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 $\mathcal{M}$

\[
f_c(z) = z^2 + c
\]
\[
f_c^0(z) = z \quad \& \quad f_c^{n+1} = (f_c^n)^2 + c
\]
\[
\mathcal{M} = \{ c \in \mathbb{C} \mid (f_c^n(0))_{n \in \mathbb{N}} \text{ bounded} \} 
\]
Polynomials of special interest

\[ p_0(c) = 0 \quad p_{k+1}(c) = p_k(c)^2 + c \quad \text{deg} \, p_k = 2^{k-1} \]

- **Hyperbolic centers:**
  \[ C_k : \text{roots of } p_k \quad \text{(reduced without roots of divisors)} \]
  \[ \rightarrow C_k \text{ parametrizes all the orbits of period } k \text{ that contain } 0. \]
  \[ \rightarrow \text{Centers of the “hyperbolic” components of } \text{Int}(\mathcal{M}). \]

- **Misiurewicz-Thurston points:**
  \[ M_{m,k} : \text{roots of } p_{m+k} - p_k \quad \text{(reduced)} \]
  \[ \rightarrow M_{m,k} \text{ describes pre-periodic orbits \((m \text{ jumps before } k\text{-periodic}). \]
  \[ \rightarrow \text{Tips & “branching nodes” in } \partial \mathcal{M}. \]
\[ C_1 \cup C_2 \cup \ldots C_{18} \text{ and } \bigcup_{m+k \leq 20} M_{m,k} \]
Introduction Challenges Now what?

Mandelbrot Polynomials

$C_k \& M_{m,k}$

François VIGNERON, Univ. de Reims Champagne-Ardenne

Terapolynomial
Introduction

Challenges

Now what?

Mandelbrot Polynomials

$C_k \& M_{m,k}$

François VIGNERON, Univ. de Reims Champagne-Ardenne

Terapolynomial
The challenges of splitting a tera-polynomial

Raw search time (in days-core, \( dc \) or years-core, \( yc \))

\[ \text{Order } N \quad \text{Hyperbolic } C_N \quad \text{ms/new root*} \quad \text{Misiurewicz } M_{m+k=N} \quad \text{ms/new root*} \]

<table>
<thead>
<tr>
<th>Order</th>
<th>Hyperbolic $C_N$</th>
<th>Order</th>
<th>Hyperbolic $C_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>1 073 741 823</td>
<td>61</td>
<td>543 658 403 426</td>
</tr>
<tr>
<td>32</td>
<td>2 147 450 880</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>4 294 966 269</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>8 589 869 055</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>17 179 869 105</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>34 359 605 280</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>68 719 476 735</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>137 438 691 327</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>274 877 902 845</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>549 755 289 480</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>1 099 511 627 775</td>
<td>71</td>
<td></td>
</tr>
</tbody>
</table>

\[ C_{41} \text{ computed (dec. 2020)} \]

\[ M_{m+k=35} \text{ in progress.} \]
Finding the roots

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

\( \rightarrow \) Universal & guarantied.
\( \rightarrow \) 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”: 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 (\texttt{long double}); when necessary, we switch \textbf{on the fly} to FP128 (MPFR library).
Certify the roots

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

\[ f(z) = z^2 \quad z_0 = e^{3i\pi/16} \]

Images of \( z_0 + [\pm r \pm ir] \)
\[ z_0 + D(0, r) \]
for \( r = 0.8 \) (top), 0.5 (middle), 0.1 (bottom).

---

Disk arithmetic is superior to interval arithmetic!

Loss after \( n \) iterates from \( |z| = 1 \):

\[ re^{0.2365n} \quad \text{(box)} \]
\[ r(1+r/2)^n \approx r \left( 1 + \frac{nr}{2} \right) \quad \text{(disk)} \]
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.

→ Search efficiency of each search task:
  - 25% of search paths provide a new root.
  - 73% of the paths produce duplicates.
  - ≤2% of the searches are trashed (non convergent, big jumps, ...).

→ Find and prune duplicates
  - Within each task: RAM only, no disk wasted.
  - Across tasks: high IO+RAM job.

→ 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(...)

François VIGNERON, Univ. de Reims Champagne-Ardenne
First example (average access time $\sim$42 s/GB $= 24$MB/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 $\mathbb{D}^c \rightarrow \mathcal{M}^c$

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

→ State of the art: 5 million FP64 coefficients

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

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

Thank you