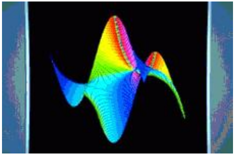
Combinatorial Newton Iteration and Efficient Random Generation

Bruno Salvy
Bruno.Salvy@inria.fr

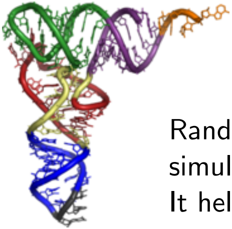
Algorithms Project, Inria

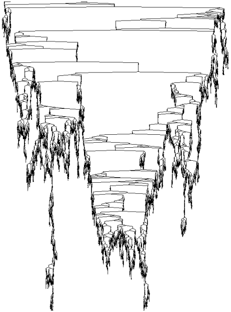


Gecko-Tera days — November 27, 2008
Joint work with Carine Pivoteau and Michèle Soria
(MCS'08)

I Introduction

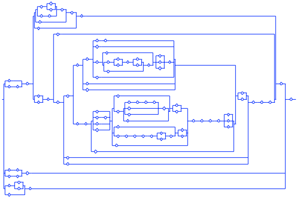
Motivation: Random Generation

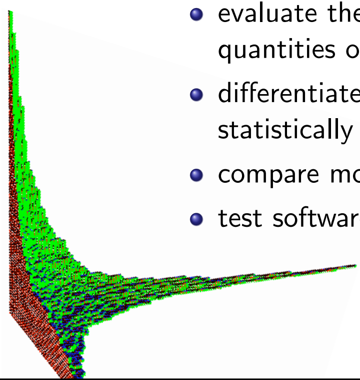




Random generation of large objects = simulation in the discrete world.
It helps

- evaluate the order of magnitude of quantities of interest;
- differentiate exceptional values from statistically expected ones;
- compare models;
- test software.





Boltzmann Samplers

Principle (Duchon, Flajolet, Louchard, Schaeffer 2004)

Generate each $t \in \mathcal{T}$ with probability $x^{|t|}/T(x)$, where: $x > 0$ fixed; $T(z) := \sum_{t \in \mathcal{T}} z^{|t|}$ = generating series of $\mathcal{T}$; $|t|$ = size.

Same size, same probability
Expected size $xT'(x)/T(x)$ increases with x.

Complexity linear in $|t|$ when the values $T(x)$ are available.

Boltzmann Samplers

Principle (Duchon, Flajolet, Louchard, Schaeffer 2004)

Generate each $t \in \mathcal{T}$ with probability $x^{|t|}/T(x)$, where: $x > 0$ fixed; $T(z) := \sum_{t\in\mathcal{T}} z^{|t|}$ = generating series of $\mathcal{T}$; $|t|$ = size.

Same size, same probability
Expected size $xT'(x)/T(x)$ increases with x.

Singleton

Easy.

Complexity linear in $|t|$ when the values $T(x)$ are available.

Boltzmann Samplers

Principle (Duchon, Flajolet, Louchard, Schaeffer 2004)

Generate each $t \in \mathcal{T}$ with probability $x^{|t|}/T(x)$, where: $x > 0$ fixed; $T(z) := \sum_{t \in \mathcal{T}} z^{|t|}$ = generating series of $\mathcal{T}$; $|t|$ = size.

Same size, same probability

Expected size $xT'(x)/T(x)$ increases with x.

Singleton

Easy.

Cartesian Product $\mathcal{C} = \mathcal{A} \times \mathcal{B}$

- Generate $a \in \mathcal{A}$; $b \in \mathcal{B}$;
- Return (a, b).

Proof. $C(x) = \sum_{(a,b)} x^{|a|+|b|} = A(x)B(x)$; $\frac{x^{|a|+|b|}}{C(x)} = \frac{x^{|a|}}{A(x)} \frac{x^{|b|}}{B(x)}$.

Complexity linear in $|t|$ when the values $T(x)$ are available.

Boltzmann Samplers

Principle (Duchon, Flajolet, Louchard, Schaeffer 2004)

Generate each $t \in \mathcal{T}$ with probability $x^{|t|}/T(x)$, where: $x > 0$ fixed; $T(z) := \sum_{t\in\mathcal{T}} z^{|t|}$ = generating series of $\mathcal{T}$; $|t|$ = size.

Same size, same probability

Expected size $xT'(x)/T(x)$ increases with x.

Singleton

Easy.

Cartesian Product $\mathcal{C} = \mathcal{A} \times \mathcal{B}$

- Generate $a \in \mathcal{A}$; $b \in \mathcal{B}$;
- Return (a, b).

Disjoint Union $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$

- Draw $b = \text{Bernoulli}(A(x)/C(x))$;
- If $b = 1$ then generate $a \in \mathcal{A}$ else generate $b \in \mathcal{B}$.

Proof. $\frac{x^{|a|}}{C(x)} = \frac{x^{|a|}}{A(x)} \frac{A(x)}{C(x)}$.

Complexity linear in $|t|$ when the values $T(x)$ are available.

Boltzmann Samplers

Principle (Duchon, Flajolet, Louchard, Schaeffer 2004)

Generate each $t \in \mathcal{T}$ with probability $x^{|t|}/T(x)$, where: $x > 0$ fixed; $T(z) := \sum_{t\in\mathcal{T}} z^{|t|}$ = generating series of $\mathcal{T}$; $|t|$ = size.

Same size, same probability

Expected size $xT'(x)/T(x)$ increases with x.

Singleton

Easy.

Cartesian Product $\mathcal{C} = \mathcal{A} \times \mathcal{B}$

- Generate $a \in \mathcal{A}$; $b \in \mathcal{B}$;
- Return (a, b).

Disjoint Union $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$

- Draw $b = \text{Bernoulli}(A(x)/C(x))$;
- If $b = 1$ then generate $a \in \mathcal{A}$ else generate $b \in \mathcal{B}$.

Use recursively (e.g., binary trees $\mathcal{B} = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$)

Also: sets, cycles,...

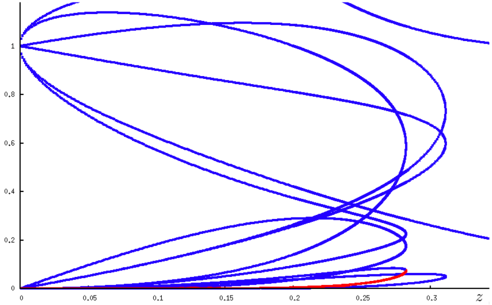
Complexity linear in $|t|$ when the values $T(x)$ are available.

Large Systems that are Interesting to Solve

The generating series are given by systems of equations.

We need:

- only one solution;
- the right one;
- only numerically.



In the worst case, these requirements would make no difference.
But these systems inherit structure from combinatorics.

Examples (I): Polynomial Systems (Darrasse 2008)

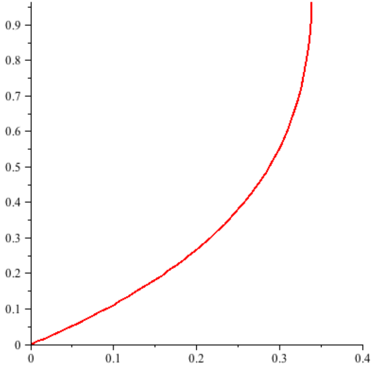
Random generation following given XML grammars.

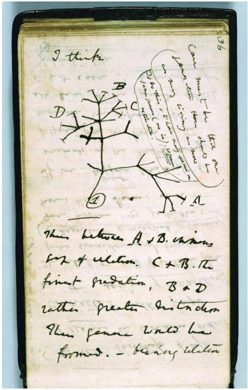
Grammar	nb eqs	max deg	nb sols	oracle (s.)	FGb (s.)
rss	10	5	2	0.02	0.03
PNML	22	4	4	0.05	0.1
xslt	40	3	10	0.4	1.5
relaxng	34	4	32	0.4	3.3
xhtml-basic	53	3	13	1.2	18
mathml2	182	2	18	3.7	882
xhtml	93	6	56	3.4	1124
xhtml-strict	80	6	32	3.0	1590
xmlschema	59	10	24	0.5	6592
SVG	117	10		5.8	>1.5Go
docbook	407	11		67.7	>1.5Go
OpenDoc	500			3.9	

Example (II): A Non-Polynomial "System"

Unlabelled rooted trees:

$$f(x) = x \exp\left(f(x) + \frac{1}{2} f(x^2) + \frac{1}{3} f(x^3) + \cdots\right)$$

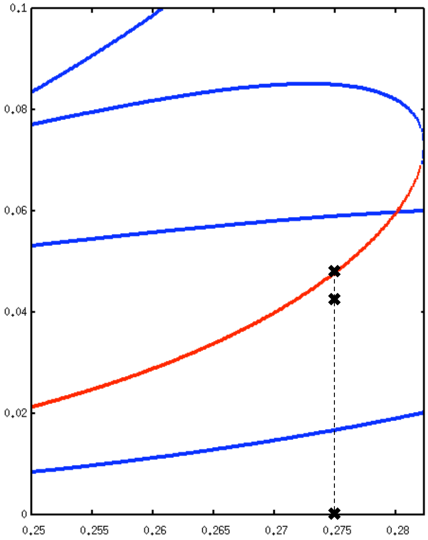




Main Result of (Pivoteau, Salvy, Soria 2008)

Theorem

For all (well-founded) systems of generating series, Newton iteration starting from 0 converges to the desired solution.



II Combinatorics

Combinatorial Specifications

Language and Gen. Fcns (labelled)

$\mathcal{A} \cup \mathcal{B}$	$A(z) + B(z)$
$\mathcal{A} \times \mathcal{B}$	$A(z) \times B(z)$
$\mathcal{A}'$	$A'(z)$
$\textsc{Seq}(\mathcal{C})$	$\frac{1}{1-C(z)}$
$\textsc{Cyc}(\mathcal{C})$	$\log \frac{1}{1-C(z)}$
$\textsc{Set}(\mathcal{C})$	$\exp(C(z))$

Examples

- Binary trees: $\mathcal{B} = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$;

$$B(z) = z + zB(z)^2$$

- General trees: $\mathcal{T} = \mathcal{Z} \times \textsc{Set}(\mathcal{T})$;

$$T(z) = z \exp(T(z))$$

Combinatorial Specifications

Language and Gen. Fcns (labelled)

$\mathcal{A} \cup \mathcal{B}$	$A(z) + B(z)$
$\mathcal{A} \times \mathcal{B}$	$A(z) \times B(z)$
$\mathcal{A}'$	$A'(z)$
$\textsc{Seq}(\mathcal{C})$	$\frac{1}{1-C(z)}$
$\textsc{Cyc}(\mathcal{C})$	$\log \frac{1}{1-C(z)}$
$\textsc{Set}(\mathcal{C})$	$\exp(C(z))$

Examples

- Binary trees: $\mathcal{B} = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$;

$$B(z) = z + zB(z)^2$$

- General trees: $\mathcal{T} = \mathcal{Z} \times \textsc{Set}(\mathcal{T})$;

$$T(z) = z \exp(T(z))$$

System (Σ): $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$

Definition (Well-founded)

$\boldsymbol{\mathcal{H}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$ and Jacobian $\partial\boldsymbol{\mathcal{H}}/\partial\boldsymbol{\mathcal{Y}}$ nilpotent at $(\varnothing, \boldsymbol{\varnothing})$.

Proposition

If (Σ) well-founded,

- $\boldsymbol{\mathcal{Y}}_{n+1} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}_n)$ *converges to a limit species $\boldsymbol{\mathcal{Y}}$ [Joyal]*
- $\mathbf{Y}(z)$ *is analytic in a neighborhood of 0.*

Combinatorial Specifications

Language and Gen. Fcns (unlabelled)

$\mathcal{A} \cup \mathcal{B}$	$A(z) + B(z)$
$\mathcal{A} \times \mathcal{B}$	$A(z) \times B(z)$
$\mathcal{A}'$	$A'(z)$
$\mathrm{SEQ}(\mathcal{C})$	$\frac{1}{1-C(z)}$
$\mathrm{PSET}(\mathrm{C})$	$\exp(\sum(-1)^i C(z^i)/i)$
$\mathrm{MSET}(\mathrm{C})$	$\exp(\sum C(z^i)/i)$
$\mathrm{CYC}(\mathrm{C})$	$\sum_{k\geq 1} \frac{\phi(k)}{k} \log \frac{1}{1-C(z^k)}$

Examples

- Binary trees: $\mathcal{B} = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$;

$$B(z) = z + zB(z)^2$$

- General trees: $\mathcal{T} = \mathcal{Z} \times \mathrm{SET}(\mathcal{T})$;

$$T(z) = z\exp(T(z) + T(z^2)/2 + \cdots)$$

System (Σ): $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$

Definition (Well-founded)

$\boldsymbol{\mathcal{H}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$ and Jacobian $\partial\boldsymbol{\mathcal{H}}/\partial\boldsymbol{\mathcal{Y}}$ nilpotent at $(\varnothing, \boldsymbol{\varnothing})$.

Proposition

If (Σ) well-founded,

- $\boldsymbol{\mathcal{Y}}_{n+1} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}_n)$ *converges to a limit species $\boldsymbol{\mathcal{Y}}$ [Joyal]*
- $\mathbf{Y}(z)$ *is analytic in a neighborhood of 0.*

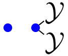
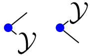
Combinatorial Newton (Décoste, Labelle, Leroux 1982)

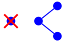
For one equation: $\mathcal{Y} = \mathcal{H}(\mathcal{Z},\mathcal{Y}) \to \mathcal{Y}_{n+1} = \mathcal{N}_{\mathcal{H}}(\mathcal{Y}_n)$

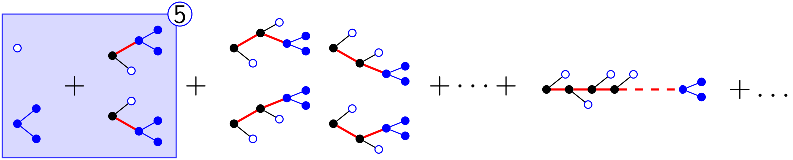
$$\mathcal{N}_{\mathcal{H}}(\mathcal{Y}) = \mathcal{Y} \cup \textsc{Seq}\left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z},\mathcal{Y})\right) \times (\mathcal{H}(\mathcal{Z},\mathcal{Y}) - \mathcal{Y}).$$

Contact doubles at each step.

Binary trees:

$\mathcal{H}(\mathcal{Z},\mathcal{Y}) = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{Y} \times \mathcal{Y} =$  $\quad \frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z},\mathcal{Y}) =$ 

$\mathcal{Y}_0 = \varnothing \qquad \mathcal{Y}_1 = \bullet \qquad \mathcal{H}(\mathcal{Y}_1) - \mathcal{Y}_1 = \mathcal{Z} + \mathcal{Z}\mathcal{Y}_1^2 - \mathcal{Y}_1 =$ 

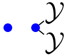
$\mathcal{Y}_2 =$ 

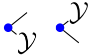
Combinatorial Newton (Décoste, Labelle, Leroux 1982)

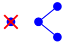
For one equation: $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) \rightarrow \mathcal{Y}_{n+1} = \mathcal{N}_{\mathcal{H}}(\mathcal{Y}_n)$

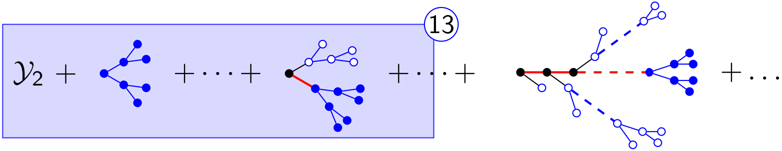
$$\mathcal{N}_{\mathcal{H}}(\mathcal{Y}) = \mathcal{Y} \cup \textsc{Seq}\left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{Y})\right) \times (\mathcal{H}(\mathcal{Z}, \mathcal{Y}) - \mathcal{Y}).$$

Binary trees:

$\mathcal{H}(\mathcal{Z}, \mathcal{Y}) = \mathcal{Z} \cup \mathcal{Z} \times \mathcal{Y} \times \mathcal{Y} =$ 

$\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{Y}) =$ 

$\mathcal{Y}_0 = \varnothing \qquad \mathcal{Y}_1 = \bullet \qquad \mathcal{H}(\mathcal{Y}_1) - \mathcal{Y}_1 = \mathcal{Z} + \mathcal{Z}\mathcal{Y}_1^2 - \mathcal{Y}_1 =$ 

$\mathcal{Y}_3 =$ 

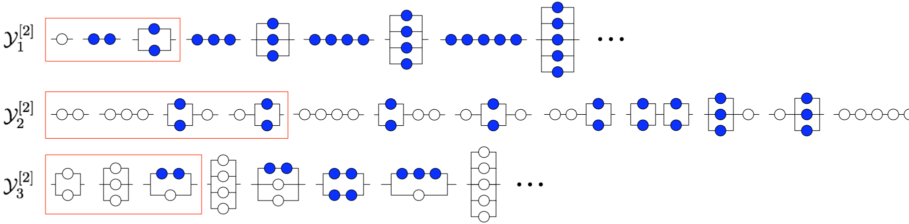
Combinatorial Newton (Décoste, Labelle, Leroux 1982)

For one equation: $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) \rightarrow \mathcal{Y}_{n+1} = \mathcal{N}_{\mathcal{H}}(\mathcal{Y}_n)$

$$\mathcal{N}_{\mathcal{H}}(\mathcal{Y}) = \mathcal{Y} \cup \textsc{Seq}\left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{Y})\right) \times (\mathcal{H}(\mathcal{Z}, \mathcal{Y}) - \mathcal{Y}).$$

Extension to systems (Pivoteau, Soria, Salvy 2008):

- $\textsc{Seq}(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}})$ replaced by $\boldsymbol{\mathcal{U}} = \cup_{k \geq 0} (\frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}})^k$ (Labelle's bloomings);



Combinatorial Newton (Décoste, Labelle, Leroux 1982)

For one equation: $\mathcal{Y} = \mathcal{H}(\mathcal{Z},\mathcal{Y}) \to \mathcal{Y}_{n+1} = \mathcal{N}_{\mathcal{H}}(\mathcal{Y}_n)$

$$\mathcal{N}_{\mathcal{H}}(\mathcal{Y}) = \mathcal{Y} \cup \textsc{Seq}\left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z},\mathcal{Y})\right) \times (\mathcal{H}(\mathcal{Z},\mathcal{Y}) - \mathcal{Y}).$$

Extension to systems (Pivoteau, Soria, Salvy 2008):

- $\textsc{Seq}(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}})$ replaced by $\boldsymbol{\mathcal{U}} = \cup_{k\geq 0}(\frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}})^k$ (Labelle's bloomings);
- this union $\boldsymbol{\mathcal{U}}$ itself is computed by Newton iteration:

$$\begin{aligned}
\boldsymbol{\mathcal{U}}_{n+1} &= \boldsymbol{\mathcal{U}}_n + \boldsymbol{\mathcal{U}}_n \boldsymbol{\mathcal{T}}_{n+1} \\
\boldsymbol{\mathcal{T}}_{n+1} &= \boldsymbol{\beta}_n \boldsymbol{\mathcal{U}}_n + \boldsymbol{\mathcal{T}}_n^2 \\
\boldsymbol{\beta}_n &= \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z},\boldsymbol{\mathcal{Y}}_n) - \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z},\boldsymbol{\mathcal{Y}}_{n-1})
\end{aligned}$$

At iteration $\boldsymbol{\mathcal{Y}}_n$, perform a single step of the calculation of $\boldsymbol{\mathcal{U}}$.

III Applications

Combinatorial Iteration → Iteration for Generating Series

- Same iteration in labelled & unlabelled universes:
  Example: General trees

$$\mathcal{Y} = \mathcal{H}(\mathcal{Y}) = \mathcal{Z} \times \textsc{Set}(\mathcal{Y}) \rightarrow \mathcal{N}_H(Y) = Y + \frac{H(Y) - Y}{1 - H(Y)}.$$

$$Y(z) = z \exp(Y(z) + Y(z^2)/2 + \cdots) \text{ at } z = 0.3 :$$

n	$Y_n(0.3)$	$Y_n(0.3^2)$	$Y_n(0.3^3)$
0	0	0	0
1	.43021322639	**0.99**370806338e-1	**0.277**59817516e-1
2	**.5**4875612912	**0.99887**132154e-1	**0.2777062918**7e-1
3	**.557**09557053	**0.9988714719**7e-1	**0.27770629189e-1**
4	**.55713907**945	**0.99887147198e-1**	**0.27770629189e-1**
5	**.55713908064**	**0.99887147198e-1**	**0.27770629189e-1**

Combinatorial Iteration → Iteration for Generating Series

- Same iteration in labelled & unlabelled universes:
  Example: General trees

$$\mathcal{Y} = \mathcal{H}(\mathcal{Y}) = \mathcal{Z} \times \textsc{Set}(\mathcal{Y}) \rightarrow \mathcal{N}_H(Y) = Y + \frac{H(Y) - Y}{1 - H(Y)}.$$

- Small arithmetic complexity:
  first N coefficients in $O(\mathrm{M}(\mathrm{N}))$ operations
  (M(N): cost of multiplying two series).
  → fast enumeration.

Combinatorial Iteration $\longrightarrow$ Iteration for Generating Series

- Same iteration in labelled & unlabelled universes:
  Example: General trees

$$\mathcal{Y} = \mathcal{H}(\mathcal{Y}) = \mathcal{Z} \times \textsc{Set}(\mathcal{Y}) \rightarrow \mathcal{N}_H(Y) = Y + \frac{H(Y) - Y}{1 - H(Y)}.$$

- Small arithmetic complexity:
  first N coefficients in $O(\mathrm{M(N)})$ operations
  (M(N): cost of multiplying two series).
  $\longrightarrow$ fast enumeration.
- Small bit complexity:
  exact enumeration in quasi-optimal time.

Numerical Values

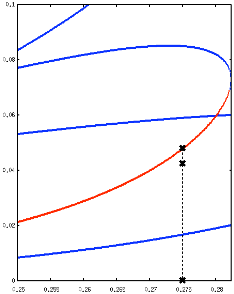
Combinatorics

- $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$
- $\mathcal{Y}_{n+1} = \mathcal{N}_{\mathcal{H}}(\mathcal{Z}, \mathcal{Y}_n)$
- $\varnothing = \mathcal{Y}_0 \subset \mathcal{Y}_1 \subset \cdots \subset \mathcal{Y}$

$$\Downarrow$$

Generating Series

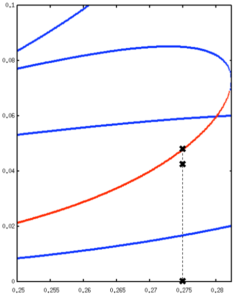
- $\mathbf{Y}(z) = \mathbf{H}(z, \mathbf{Y}(z))$
  (analytic in $|z| < r$ for some $r > 0$)
- $\mathbf{Y}_{n+1}(z) = \mathcal{N}_{\mathbf{H}}(z, \mathbf{Y}_n(z))$
- coeffs ≥ 0 and increasing $\Rightarrow$
  1. $\mathbf{Y}_n$ analytic in $|z| < r$;
  2. $\mathbf{Y}_n(x) \leq \mathbf{Y}_{n+1}(x)$ for $0 \leq x < r$;
  3. $\mathbf{Y}_n(z) \to \mathbf{Y}(z)$ for $|z| < r$.



Numerical Values

Generating Series

- $\mathbf{Y}(z) = \mathbf{H}(z, \mathbf{Y}(z))$
  (analytic in $|z| < r$ for some $r > 0$)
- $\mathbf{Y}_{n+1}(z) = \mathcal{N}_{\mathbf{H}}(z, \mathbf{Y}_n(z))$
- coeffs ≥ 0 and increasing $\Rightarrow$
  1. $\mathbf{Y}_n$ analytic in $|z| < r$;
  2. $\mathbf{Y}_n(x) \leq \mathbf{Y}_{n+1}(x)$ for $0 \leq x < r$;
  3. $\mathbf{Y}_n(z) \to \mathbf{Y}(z)$ for $|z| < r$.



$\Downarrow$

Proposition

For $0 \leq x < r$, define
$\alpha_0 := 0$ *and* $\boldsymbol{\alpha}_{n+1} := \mathcal{N}_{\mathbf{H}}(x, \boldsymbol{\alpha}_n)$.
Then, $\boldsymbol{\alpha}_n = \mathbf{Y}_n(x) \to \mathbf{Y}(x)$.

(Technical point: prove that **H** is analytic there.)

IV Conclusion

Conclusions

- More info at Carine Pivoteau's defense next Wednesday at LIP6;
- Next: use Schröder iteration at $x = r$ (exploiting a result of Drmota-Lalley-Woods);
- Next: singularity analysis of these systems.