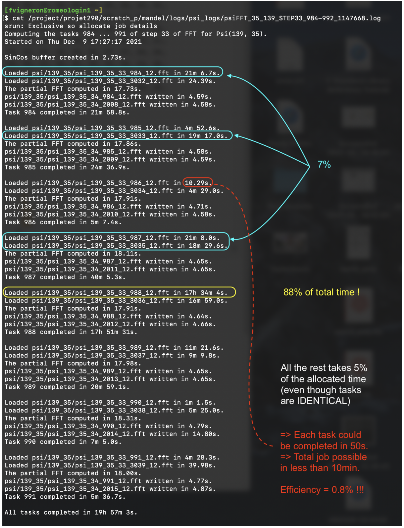
$\rightarrow$ Count & map the roots (dyadic tree) + write sorted list.

Secure file format: header contains MD5 of each data set !

Disk instabilities on *Romeo* (investigation pending):

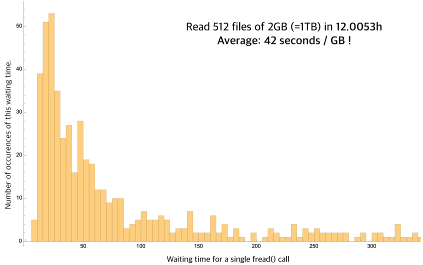
When *Romeo* is under load, the variability in loading times is extreme:

> **99% of the allocation time is lost** waiting for the completion of `fread()`...

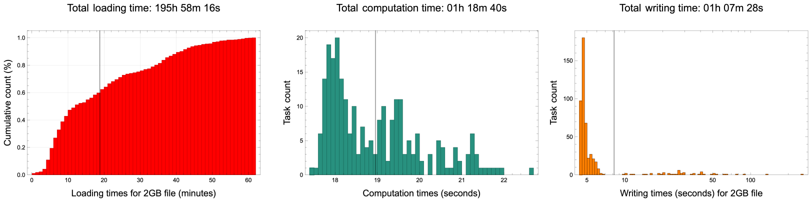


```
[fvigneron@romeologin1 ~]
$ cat /project/projet290/scratch_p/mandel/logs/psi_logs/psiFFT_35_139_STEP33_984-992_1147668.log
srun: Exclusive so allocate job details
Computing the tasks 984 ... 991 of step 33 of FFT for Psi(139, 35).
Started on Thu Dec  9 17:27:17 2021

SinCos buffer created in 2.73s.

Loaded psi/139_35/psi_139_35_33_984_12.fft in 21m 6.7s.
Loaded psi/139_35/psi_139_35_33_3032_12.fft in 24.39s.
The partial FFT computed in 17.73s.
psi/139_35/psi_139_35_34_984_12.fft written in 4.59s.
psi/139_35/psi_139_35_34_2008_12.fft written in 4.58s.
Task 984 completed in 21m 58.8s.

Loaded psi/139_35/psi_139_35_33_985_12.fft in 4m 52.6s.
Loaded psi/139_35/psi_139_35_33_3033_12.fft in 19m 17.0s.
The partial FFT computed in 17.86s.
psi/139_35/psi_139_35_34_985_12.fft written in 4.58s.
psi/139_35/psi_139_35_34_2009_12.fft written in 4.59s.
Task 985 completed in 24m 36.9s.

Loaded psi/139_35/psi_139_35_33_986_12.fft in 10.29s.
Loaded psi/139_35/psi_139_35_33_3034_12.fft in 4m 29.0s.
The partial FFT computed in 17.91s.
psi/139_35/psi_139_35_34_986_12.fft written in 4.71s.
psi/139_35/psi_139_35_34_2010_12.fft written in 4.58s.
Task 986 completed in 5m 7.4s.

Loaded psi/139_35/psi_139_35_33_987_12.fft in 21m 8.0s.
Loaded psi/139_35/psi_139_35_33_3035_12.fft in 18m 29.6s.
The partial FFT computed in 18.11s.
psi/139_35/psi_139_35_34_987_12.fft written in 4.65s.
psi/139_35/psi_139_35_34_2011_12.fft written in 4.65s.
Task 987 completed in 40m 5.3s.

Loaded psi/139_35/psi_139_35_33_988_12.fft in 17h 34m 4s.
Loaded psi/139_35/psi_139_35_33_3036_12.fft in 16m 59.0s.
The partial FFT computed in 17.91s.
psi/139_35/psi_139_35_34_988_12.fft written in 4.64s.
psi/139_35/psi_139_35_34_2012_12.fft written in 4.66s.
Task 988 completed in 17h 51m 31s.

Loaded psi/139_35/psi_139_35_33_989_12.fft in 11m 21.6s.
Loaded psi/139_35/psi_139_35_33_3037_12.fft in 9m 9.8s.
The partial FFT computed in 17.98s.
psi/139_35/psi_139_35_34_989_12.fft written in 4.65s.
psi/139_35/psi_139_35_34_2013_12.fft written in 4.65s.
Task 989 completed in 20m 59.1s.

Loaded psi/139_35/psi_139_35_33_990_12.fft in 1m 1.5s.
Loaded psi/139_35/psi_139_35_33_3038_12.fft in 5m 25.0s.
The partial FFT computed in 18.31s.
psi/139_35/psi_139_35_34_990_12.fft written in 4.79s.
psi/139_35/psi_139_35_34_2014_12.fft written in 14.80s.
Task 990 completed in 7m 5.0s.

Loaded psi/139_35/psi_139_35_33_991_12.fft in 4m 28.3s.
Loaded psi/139_35/psi_139_35_33_3039_12.fft in 39.98s.
The partial FFT computed in 18.00s.
psi/139_35/psi_139_35_34_991_12.fft written in 4.77s.
psi/139_35/psi_139_35_34_2015_12.fft written in 4.87s.
Task 991 completed in 5m 36.7s.

All tasks completed in 19h 57m 3s.
```

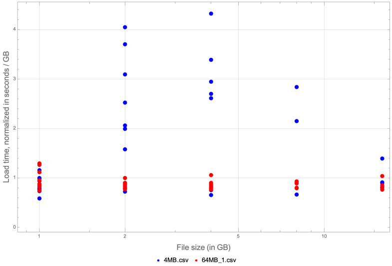
First example (average access time $\sim$42 s/GB = 24MB/s):

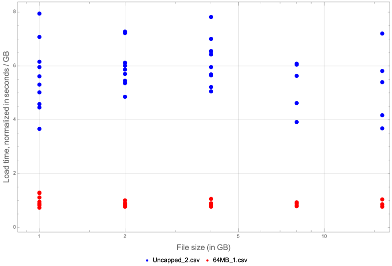


Second example (average access time $\sim$9 min/GB = 1.9 MB/s):



Paradoxal influence of `fread()` size:





→ optimal read size : 64MB without load, unknown under high load
→ common accross file size (1GB to 16GB test; 65GB in production)

→ `fwrite()` is not (much of) a problem, but is unstable too!

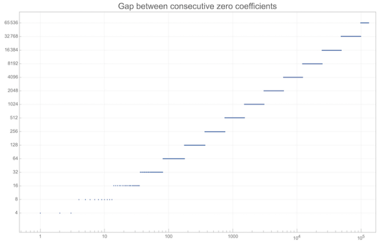
Next challenge

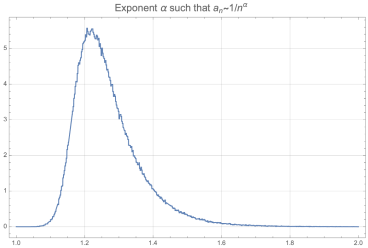
Compute the Riemann map $\mathbb{D}^c \to \mathcal{M}^c$

$$\Psi(z) = z + \sum_{n\geq 0} b_n z^{-n}$$

$\rightarrow$ State of the art: 5 million FP64 coefficients[2]

$\rightarrow$ 10 Dec. 2021: $\sim$ 34 billion coefficients with 139 certified digits (2TB) using new method based on FFT with 4TB dataset.





[2] D. Bittner, L. Cheong, D. Gates, H.D. Nguyen, 2014.

Detail of Hincmar's tomb, ca. 882. St. Remy museum, Reims.

Thank you