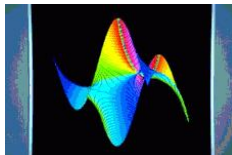


Combinatorial Newton Iteration and Alea

Bruno Salvy
`Bruno.Salvy@inria.fr`

Algorithms Project, Inria

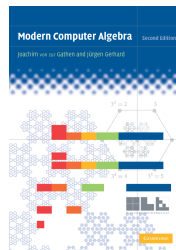
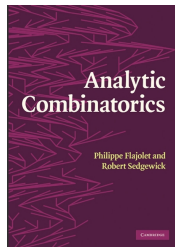
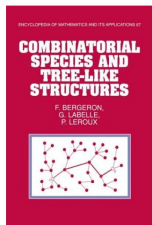


Joint work with Carine Pivoteau and Michèle Soria

Alea, March 9, 2011

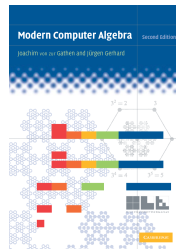
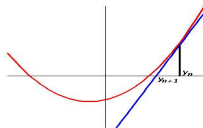
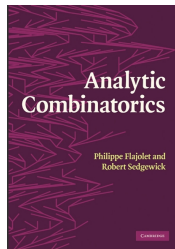
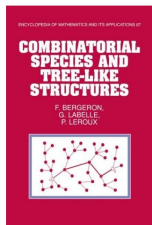
I Introduction

Context



Aim: Algorithms for analytic combinatorics.
Well-defined input provided by species theory.
Efficiency by computer algebra.

Context



Aim: Algorithms for analytic combinatorics.
Well-defined input provided by species theory.
Efficiency by computer algebra.

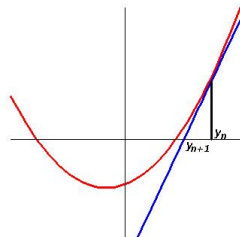
Main Tool: Newton Iteration

To solve $\phi(y) = 0$, iterate

$$y^{[n+1]} = y^{[n]} - u^{[n+1]}, \quad \phi'(y^{[n]})u^{[n+1]} = \phi(y^{[n]}).$$

Good case: **quadratic** convergence if

- the initial point is close enough;
- the root is simple.



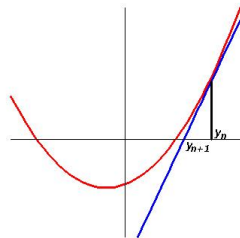
Main Tool: Newton Iteration

To solve $\phi(y) = 0$, iterate

$$y^{[n+1]} = y^{[n]} - u^{[n+1]}, \quad \phi'(y^{[n]})u^{[n+1]} = \phi(y^{[n]}).$$

Good case: **quadratic** convergence if

- the initial point is close enough;
- the root is simple.



Proof: simple root at ζ

$$\left. \begin{aligned} \phi(y^{[n]}) &= \phi'(\zeta)(y^{[n]} - \zeta) + O((y^{[n]} - \zeta)^2) \\ \phi'(y^{[n]}) &= \phi'(\zeta) + O(y^{[n]} - \zeta) \end{aligned} \right\} \Rightarrow y^{[n]} - \zeta = \frac{\phi(y^{[n]})}{\phi'(y^{[n]})} + O((y^{[n]} - \zeta)^2)$$

Examples of Quadratic Convergence

$$\phi(y) = 1 + \frac{1}{8}y^2 - y$$

$$y^{[n+1]} = y^{[n]} + \frac{1 + y^{[n]2}/8 - y^{[n]}}{1 - y^{[n]}/4}$$

$$y^{[0]} = 0.$$

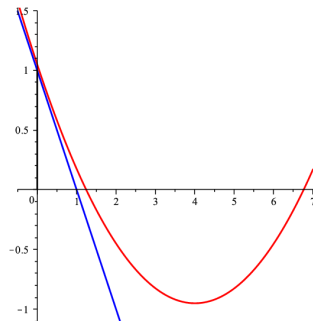
$$y^{[1]} = 1.000000000000000000000000000000$$

$$y^{[2]} = 1.166666666666666666666666666666$$

$$y^{[3]} = 1.17156862745098039215686275$$

$$y^{[4]} = 1.17157287525062017874740884$$

$$y^{[5]} = 1.17157287525380990239662075$$



Examples of Quadratic Convergence

$$\phi(y) = 1 + zy^2 - y$$

$$y^{[n+1]} = y_n + \frac{1 + zy^{[n]2} - y_n}{1 - 2zy^{[n]}}$$

$$y^{[0]} = 0$$

$$y^{[1]} = 1$$

$$y^{[2]} = 1 + z + 2z^2 + 4z^3 + 8z^4 + 16z^5 + 32z^6 + 64z^7 + \dots$$

$$y^{[3]} = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6 + 428z^7 + \dots$$

[Newton 1671]

$y^3 + a^2y - 2a^3 + a^2xy - x^3 = 0. \quad y = a - \frac{x}{4} + \frac{x^2}{64a} + \frac{111x^3}{512a^2} + \frac{509x^4}{16184a^3} \&c.$

$+a + p = y.$	$+y^3$ $+axy$ $+x^2y$ $-x^3$ $-2a^3$	$+a^3 + 3a^2p + 3ap^2 + p^3$ $+a^2x + axp$ $+x^3 + a^2p$ $-x^3$ $-2a^3$
$-\frac{1}{4}x + q = p.$	$+p^3$ $+3ap^2$ $+axp$ $+4a^2p$ $+a^2x$ $-x^3$	$-\frac{1}{4}x^3 + \frac{1}{16}x^2q - \frac{1}{4}xq^2 + q^3$ $+\frac{1}{16}ax^2 - \frac{1}{4}axq + 3aq^2$ $-\frac{1}{4}ax^2 + axq$ $-a^2x + 4a^2q$ $+a^2x$ $-x^3$
$+\frac{x^2}{64a} + r = q.$	$+q^3$ $-\frac{1}{4}xq^2$ $+3aq^2$ $+\frac{1}{16}x^2q$ $-\frac{1}{4}axq$ $+4a^2q$ $-\frac{6}{64}x^3$ $-\frac{1}{16}ax^2$	$*$ $*$ $+\frac{1x^4}{4096a} * + \frac{1}{16}x^2r + 3ar^2$ $+ \frac{1x^4}{1024a} * + \frac{1}{16}x^2r$ $-\frac{1}{128}x^3 - \frac{1}{4}axr$ $+\frac{1}{16}ax^2 + 4a^2r$ $-\frac{6}{64}x^3$ $-\frac{1}{16}ax^2$
$* + 4a^2 - \frac{1}{4}ax + \frac{1}{16}x^2 + \frac{111x^3}{512a^2} - \frac{15x^4}{4096a} \left(+ \frac{111x^3}{512a^2} + \frac{509x^4}{16184a^3} \right)$		

Examples of Quadratic Convergence

$$\phi(\mathcal{Y}) = (\mathcal{E} + \mathcal{Z} \cdot \mathcal{Y}^2) \setminus \mathcal{Y}$$

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \text{SEQ}(\mathcal{Z} \cdot \mathcal{Y}^{[n]} \cdot \square + \mathcal{Z} \cdot \square \cdot \mathcal{Y}^{[n]}) \cdot \phi(\mathcal{Y}^{[n]}).$$

$$\mathcal{Y}^{[0]} = \emptyset \quad \mathcal{Y}^{[1]} = \circ$$

$$\mathcal{Y}^{[2]} = \begin{array}{c} \text{Diagram 1} \\ \text{Diagram 2} \end{array} + \begin{array}{c} \text{Diagram 3} \\ \text{Diagram 4} \end{array} + \dots + \begin{array}{c} \text{Diagram 5} \end{array} + \dots$$

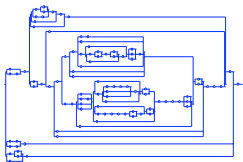
The diagram for $\mathcal{Y}^{[2]}$ shows a sequence of terms. The first term is a blue-shaded box containing two diagrams: one with a single white node and another with a black node connected to a blue node, which is then connected to two white nodes. A circled '2' is next to this box. The second term is a diagram with a black node connected to a blue node, which is then connected to two white nodes. The third term is a diagram with a black node connected to a blue node, which is then connected to two white nodes. The fourth term is a diagram with a black node connected to a blue node, which is then connected to two white nodes. The sequence continues with an ellipsis.

$$\mathcal{Y}^{[3]} = \mathcal{Y}^{[2]} + \begin{array}{c} \text{Diagram 6} \end{array} + \dots + \begin{array}{c} \text{Diagram 7} \end{array} + \dots$$

The diagram for $\mathcal{Y}^{[3]}$ shows a sequence of terms. The first term is a blue-shaded box containing the entire $\mathcal{Y}^{[2]}$ sequence. A circled '6' is next to this box. The second term is a diagram with a black node connected to a blue node, which is then connected to two white nodes. The third term is a diagram with a black node connected to a blue node, which is then connected to two white nodes. The sequence continues with an ellipsis.

[Décoste, Labelle, Leroux 1982]

Motivation: Random Generation / Discrete Simulation



$$\mathcal{G} = \mathcal{S} + \mathcal{P},$$

$$\mathcal{S} = \text{Seq}(\mathcal{Z} + \mathcal{P}, \text{card} > 1),$$

$$\mathcal{P} = \text{Set}(\mathcal{Z} + \mathcal{S}, \text{card} > 0).$$

Definition (Generating function)

$$Y(z) = y_0 + y_1 z + \cdots + y_n z^n (/n!) + \cdots \quad (y_n: \text{nb objects of size } n).$$

Algorithms for **uniform random generation** in size N use either

- $y_0, \dots, y_N$ (recursive method);
- $Y(x)$ for some $x > 0$ (Boltzmann sampler).

[Nijenhuis, Wilf 1978; Flajolet, Zimmermann, Van Cutsem 1994; Duchon, Flajolet, Louchard, Schaeffer 2004; Flajolet, Fusy, Pivoteau 2007]

Combinatorial Newton Iteration

Well-Founded Combinatorial Specification



Combinatorial Newton iteration for $\mathcal{Y}$

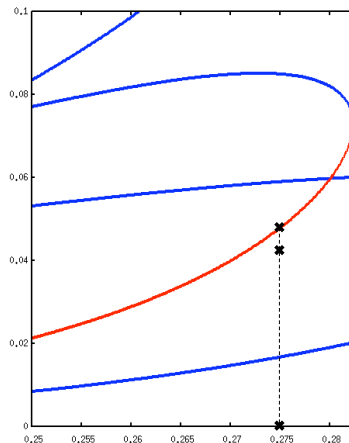


Newton iteration for the gf $Y(z)$

$(y_0, \dots, y_N)$ fast



Numerical Newton iteration starting from 0 converges to the value of $Y(x)$.



[Labelle *et alii* 80-90] combinatorial part, without $\mathcal{E}$, using species theory

[Pivoteau, Salvy, Soria 2008] labelled case, without $\mathcal{E}$

Today: General case, using species theory

Combinatorial Newton Iteration

Well-Founded Combinatorial Specification



Combinatorial Newton iteration for $\mathcal{Y}$

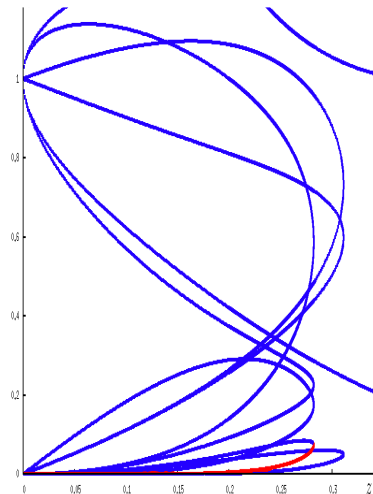


Newton iteration for the gf $Y(z)$

$(y_0, \dots, y_N)$ **fast**



Numerical Newton iteration **starting from 0** converges to the value of $Y(x)$.



[Labelle *et alii* 80-90] combinatorial part, without $\mathcal{E}$, using species theory

[Pivoteau, Salvy, Soria 2008] labelled case, without $\mathcal{E}$

Today: General case, using species theory

Examples of Polynomial Systems

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	

[Darrasse 2008]

Results (1/3): Well-founded Systems

Definition ($\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$ *well-founded*)

The iteration $\mathcal{Y}^{[n+1]} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}^{[n]})$, $\mathcal{Y}^{[0]} = 0$ is defined, converges, and none of the coordinates of the limit is 0.

Theorem (General Implicit Species Theorem)

Well-founded iff condition that is easily checked on $\mathcal{H}$.

Results (2/3): Fast Enumeration

Definition (Constructible Species)

$1, \mathcal{Z}, +, \cdot, \text{SEQ}, \text{CYC}, \text{SET}, \mathcal{Y}_1, \mathcal{Y}_2, \dots$; cardinality constraints that are finite unions of intervals; composition of constructible; solutions of well-founded systems with constructible coordinates.

Theorem (Enumeration in Quasi-Optimal Complexity)

First N coefficients of gfs of constructible species in

- ① *arithmetic complexity:*
 - $O(\textcolor{red}{N} \log N)$ (both ogf and egf);
- ② *binary complexity:*
 - $O(\textcolor{red}{N}^2 \log^2 N \log \log N)$ (ogf);
 - $O(\textcolor{red}{N}^2 \log^3 N \log \log N)$ (egf).

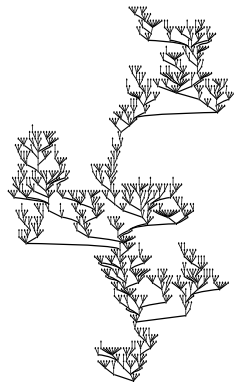
Results (3/3): Oracle

- 1 The ogfs of constructible species are convergent in the neighborhood of 0;
- 2 A numerical iteration converging to the sequence $\mathbf{Y}(\alpha), \mathbf{Y}(\alpha^2), \mathbf{Y}(\alpha^3), \dots$ for $\|\cdot\|_\infty$.



Example: Unlabelled Rooted Trees

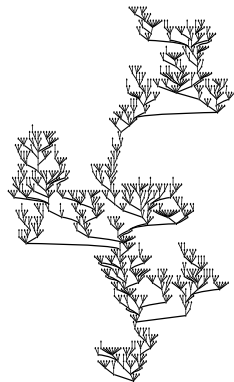
- ① Well-founded system: $\mathcal{Y} = \mathcal{Z} \cdot \text{SET}(\mathcal{Y}) =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$;



Example: Unlabelled Rooted Trees

- ① Well-founded system: $\mathcal{Y} = \mathcal{Z} \cdot \text{SET}(\mathcal{Y}) =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$;
- ② Combinatorial Newton iteration:

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \text{SEQ}(\mathcal{H}(\mathcal{Y}^{[n]})) \cdot (\mathcal{H}(\mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]})$$

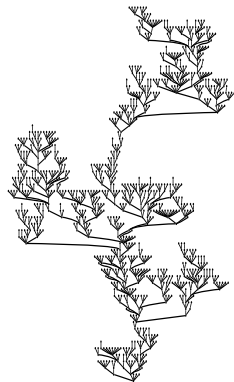


Example: Unlabelled Rooted Trees

- ① Well-founded system: $\mathcal{Y} = \mathcal{Z} \cdot \text{SET}(\mathcal{Y}) =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$;
- ② Combinatorial Newton iteration:

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \text{SEQ}(\mathcal{H}(\mathcal{Y}^{[n]})) \cdot (\mathcal{H}(\mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]})$$
- ③ OGF equation: $\tilde{Y}(z) = H(z, \tilde{Y}(z))$

$$= z \exp(\tilde{Y}(z) + \frac{1}{2} \tilde{Y}(z^2) + \frac{1}{3} \tilde{Y}(z^3) + \dots)$$



Example: Unlabelled Rooted Trees

① Well-founded system: $\mathcal{Y} = \mathcal{Z} \cdot \text{SET}(\mathcal{Y}) =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$;

② Combinatorial Newton iteration:

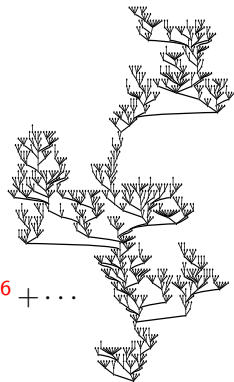
$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \text{SEQ}(\mathcal{H}(\mathcal{Y}^{[n]})) \cdot (\mathcal{H}(\mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]})$$

③ OGF equation: $\tilde{Y}(z) = H(z, \tilde{Y}(z))$

$$= z \exp(\tilde{Y}(z) + \frac{1}{2} \tilde{Y}(z^2) + \frac{1}{3} \tilde{Y}(z^3) + \dots)$$

④ Newton for OGF: $\tilde{Y}^{[n+1]} = \tilde{Y}^{[n]} + \frac{H(z, \tilde{Y}^{[n]}) - \tilde{Y}^{[n]}}{1 - H(z, \tilde{Y}^{[n]})}$

$$0, z + z^2 + z^3 + z^4 + \dots, z + z^2 + 2z^3 + 4z^4 + 9z^5 + 20z^6 + \dots$$



Example: Unlabelled Rooted Trees

① Well-founded system: $\mathcal{Y} = \mathcal{Z} \cdot \text{SET}(\mathcal{Y}) =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$;

② Combinatorial Newton iteration:

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \text{SEQ}(\mathcal{H}(\mathcal{Y}^{[n]})) \cdot (\mathcal{H}(\mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]})$$

③ OGF equation: $\tilde{Y}(z) = H(z, \tilde{Y}(z))$

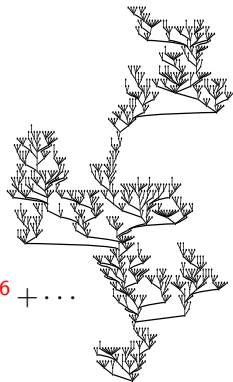
$$= z \exp(\tilde{Y}(z) + \frac{1}{2} \tilde{Y}(z^2) + \frac{1}{3} \tilde{Y}(z^3) + \dots)$$

④ Newton for OGF: $\tilde{Y}^{[n+1]} = \tilde{Y}^{[n]} + \frac{H(z, \tilde{Y}^{[n]}) - \tilde{Y}^{[n]}}{1 - H(z, \tilde{Y}^{[n]})}$

$$0, z + z^2 + z^3 + z^4 + \dots, z + z^2 + 2z^3 + 4z^4 + 9z^5 + 20z^6 + \dots$$

⑤ Numerical iteration:

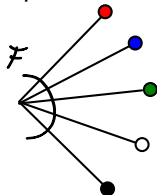
n	$\tilde{Y}^{[n]}(0.3)$	$\tilde{Y}^{[n]}(0.3^2)$	$\tilde{Y}^{[n]}(0.3^3)$
0	0	0	0
1	.43021322639	0.99370806338e-1	0.27759817516e-1
2	.54875612912	0.99887132154e-1	0.27770629187e-1
3	.55709557053	0.99887147197e-1	0.27770629189e-1
4	.55713907945	0.99887147198e-1	0.27770629189e-1
5	.55713908064	0.99887147198e-1	0.27770629189e-1



II Combinatorics

Mini-Introduction to Species

- Species $\mathcal{F}$:

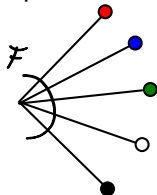


Examples:

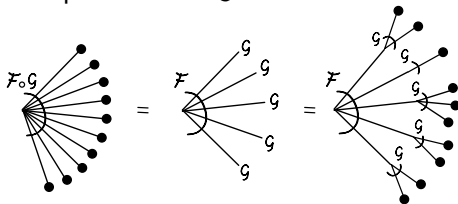
- $0, 1, \mathbb{Z}$;
- SET;
- SEQ, CYC.

Mini-Introduction to Species

- Species $\mathcal{F}$:



- Composition $\mathcal{F} \circ \mathcal{G}$:

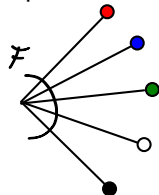


Examples:

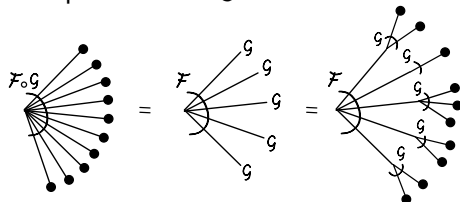
- $0, 1, \mathbb{Z}$;
- SET;
- SEQ, CYC.

Mini-Introduction to Species

- Species $\mathcal{F}$:



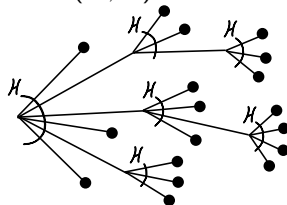
- Composition $\mathcal{F} \circ \mathcal{G}$:



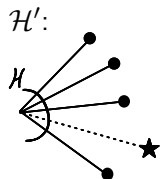
Examples:

- 0, 1, $\mathbb{Z}$;
- SET;
- SEQ, CYC.

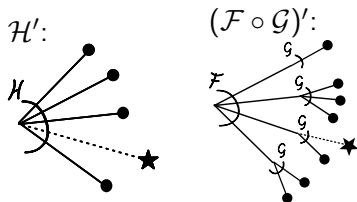
- $\mathcal{Y} = \mathcal{H}(\mathbb{Z}, \mathcal{Y})$



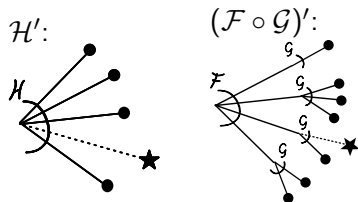
Derivative



Derivative

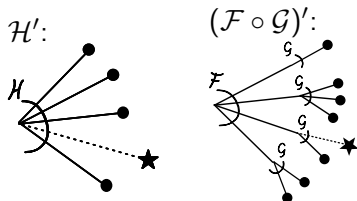


Derivative



species	derivative
$A + B$	$A' + B'$
$A \cdot B$	$A' \cdot B + A \cdot B'$
$\text{SEQ}(B)$	$\text{SEQ}(B) \cdot B' \cdot \text{SEQ}(B)$
$\text{CYC}(B)$	$\text{SEQ}(B) \cdot B'$
$\text{SET}(B)$	$\text{SET}(B) \cdot B'$

Derivative



Example:

species	derivative
$\mathcal{A} + \mathcal{B}$	$\mathcal{A}' + \mathcal{B}'$
$\mathcal{A} \cdot \mathcal{B}$	$\mathcal{A}' \cdot \mathcal{B} + \mathcal{A} \cdot \mathcal{B}'$
$\text{SEQ}(\mathcal{B})$	$\text{SEQ}(\mathcal{B}) \cdot \mathcal{B}' \cdot \text{SEQ}(\mathcal{B})$
$\text{CYC}(\mathcal{B})$	$\text{SEQ}(\mathcal{B}) \cdot \mathcal{B}'$
$\text{SET}(\mathcal{B})$	$\text{SET}(\mathcal{B}) \cdot \mathcal{B}'$

$$\mathcal{H}(\mathcal{G}, \mathcal{S}, \mathcal{P}) := (\mathcal{S} + \mathcal{P}, \text{Seq}(\mathcal{Z} + \mathcal{P}), \text{Set}(\mathcal{Z} + \mathcal{S})).$$

$$\frac{\partial \mathcal{H}}{\partial \mathcal{Y}} = \begin{pmatrix} \emptyset & \mathcal{E} & \mathcal{E} \\ \emptyset & \emptyset & \text{Seq}(\mathcal{Z} + \mathcal{P}) \cdot \mathcal{E} \cdot \text{Seq}(\mathcal{Z} + \mathcal{P}) \\ \emptyset & \text{Set}(\mathcal{Z} + \mathcal{S}) \cdot \mathcal{E} & \emptyset \end{pmatrix}$$

Generating Series

Not a Definition: Cycle Index Series

$$Z_{\mathcal{F}}(z_1, z_2, z_3, \dots).$$

Main results of the theory

- ① EGF: $F(z) = Z_{\mathcal{F}}(z, 0, 0, \dots)$
- ② OGF: $\tilde{F}(z) = Z_{\mathcal{F}}(z, z^2, z^3, \dots)$
- ③ Composition: $Z_{\mathcal{F} \circ \mathcal{G}}(z_1, z_2, z_3, \dots) = Z_{\mathcal{F}}(Z_{\mathcal{G}}(z_1, z_2, z_3, \dots), Z_{\mathcal{G}}(z_2, z_4, z_6, \dots), Z_{\mathcal{G}}(z_3, z_6, z_9, \dots), \dots)$
- ④ Derivative $Z_{\mathcal{F}'} = \frac{\partial}{\partial z_1} Z_{\mathcal{F}}$.

Consequences:

- ① EGFs easy;
- ② Pólya operators for ogfs;
- ③ Newton iteration more difficult for ogfs.

Joyal's Implicit Species Theorem

Theorem

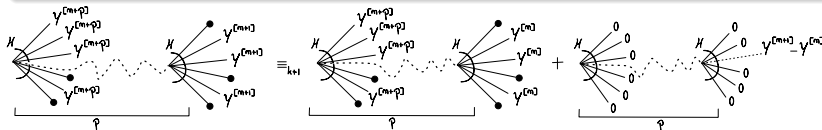
If $\mathcal{H}(0, 0) = 0$ and $\partial\mathcal{H}/\partial\mathcal{Y}(0, 0)$ is nilpotent, then $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$ has a unique solution, limit of

$$\mathcal{Y}^{[0]} = 0, \quad \mathcal{Y}^{[n+1]} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}^{[n]}) \quad (n \geq 0).$$

Def. $\mathcal{A} =_k \mathcal{B}$ if they coincide up to size k (contact k).

Key Lemma

If $\mathcal{Y}^{[n+1]} =_k \mathcal{Y}^{[n]}$, then $\mathcal{Y}^{[n+p+1]} =_{k+1} \mathcal{Y}^{[n+p]}$, ($p = \text{dimension}$).



Joyal's Implicit Species Theorem

Theorem

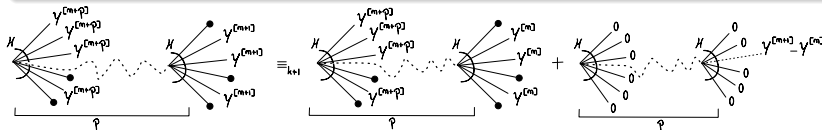
If $\mathcal{H}(0,0) = 0$ and $\partial\mathcal{H}/\partial\mathcal{Y}(0,0)$ is nilpotent, then $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$ has a unique solution, limit of

$$\mathcal{Y}^{[0]} = 0, \quad \mathcal{Y}^{[n+1]} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}^{[n]}) \quad (n \geq 0).$$

Def. $\mathcal{A} =_k \mathcal{B}$ if they coincide up to size k (contact k).

Key Lemma

If $\mathcal{Y}^{[n+1]} =_k \mathcal{Y}^{[n]}$, then $\mathcal{Y}^{[n+p+1]} =_{k+1} \mathcal{Y}^{[n+p]}$, ($p = \text{dimension}$).



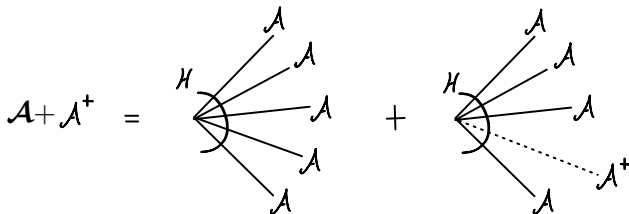
Combinatorial Newton Iteration

Theorem (essentially Labelle)

For any well-founded system $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$, if $\mathcal{A}$ has contact k with the solution and $\mathcal{A} \subset \mathcal{H}(\mathcal{Z}, \mathcal{A})$, then

$$\mathcal{A} + \sum_{i \geq 0} \left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{A}) \right)^i \cdot (\mathcal{H}(\mathcal{Z}, \mathcal{A}) - \mathcal{A})$$

has contact $2k + 1$ with it.



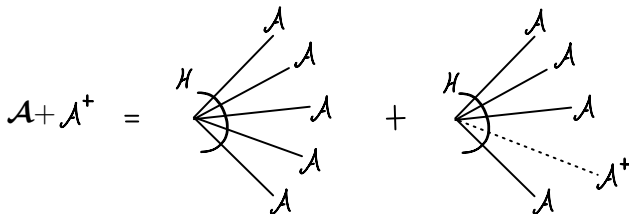
Combinatorial Newton Iteration

Theorem (essentially Labelle)

For any well-founded system $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$, if $\mathcal{A}$ has contact k with the solution and $\mathcal{A} \subset \mathcal{H}(\mathcal{Z}, \mathcal{A})$, then

$$\mathcal{A} + \sum_{i \geq 0} \left(\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{A}) \right)^i \cdot (\mathcal{H}(\mathcal{Z}, \mathcal{A}) - \mathcal{A})$$

has contact $2k + 1$ with it.



Generation by increasing Strahler numbers.

III Newton Iteration: Fast Algorithms

Basic Idea: Newton Iteration has Good Complexity

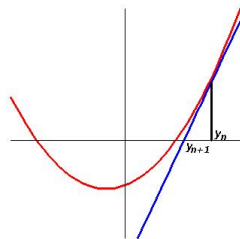
To solve $\phi(y) = 0$, iterate

$$y^{[n+1]} = y^{[n]} - u^{[n+1]}, \quad \phi'(y^{[n]})u^{[n+1]} = \phi(y^{[n]}).$$

Quadratic convergence



Divide-and-Conquer



Basic Idea: Newton Iteration has Good Complexity

To solve $\phi(y) = 0$, iterate

$$y^{[n+1]} = y^{[n]} - u^{[n+1]}, \quad \phi'(y^{[n]})u^{[n+1]} = \phi(y^{[n]}).$$

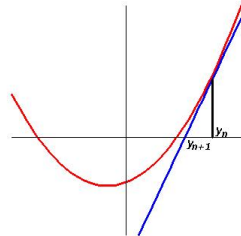
Quadratic convergence



Divide-and-Conquer

To solve at precision N

- ① Solve at precision $N/2$;
- ② Compute ϕ and ϕ' there;
- ③ Solve for $u^{[n+1]}$.



$$\text{Cost}(y^{[n]}) = \text{constant} \times \text{Cost}(\text{last step}).$$

Basic Idea: Newton Iteration has Good Complexity

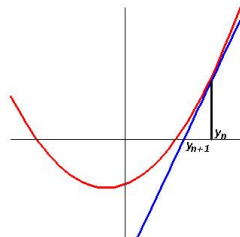
To solve $\phi(y) = 0$, iterate

$$y^{[n+1]} = y^{[n]} - u^{[n+1]}, \quad \phi'(y^{[n]})u^{[n+1]} = \phi(y^{[n]}).$$

Quadratic convergence



Divide-and-Conquer



To solve at precision N

- 1 Solve at precision $N/2$;
- 2 Compute ϕ and ϕ' there;
- 3 Solve for $u^{[n+1]}$.

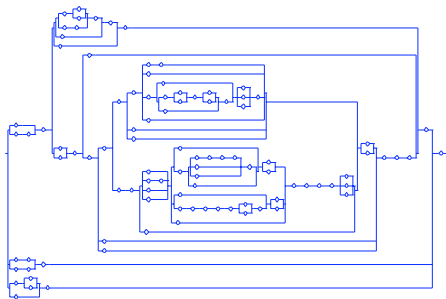
$$\text{Cost}(y^{[n]}) = \text{constant} \times \text{Cost}(\text{last step}).$$

Useful in conjunction with **fast multiplication** (quasi-linear):

- power series at order N : $O(N \log N)$ ops on the coefficients;
- N -bit integers: $O(N \log N \log \log N)$ bit ops.

Example

$$\begin{cases} \mathcal{G} &= \mathcal{S} + \mathcal{P}, \\ \mathcal{S} &= \text{Seq}_{>0}(\mathcal{Z} + \mathcal{P}), \\ \mathcal{P} &= \text{Set}_{>0}(\mathcal{Z} + \mathcal{S}). \end{cases} \quad \frac{\partial \mathcal{H}}{\partial \mathbf{y}} = \begin{pmatrix} \emptyset & \mathcal{E} & \mathcal{E} \\ \emptyset & \emptyset & \text{Seq}^2(\mathcal{Z} + \mathcal{P}) \\ \emptyset & \text{Set}(\mathcal{Z} + \mathcal{S}) & \emptyset \end{pmatrix}$$



Example

$$\begin{cases} \mathcal{G} &= \mathcal{S} + \mathcal{P}, \\ \mathcal{S} &= \text{Seq}_{>0}(\mathcal{Z} + \mathcal{P}), \\ \mathcal{P} &= \text{Set}_{>0}(\mathcal{Z} + \mathcal{S}). \end{cases} \quad \frac{\partial \mathcal{H}}{\partial \mathbf{Y}} = \begin{pmatrix} \emptyset & \mathcal{E} & \mathcal{E} \\ \emptyset & \emptyset & \text{Seq}^2(\mathcal{Z} + \mathcal{P}) \\ \emptyset & \text{Set}(\mathcal{Z} + \mathcal{S}) & \emptyset \end{pmatrix}$$

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration: $\mathbf{Y}^{[n]} := \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix},$

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + \left(\text{Id} - \frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \right)^{-1} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right).$$

Example

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration: $\mathbf{Y}^{[n]} := \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix},$

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + \left(\text{Id} - \frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \right)^{-1} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right).$$

Example

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration: $\mathbf{Y}^{[n]} := \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix},$

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + \left(\text{Id} - \frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \right)^{-1} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right) \bmod z^{2^{n+1}}.$$

Example

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration: $\mathbf{Y}^{[n]} := \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix},$

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + \left(\text{Id} - \frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \right)^{-1} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right) \bmod z^{2^{n+1}}.$$

$\Rightarrow$ **Wanted:** efficient matrix inverse, efficient exp.

Newton Iteration for Inverses

$$\phi(y) = a - 1/y \Rightarrow 1/\phi'(y) = y^2 \Rightarrow \boxed{y^{[n+1]} = y^{[n]} - y^{[n]}(ay^{[n]} - 1)}.$$

Cost: a small number of multiplications

Works for:

- ① Numerical inversion;
- ② Reciprocal of power series;
- ③ Inversion of matrices.

Applications:

- SEQ
- $(I - \frac{\partial H}{\partial Y})^{-1}$

[Schulz 1933; Cook 1966; Sieveking 1972; Kung 1974]

Example

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration:

$$\begin{cases} U^{[n+1]} &= U^{[n]} + U^{[n]} \cdot \left(\frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \cdot U^{[n]} + \text{Id} - U^{[n]} \right) \bmod z^{2^n}, \\ \mathbf{Y}^{[n+1]} &= \mathbf{Y}^{[n]} + U^{[n+1]} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right) \bmod z^{2^{n+1}}. \end{cases}$$

Can be lifted combinatorially.

Also a numerical iteration!

Example

$$\begin{cases} G &= S + P, \\ S &= (1 - z - P)^{-1}, \\ P &= \exp(z + S) - 1. \end{cases} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & (1 - z - P)^{-2} \\ 0 & \exp(z + S) & 0 \end{pmatrix}$$

Newton iteration:

$$\begin{cases} U^{[n+1]} &= U^{[n]} + U^{[n]} \cdot \left(\frac{\partial \mathbf{H}}{\partial \mathbf{Y}}(\mathbf{Y}^{[n]}) \cdot U^{[n]} + \text{Id} - U^{[n]} \right) \bmod z^{2^n}, \\ \mathbf{Y}^{[n+1]} &= \mathbf{Y}^{[n]} + U^{[n+1]} \cdot \left(\mathbf{H}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]} \right) \bmod z^{2^{n+1}}. \end{cases}$$

Can be lifted combinatorially.

Also a numerical iteration!

$\Rightarrow$ **Wanted:** efficient exp.

From the Inverse to the Exponential

- ① Logarithm of power series: $\log f = \int (f'/f)$;
- ② exponential of power series: $\phi(y) = a - \log y$.

$$\begin{aligned} e^{[n+1]} &= e^{[n]} + \frac{a - \log e^{[n]}}{1/e^{[n]}} \bmod z^{2^{n+1}}, \\ &= e^{[n]} + e^{[n]} \left(a - \int e^{[n]}'/e^{[n]} \right) \bmod z^{2^{n+1}}. \end{aligned}$$

And $1/e^{[n]}$ is computed by Newton iteration too!

[Brent 1975]

Example

$$\mathcal{G} = \mathcal{S} + \mathcal{P}, \quad \mathcal{S} = \text{SEQ}_{>0}(\mathcal{Z} + \mathcal{P}), \quad \mathcal{P} = \text{SET}_{>0}(\mathcal{Z} + \mathcal{S})$$

compiles into the Newton iteration:

$$\left\{ \begin{array}{l} i^{[n+1]} = i^{[n]} - i^{[n]}(e^{[n]}i^{[n]} - 1), \\ e^{[n+1]} = e^{[n]} - e^{[n]} \left(1 + \frac{d}{dz} S^{[n]} - \int \left(\frac{d}{dz} e^{[n]} \right) i^{[n]} \right), \\ v^{[n+1]} = v^{[n]} - v^{[n]}((1 - z - P^{[n]})v^{[n]} - 1), \\ U^{[n+1]} = U^{[n]} + U^{[n]} \cdot \left(\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & v^{[n+1]^2} \\ 0 & e^{[n+1]} & 0 \end{pmatrix} \cdot U^{[n]} + \text{Id} - U^{[n]} \right), \\ \begin{pmatrix} G^{[n+1]} \\ S^{[n+1]} \\ P^{[n+1]} \end{pmatrix} = \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix} + U^{[n+1]} \cdot \begin{pmatrix} S^{[n]} + P^{[n]} - G^{[n]} \\ v^{[n+1]} - S^{[n]} \\ e^{[n+1]} - P^{[n]} \end{pmatrix} \bmod z^{2^{n+1}}. \end{array} \right.$$

Computation reduced to products and linear ops.

Application: Power Sums

$$F = t^N + a_{N-1}t^{N-1} + \cdots + a_0 \leftrightarrow S_i = \sum_{F(\alpha)=0} \alpha^i, \quad i = 0, \dots, N.$$

Application: Power Sums

$$F = t^N + a_{N-1}t^{N-1} + \cdots + a_0 \leftrightarrow S_i = \sum_{F(\alpha)=0} \alpha^i, \quad i = 0, \dots, N.$$

Fast conversion using the generating series:

$$\frac{\text{rev}(F)'}{\text{rev}(F)} = - \sum_{i \geq 0} S_{i+1} t^i \leftrightarrow \text{rev}(F) = \exp \left(- \sum \frac{S_i}{i} t^i \right).$$

Application: Power Sums

$$F = t^N + a_{N-1}t^{N-1} + \cdots + a_0 \leftrightarrow S_i = \sum_{F(\alpha)=0} \alpha^i, \quad i = 0, \dots, N.$$

Fast conversion using the generating series:

$$\frac{\text{rev}(F)'}{\text{rev}(F)} = - \sum_{i \geq 0} S_{i+1} t^i \leftrightarrow \text{rev}(F) = \exp \left(- \sum \frac{S_i}{i} t^i \right).$$

Application: composed product and sums

$$(F, G) \mapsto \prod_{F(\alpha)=0, G(\beta)=0} (t - \alpha\beta) \quad \text{or} \quad \prod_{F(\alpha)=0, G(\beta)=0} (t - (\alpha + \beta)).$$

Application: Power Sums

$$F = t^N + a_{N-1}t^{N-1} + \cdots + a_0 \leftrightarrow S_i = \sum_{F(\alpha)=0} \alpha^i, \quad i = 0, \dots, N.$$

Fast conversion using the generating series:

$$\frac{\text{rev}(F)'}{\text{rev}(F)} = - \sum_{i \geq 0} S_{i+1} t^i \leftrightarrow \text{rev}(F) = \exp \left(- \sum \frac{S_i}{i} t^i \right).$$

Application: composed product and sums

$$(F, G) \mapsto \prod_{F(\alpha)=0, G(\beta)=0} (t - \alpha\beta) \quad \text{or} \quad \prod_{F(\alpha)=0, G(\beta)=0} (t - (\alpha + \beta)).$$

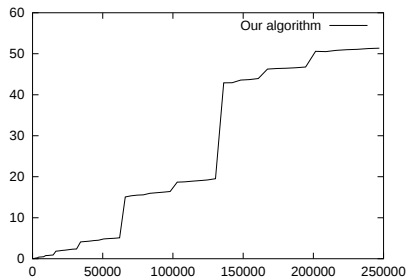
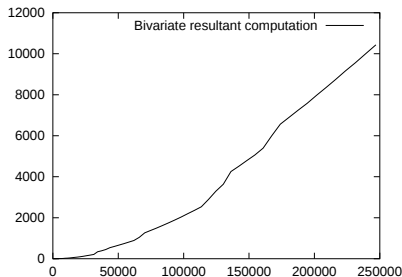
Easy in Newton representation: $\sum \alpha^s \sum \beta^s = \sum (\alpha\beta)^s$ and

$$\sum \frac{\sum (\alpha + \beta)^s}{s!} t^s = \left(\sum \frac{\sum \alpha^s}{s!} t^s \right) \left(\sum \frac{\sum \beta^s}{s!} t^s \right).$$

[Schönhage 1982; Bostan, Flajolet, Salvy, Schost 2006]

Timings

Applications (crypto): over finite fields, degree > 200000 expected.



Timings in seconds vs. output degree N , over $\mathbb{F}_p$, 26 bits prime p

Linear Differential Equations of Arbitrary Order

Given a linear differential equation with power series coefficients,

$$a_r(t)y^{(r)}(t) + \cdots + a_0(t)y(t) = 0,$$

compute the first N terms of a basis of power series solutions.

Linear Differential Equations of Arbitrary Order

Given a linear differential equation with power series coefficients,

$$a_r(t)y^{(r)}(t) + \cdots + a_0(t)y(t) = 0,$$

compute the first N terms of a basis of power series solutions.

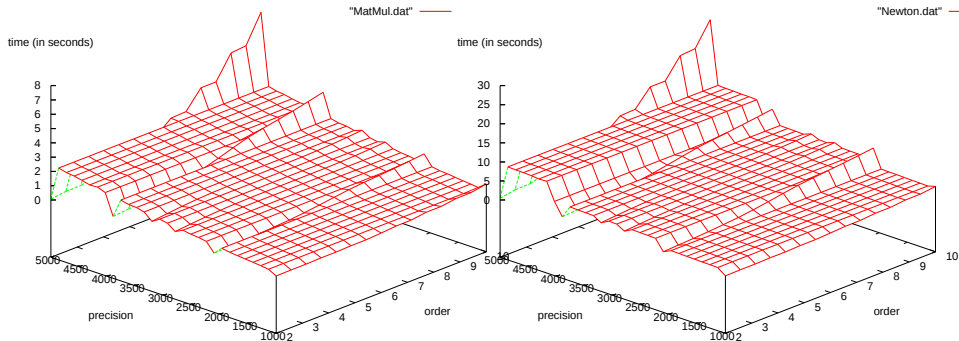
Algorithm

- ① Convert into a system $\Phi : Y \mapsto Y' - A(t)Y$ ($D\Phi = \Phi$);
- ② $D\Phi|_Y(U) = \Phi(Y)$ rewrites $U' - AU = Y' - AY$;
- ③ Variation of constants: $U = Y \int Y^{-1}(Y' - AY)$;
- ④ Y^{-1} by Newton iteration too.

Special case: recover good exponential.

[Bostan, Chyzak, Ollivier, Salvy, Schost, Sedoglavic 2007]

Timings

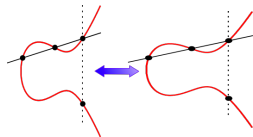


Polynomial matrix multiplication vs. solving $Y' = AY$.

Non-Linear Differential Equations

Example from cryptography:

$$\phi : y \mapsto (x^3 + Ax + B)y'^2 - (y^3 + \tilde{A}y + \tilde{B}).$$



Non-Linear Differential Equations

Example from cryptography:

$$\phi : y \mapsto (x^3 + Ax + B)y'^2 - (y^3 + \tilde{A}y + \tilde{B}).$$

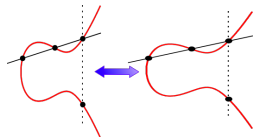
Differential:

$$D\phi|_y : u \mapsto 2(x^3 + Ax + B)y'u' - (3y^2 + \tilde{A})u.$$

Solve the **linear** differential equation

$$D\phi|_y u = \phi(y)$$

at each iteration.



Again, **quasi-linear** complexity.

[Bostan, Morain, Salvy, Schost 2008]

IV Conclusion

Conclusion

- Summary:
 - Species $\rightarrow$ statements + ogf;
 - Newton iteration has good complexity;
 - Numerical Newton iteration that gives the values of ...
power series that are the gfs of ...
combinatorial iterates.

Conclusion

- Summary:
 - Species $\rightarrow$ statements + ogf;
 - Newton iteration has good complexity;
 - Numerical Newton iteration that gives the values of ...
power series that are the gfs of ...
combinatorial iterates.
- Read the paper for:
 - Polynomial species and systems with $\mathcal{E}$;
 - Majorant species;
 - Combinatorial differential systems and linear species

$$\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) + \int \mathcal{G}(\mathcal{Z}, \mathcal{Y})$$
 (Leroux-Viennot, Labelle; combinatorial lifting of Bostan et alii);
 - PowerSet (it is not a species).

Conclusion

- Summary:

- Species $\rightarrow$ statements + ogf;
- Newton iteration has good complexity;
- Numerical Newton iteration that gives the values of ... power series that are the gfs of ... combinatorial iterates.

- Read the paper for:

- Polynomial species and systems with $\mathcal{E}$;
- Majorant species;
- Combinatorial differential systems and linear species
 $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) + \int \mathcal{G}(\mathcal{Z}, \mathcal{Y})$
 (Leroux-Viennot, Labelle; combinatorial lifting of Bostan et alii);
- PowerSet (it is not a species).

THE END

