Algorithms Seminar, 2001-2002

Frédéric Chyzak

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Algorithms Seminar, 2001–2002

Frédéric CHYZAK, éditeur scientifique

N ° 5003
Novembre 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5003--FR+ENG

Algorithms Seminar, 2001–2002

Frédéric CHYZAK, éditeur scientifique

Thème 2 — Génie logiciel
et calcul symbolique
Projet Algo

Rapport de recherche n ° 5003 — Novembre 2003 — 190 pages

Abstract: These seminar notes constitute the proceedings of a seminar devoted to the analysis of algorithms and related topics. The subjects covered include combinatorics, symbolic computation, asymptotic analysis, number theory, as well as the analysis of algorithms, data structures, and network protocols.

Key-words: combinatorics, symbolic computation, analysis of algorithms, network protocols

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Séminaire algorithmes, 2001–2002

Résumé : Ces notes de séminaires constituent les actes, le plus souvent en anglais, d'un séminaire consacré à l'analyse d'algorithmes et à ses domaines connexes. Les thèmes abordés comprennent la combinatoire, le calcul formel, l'analyse asymptotique, la théorie des nombres, ainsi que l'analyse d'algorithmes, de structures de données et de protocoles de réseaux.

Mots-clé : combinatoire, calcul formel, analyse d'algorithmes, protocoles de réseaux

ALGORITHMS SEMINAR 2001–2002[1]

Frédéric Chyzak
(Editor)

These seminar notes constitute the proceedings of a seminar whose primary goal is to cover the major methods for the average-case analysis of algorithms and data structures. Neighbouring topics of study are combinatorics, symbolic computation, network protocols, asymptotic analysis, probabilistic methods, number theory, and computational biology. The content of these annual proceedings consists of summaries of the talks, written by members of the audience.[2]

The study of combinatorial objects—their description, their enumeration according to various parameters—arises naturally in the process of analysing algorithms that often involve classical combinatorial structures like strings, trees, graphs, and permutations. Beside the traditional topics of combinatorics of words and algorithmics on words, some attention has been given in the seminar to biological applications of combinatorics.

Symbolic computation, and in particular computer algebra, plays an increasingly important role in these areas. It provides a collection of tools that allow one to attack complex models of combinatorics and the analysis of algorithms via generating functions; at the same time, it inspires the quest for developing ever more systematic solutions and decision procedures for the analysis of well-characterized classes of problems.

Asymptotic analysis is an essential ingredient in the interpretation of quantitative results supplied by the resolution of combinatorial models. Various asymptotic methods are found to be relevant to the analysis of particular algorithms. These proceedings include singularity analysis, the saddle-point method, Rice's method or Mellin transform techniques.

Our seminar shares a large part of its audience with Aléa, a working group dedicated to the analysis of algorithms and to the analysis of properties of discrete random structures. Like the previous year, this year's workshop, Alea'2002, started with a series of short courses on various aspects of probability and enumerative combinatorics. For the second time, it was decided to include lecture notes for the courses in the seminar proceedings.

The 23 articles included in this book represent snapshots of current research in the areas mentioned above. A tentative organization of their contents is given below. Five Aléa lecture notes follow.

PART I. COMBINATORICS

Several talks have been devoted to the combinatorial study of geometrical objects. Polyominoes, also known as animals, are related to percolation models and have been studied since the 1950's; yet, their enumeration is tractable for constrained classes only. The study in [**1**] is one of the few examples where the constraint of full convexity is removed, leading to an enumeration in terms of the perimeter. Classes of polyominoes that are enumerable in terms of the number of cells in

[1]Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

[2]The summaries for the past ten years are also available on the web at `http://algo.inria.fr/seminars/`.

an exact way are presented and analysed in [**2**]. The method uses a correspondence with domino tilings, which is also the topic of [**3**].

A different topic of combinatorial interest is the automatic generation of structures of a certain kind. In order to obtain a generation of trees of low complexity, giving up the exact size of the output tree proves to be an efficient paradigm as shown in [**5**]. The methodology is extended to many decomposable combinatorial classes in [**4**].

Detecting that some combinatorial enumeration series is differentiably finite is of utmost interest, for instance in the perspective of asymptotic analysis or automatic generation. This has motivated the extension of the theory of differentiably finite functions to symmetric functions. Known results are collected and new questions asked in [**6**], which has been the author's starting point for the development of a computational version of the theory.

[**1**] The Site Perimeter of Bargraphs. *M. Bousquet-Mélou.*
[**2**] Animals, Domino Tilings, Functional Equations. *M. Bousquet-Mélou.*
[**3**] Counting Domino Tilings of Rectangles via Resultants. *V. Strehl.*
[**4**] Random Generation from Boltzmann Principles. *Ph. Flajolet.*
[**5**] A Relaxed Approach to Tree Generation. *Ph. Duchon.*
[**6**] Symmetric Functions and P-Recursiveness. *M. Mishna.*

PART II. SYMBOLIC COMPUTATION

Linear algebra lies at the heart of many algorithms in computer algebra. This motivates the search for efficient algorithms dedicated to inverting or solving a linear system, and computing determinants. A major breakthrough recently occurred, making it possible to decrease the exponents of the polynomial complexity for such operations in several complexity models. This is discussed in [**7**]. Linear algebra and duality are used in [**8**] to obtain minimal polynomials of algebraic numbers efficiently; from this, one derives fast algorithms for computing rational parameterizations of the zeros of a set of polynomial equations. A careful classification of ordinary differential equations of order 2 by their differential Galois group and invariant theory enables further optimization of the Kovacic algorithm [**10**]. The structure of multivariate hypergeometric terms is studied in [**11**], where a property characterizing those that are holonomic is also given. A problem of a symbolic-numerical nature is discussed in [**12**]: Newton's iteration method is extended in order to preserve its quadratic convergence in presence of multiple roots, leading to an efficient deflation scheme.

[**7**] Computation of the Inverse and Determinant of a Matrix. *G. Villard.*
[**8**] Fast Algorithms for Polynomial Systems Solving. *A. Bostan.*
[**9**] Transseries Solutions of Algebraic Differential Equations. *J. van der Hoeven.*
[**10**] Recent Algorithms for Solving Second-Order Differential Equations. *J.-A. Weil.*
[**11**] The Structure of Multivariate Hypergeometric Terms. *M. Petkovšek.*
[**12**] Numerical Elimination, Newton Method and Multiple Roots. *J.-C. Yakoubsohn.*

PART III. ANALYSIS OF ALGORITHMS, DATA STRUCTURES, AND NETWORK PROTOCOLS

Several variants of the sorting algorithm Quicksort (classical pivot, pivot chosen as a median of three terms, or as a median of three medians of three terms) are analysed in [**13**], providing the constant in the asymptotic estimate of the expected number of comparisons, as well as various parameters of the related binary search tree (BST) structure. BST's are first and foremost a data structure for storing and retrieving data. The distribution of their height under the natural random permutation model remains a difficult problem. New results and conjectures are given in [**14**].

The next three talks deal with various aspects of the Transmission Control Protocol (TCP), the data transfer protocol used on most communication networks: the behaviour of one long TCP connexion with very low loss rate is analysed in [**15**]; the interaction of TCP sources in a large network is modeled in [**16**]; the performance—throughput, congestion—of a router routing a large number of connections constitutes the study of [**17**].

[**13**] Everything You Always Wanted to Know about Quicksort, but Were Afraid to Ask. *M. Durand.*
[**14**] Traveling Waves and the Height of Binary Search Trees. *M. Drmota.*
[**15**] Microscopic Behavior of TCP. *Ph. Robert.*
[**16**] Interaction Between Sources Controlled by TCP. *F. Baccelli.*
[**17**] Asymptotic Analysis of TCP Performances Under Mean-field Approximation. *Ph. Jacquet.*

PART IV. ASYMPTOTICS AND ANALYSIS

The saddle-point method is the basis for many asymptotic complexity analyses of algorithms and for many asymptotic analyses of combinatorial enumerations. An extension of the theory for multi-dimensional integrals is presented in [**18**], together with an application to optics. A variant of the Borel–Laplace transform is introduced in [**19**] and provides an integral representation for a certain kind of divergent series. Highly oscillatory multivariate integrals that are related to volumes of polyhedra are estimated in [**20**].

[**18**] A Hyperasymptotic Approach of the Multi-Dimensional Saddle-Point Method. *É. Delabaere.*
[**19**] Ramanujan's Summation. *É. Delabaere.*
[**20**] Multi-Variable sinc Integrals and the Volumes of Polyhedra. *J. Borwein.*

PART V. NUMBER THEORY

A general identity for Dirichlet convolutions of completely multiplicative sequences is provided in [**21**]. The calculations of irrationality measures in [**23**] make use of Padé approximations, a tool that is also of frequent use in computer algebra algorithms. This leaves us with the open problem of automating such calculations. Partial independence results between multiple Zeta values have recently been obtained by a computer algebraic approach; in [**22**], it is proved that the ζ function has infinitely many irrational values on odd integers, and that more specifically, at least one number between $\zeta(5)$, $\zeta(7)$, $\ldots$, $\zeta(21)$ is irrational.

[**21**] L-Series of Squares of Squares. *J. Borwein.*
[**22**] Irrationality of the ζ Function on Odd Integers. *T. Rivoal.*
[**23**] Irrationality Measures of $\log 2$ and $\pi/\sqrt{3}$. *N. Brisebarre.*

PART VI. MISCELLANY

The spontaneous folds in the geometrical conformation of t-RNA (transfer RNA) are reflected in the nucleotidic sequence by quasi-palindromical subsequences. This secondary RNA structure has to be analysed in order to understand the biological function of a given t-RNA. An algorithm for approximate matching of sequences with folding constraints is introduced in [**24**]. Evolutionary algorithms are algorithms of stochastic optimization based on a rough parallel with the Darwinian evolution of biological populations. They provide a heuristic, yet occasionally effective approach, and can be justified rigourously in a number of cases. A survey of the domain is presented in [**25**].

[**24**] Approximate Matching of Secondary Structures. *M. Raffinot.*

[25] Les algorithmes évolutionnaires : état de l'art et enjeux *(Evolutionary Algorithms: State of the Art and Stakes). M. Schoenauer.*

PART VII. ALEA'2002 LECTURE NOTES

The by now classical approach to average-case analysis by means of generating series ceases to be effective when the operations in an algorithm are too correlated. An alternative methodology based on dynamical systems theory has recently been introduced to tackle such situations. A survey of almost ten years of research is presented in **[26]**. Martingales are one of the tools from probability theory often used in the analysis of algorithms and data structures, starting with the study of Galton–Watson trees. Theory and applications are discussed in **[27]**. The 3-SAT problem consists in determining if a boolean formula with 3 literals per clause is satisfiable; it is the prototype for problems with phase transitions. A survey of methods giving upper bounds for the transition threshold is offered in **[28]**. Two approaches to the random generation of combinatorial structures are the topic of **[29]**: a recursive approach to uniform random generation of nicely decomposable structures; an approach by means of Markov chains for less well-behaved structures, where almost-uniform generation can be envisaged. Other attacks by ad hoc algorithms and rejection methods are discussed in **[30]**.

[26] Systèmes dynamiques et algorithmique *(Dynamical Systems and Algorithms). V. Baladi and B. Vallée.*

[27] Martingales discrètes et applications à l'analyse d'algorithmes *(Discrete Martingales Applied to Algorithms Analysis). B. Chauvin.*

[28] Phase Transitions and Satisfiability Threshold. *O. Dubois and V. Puyhaubert.*

[29] Génération aléatoire *(Random Generation). A. Denise.*

[30] Combinatorics and Random Generation. *D. Gouyou-Beauchamps.*

Acknowledgements. The lectures summarized here emanate from a seminar attended by a community of researchers from the Algorithms Project at INRIA and the greater Paris area. Its organization is conducted by Frédéric Chyzak, Philippe Flajolet, and Bruno Salvy. The editor expresses his gratitude to the various people who have actively supported this joint enterprise. Thanks are due to the speakers and to the authors of summaries. Many of them have come from far away to attend a seminar and kindly accepted to write the summary. We are also greatly indebted to Virginie Collette for making all of the organization work smoothly.

The editor,
F. CHYZAK

Part I

Combinatorics

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 3–6.

Available online at the URL
http://algo.inria.fr/seminars/.

The Site Perimeter of Bargraphs

Mireille Bousquet-Mélou
CNRS, Labri, Bordeaux (France)

May 13, 2002

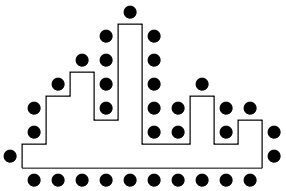
Summary by Sylvie Corteel

Abstract

The site perimeter enumeration of polyominoes that are both column- and row-convex is a well-understood problem that always yields algebraic generating functions. Counting more general families of polyominoes is a far more difficult problem. Here Mireille Bousquet-Mélou and Andrew Rechnitzer enumerate (by their site perimeter) the simplest family of polyominoes that are not fully convex—bargraphs. The generating function they obtain is of a type that has rarely been encountered so far in the combinatorics literature: a q-series into which an algebraic series has been substituted.

1. Introduction

A *polyomino* is a finite connected union of cells on a regular planar lattice (here the square lattice). The enumeration of polyominoes is a long-standing "elementary" combinatorial problem [1, 2, 3] that has some motivations in physics. The authors study the site perimeter of bargraphs. The *site perimeter* of a polyomino the number of nearest-neighbour vacant cells. A *bargraph* is a column-convex polyomino, such that its lower edge lies on the horizontal axis. It is uniquely defined by the heights of its columns. Here is a bargraph whose site perimeter is 33:



The site perimeter parameter is of considerable interest to physicists and probabilists. The enumeration of polyominoes according to their area and their site perimeter is equivalent to solve the problem of the site percolation.

This abstract is in four parts: functional equation, generating function, analysis of the generating function and application to chemistry (self-avoiding polymers). We will not give the details of the proofs. The article [1] can be found on Bousquet-Mélou's web page.

2. Functional Equation

Let $B(s; x, y, p)$ be the generating function of bargraphs enumerated by the height of the last column, the horizontal and vertical perimeter and the site perimeter.

Proposition 1. *The generating function of bargraphs satisfies the functional equation*

$$B(s) = a(s) + b(s)B(1) + c(s)B(sq) + d(s)B(s),$$

where $B(s)$ *denotes* $B(s;x,y,p)$, $q = py$, *and*

$$a(s) = \frac{xsqp^3}{1-spq}, \qquad c(s) = \frac{x^2sqp^3(1-p)}{(1-q)(1-sq)(1-spq)},$$
$$b(s) = \frac{xsp\big((1-sq)(1-spq) + xs^2p^2q^2(1-p)\big)}{(1-s)(1-sq)(1-spq)},$$
$$d(s) = -\frac{xp\Big((1-q)(1-p)(1+s^2pq) + sp\big((1-q)(1+pq-2q) + xpq^2(1-p)\big)\Big)}{(1-q)(1-s)(1-sqp)}.$$

Let us briefly explain that equation. A bargraph:

- may consist of a single column, whence the term $xsqp^3/(1-spq)$;
- may be *steady*. It can then be (uniquely) constructed by duplicating the last column of some bargraph, whence the term $xp^2B(s)$;
- may be *descending*. It can then be constructed by appending a shorter column to the right of some bargraph, whence the term $xp\big(sB(1) - B(s)\big)/(1-s)$;
- may be *ascending*, and then there are two cases:
  - the ascending bargraph is construted from a *ascending* or a *steady* one:

$$xsqp^2\Big(B(s) - xp\big(sB(1) - B(s)\big)/(1-s)\Big)/(1-spq);$$

  - the ascending bargraph is construted from a *descending* one:

$$\frac{x^2sp^3B(1)}{(1-s)(1-sq)} - \frac{x^2sp^3q(1-pq)B(s)}{(1-s)(1-q)(1-spq)} + \frac{x^2sp^3q(1-p)B(sq)}{(1-q)(1-sq)(1-spq)}.$$

3. Site Perimeter Generating Functions

In this section the functional equation of Proposition 1 is solved. The method combines two different techniques that have appeared previously in the combinatorics literature, but which have so far been applied independently. One of them is a simple iteration technique, which aims to "kill" the $B(sq)$ term. It was the key tool in [3]. The other one is the so-called *kernel method* which has been known since the 70's and is currently undergoing something of a revival.

Iteration. Let

$$\alpha(s) = \frac{a(s)}{1-d(s)}, \quad \beta(s) = \frac{b(s)}{1-d(s)}, \quad \gamma(s) = \frac{c(s)}{1-d(s)}.$$

Then $B(s) = \alpha(s) + \beta(s)B(1) + \gamma(s)B(sq)$ and when one iterates, the result is:

$$B(s) = \sum_{k\geq 0} \gamma(s)\dots\gamma(sp^{k-1})\big(\alpha(sp^k) + \beta(sp^k)B(1)\big). \tag{1}$$

The denominators of all the summands have a commun factor: $\eta(s) = (1+p-p^2)(1+s^2p^2) - s(1+2p^3-p^4-p^5)$. Moreover $1-d(s) = \eta(s)/\big((1-s)(1-sp^2)\big)$.

Kernel. The equation (1) is multiplied with $(1-d(s))$ and $B(s)$ is eliminated by taking $s=\sigma$ with $\eta(\sigma)=0$:

$$\sigma(p)=\frac{1+2p^3-p^4-p^5-\sqrt{(1+2p^3-p^4-p^5)^2-4p^2(1+p-p^2)^2}}{2p^2(1+p-p^2)}.$$

Then

$$B(1)=-\frac{a(\sigma)+c(\sigma)+\sum_{k\geq 0}\gamma(\sigma)\dots\gamma(\sigma p^{k-1})\alpha(\sigma p^k)}{b(\sigma)+c(\sigma)+\sum_{k\geq 0}\gamma(\sigma)\dots\gamma(\sigma p^{k-1})\beta(\sigma p^k)}.$$

The result is:

Theorem 1. *Let b_n be the number of bargraphs with site perimeter n. Let $\sigma=\sigma(p)$ be the following algebraic power series in p:*

$$\sigma(p)=\frac{1+2p^3-p^4-p^5-\sqrt{(1+2p^3-p^4-p^5)^2-4p^2(1+p-p^2)^2}}{2p^2(1+p-p^2)}.$$

Then the site perimeter generating function of bargraphs is

$$\sum_{n\geq 0}b_np^n=\frac{-p^3\displaystyle\sum_{n\geq 0}\frac{\sigma^n p^{\binom{n+5}{2}}}{(p)_n(\sigma^2p^3)_n(1+p-p^2)^n}}{\displaystyle\sum_{n\geq 0}\frac{\sigma^n p^{\binom{n+5}{2}}}{(p)_n(\sigma^2p^3)_n(1+p-p^2)^n}\frac{(1-\sigma p^{n+1})(1-\sigma p^{n+2})+\sigma^2p^{2n+4}(1-p)}{(1-\sigma p^n)(1-\sigma p^{n+1})}},$$

where we use $(a)_n$ to denote the product $(1-a)(1-ap)\dots(1-ap^{n-1})$.

4. Analysis of the Generating Function

In this section two aspects of the generating function of Theorem 1 are analysed: the asymptotic behaviour of the number of bargraphs with site perimeter n, and the nature of the width and site perimeter generating function.

Asymptotic behavior. The asymptotic behavior of the number of bargraphs with site perimeter n is determined by analysing the singularity structure of the generating function of Theorem 1. An examination of this series shows that the possible sources of singularities are:

- divergence of the summands in the numerator and denominator,
- divergence of the numerator or denominator,
- a singularity arising from the square-root in $\sigma(p)$,
- poles given by the zeros of the denominator.

It is in fact the case that the dominant singularity is a square-root singularity arising from the square-root singularity in $\sigma(p)$.

Theorem 2. *The number of bargraphs with site perimeter n grows asymptotically like $Cp_c^{-n}n^{-3/2}$ for some positive constant C, where $p_c=0.45002...$ is the smallest positive solution of*

$$1-2p-2p^2+4p^3-p^4-p^5=0.$$

Nature of the width and site permeter generating function. By iterating the functional equation, the coefficient of x^n in the bargraph generating function $B(1; x, 1, p)$ is a rational function of p, whose denominator is a product of cyclotomic polynomials.[1] They suggest that a new cyclotomic polynomial factor appears in the denominator of every second coefficient of x, so that more and more singularities accumulate on the unit circle $|p| = 1$.

Proposition 2. *For $n \geq 2$, the coefficient of x^{2n-3} in the bargraph generating function $B(1; x, 1, p)$ is a rational function of p that is singular at any primitive n-th root of unity.*

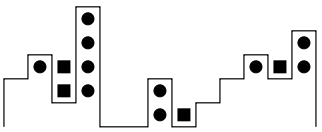
Such an accumulation of singularities indicates that the power series is not D-finite, so that:

Theorem 3. *The generating function $B(1; x, 1, p)$ which counts bargraphs by width and site perimeter is not D-finite. Consequently, the series $B(1; x, y, p)$ and $B(s; x, y, p)$ are not D-finite either.*

Remark. The non-D-finitedness of $B(1; x, 1, p)$ does not give any information about the nature of the power series $B(1; 1, 1, p)$. One can readily construct multivariate series that are not D-finite, whose specialisations are D-finite.

5. Self-Avoiding Polymers

Another model is the model of self-avoiding polymers. These polymers consist of walks above the horizontal axis that use North, East, and South steps. For this model, two parameters are important: the number of contacts with the horizontal axis (East step of height 0) and the number of interactions (circles and squares on the figure). Here is a polymer of length 41 with 3 contacts and 14 interactions.



Let $Z(t, w, q)$ be generating function of the walks enumerated by the length, the number of contacts and the number of interactions. It is conjectured that the phase diagram with coordinates q and w has three phases: confined, collapsed, and free.

By adding a North step at the beginning of the walk and a South step at the end, one gets a bargraph. The study of the parameters are possible if the interactions are separated into two classesé: internal (circles) and external (squares). For example the polymer on the figure has 9 internal interactions and 5 external interactions. It is possible to enumerate these bargraphs according to their length, contacts and internal or external interactions with the same techniques as before. For the external interactions, the generating function can be calculated explicitely and the authors can show that there is no collapsed phase. For the external interactions, the generating function can be calculated explicitely and is algebraic. Is it possible to obtain the generating function of these walks enumerated by internal and external interactions?

Bibliography

[1] Bousquet-Mélou (M.) and Rechnitzer (A.). – The site-perimeter of bargraphs. *Advances in Applied Mathematics*, vol. 31, n° 1, 2003, pp. 86–112.

[2] Bousquet-Mélou (Mireille). – *Combinatoire énumérative.* – Habilitation à diriger des recherches, LaBRI, Université Bordeaux 1, December 1996. 89 pages.

[3] Bousquet-Mélou (Mireille). – A method for the enumeration of various classes of column-convex polygons. *Discrete Mathematics*, vol. 154, n° 1-3, 1996, pp. 1–25.

[1] The cyclotomic polynomials $\Psi_d(x)$ are the factors of $(1 - x^n)$, for $n \geq 1$. More precisely, $(1 - x^n) = \prod_{d|n} \Psi_d(x)$.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 7–12.

Available online at the URL
http://algo.inria.fr/seminars/.

Animals, Domino Tilings, Functional Equations

Mireille Bousquet-Mélou
Labri, Université Bordeaux 1

May 13, 2002

Summary by Cyril Banderier

Abstract

This work lies within the framework of the great beat of animals in a square lattice: how to construct some new classes of animals, as large as possible, with enough structure to be exactly enumerable? It should be borne in mind that an animal is a finite connected set of vertices of a lattice (e.g., the square lattice), defined up to a translation, and that we still do not know the asymptotics of the number of animals with n vertices.

Our starting point is a correspondence, due to Viennot, between directed animals and pyramids of dominoes. We define a (much) larger class of animals, in one-to-one correspondence with some so-called connected domino tilings, and we proceed to their enumeration.

To this aim, we have to solve a functional equation, a variant of which gives the generating functions of directed animals. The two models are however quite distinct: directed animals have an algebraic generating functions, and a growing constant equal to 3, whereas our new class of animals has a non D-finite generating function and a growing constant of 3.58.

We can say that we did half the journey until the animal Graal: their growing constant is estimated to 4.06... (joint work with Andrew Rechnitzer).

One of the most celebrated open problems in combinatorics is the enumeration of animals (also called polyominoes). A polyomino of area n is a connected union of n cells on a lattice (symmetries are not taken into account: e.g., there are two polyominoes of area 2). Animals can be seen as duals of a polyominoes, with each cell replaced by a vertex at its centre.

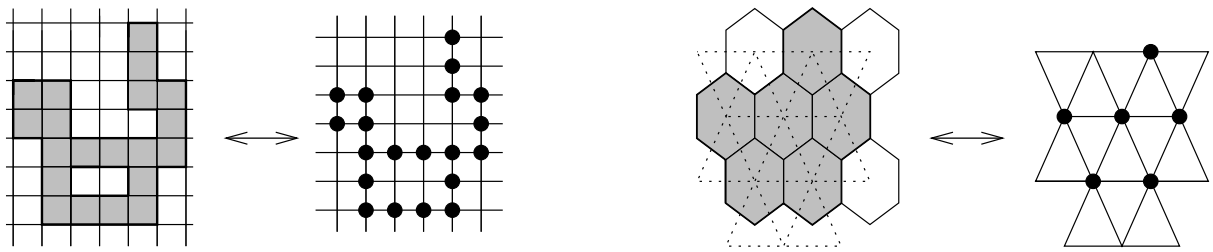


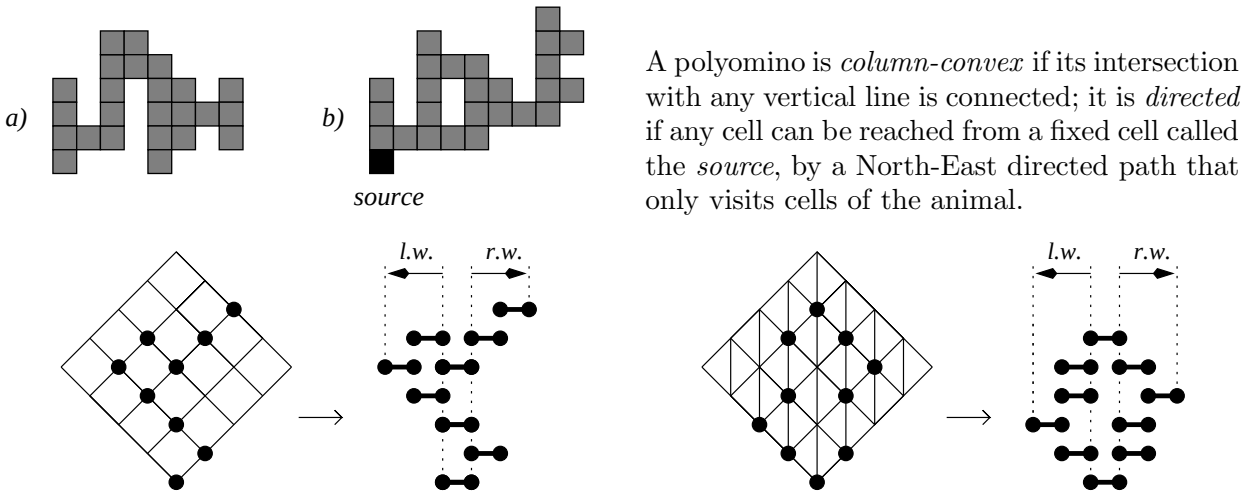
FIGURE 1. Polyominoes with square and hexagonal cells, and the corresponding animals on the square and triangular lattices.

Since the 1950's, combinatoricians and physicists (as animals are related to percolation models) tried without success to get a nice formula for the number of animals of size n or to make their asymptotics explicit.

Let a_n be the number of animals of size n on the square lattice. A concatenation argument due to Klarner implies that a_n has an exponential growing rate, i.e., $a_n^{1/n}$ converges to a constant μ (called

Model	μ	Nature of the GF	Who solved it (first)
Rectangles	1	q-series	obvious
Ferrers diagrams (partitions)	1	q-series	Euler 1748
Stacks	1	q-series	Auluck 1951, Wright 1968
Parallelogram	2.30...	q-series	Klarner & Rivest 1974
Directed convex	2.30...	q-series	Bousquet-Mélou & Viennot 1992
Convex	2.30...	q-series	Bousquet-Mélou & Fédou 1995
Bargraph (compositions)	2	rational	obvious
Directed column convex	2.62...	rational	Moser, Klarner 1965
Column convex	3.20...	rational	Temperley 1956
Directed	3	algebraic	Dhar 1982
Stacked directed	3.5	algebraic	Bousquet-Mélou & Rechnitzer 2002
Multi directed	3.58...	non D-finite	Bousquet-Mélou & Rechnitzer 2002
General	4.06?	???	You?

TABLE 1. Some of the solved subclasses of square lattice polyominoes and their growth constants.



A polyomino is *column-convex* if its intersection with any vertical line is connected; it is *directed* if any cell can be reached from a fixed cell called the *source*, by a North-East directed path that only visits cells of the animal.

FIGURE 2. A column-convex polyomino (a) and a directed polyomino (b). The second line illustrates that replacing each vertex of an animal by a dimer transforms a directed animal into a "pyramid" (a heap of dimers).

Klarner's constant). Numerical studies suggest that $a_n \approx C\frac{4.06^n}{n}$. The first 46 terms $a_1, \ldots, a_{46}$ have been computed;[1] it begins like: 1, 2, 6, 19, 63, 216, 760, 2725, 9910, ... As a byproduct (via Klarner's concatenation argument), it implies that $3.9 < \mu < 4.65$.

As is usual, people tried to solve simpler problems which were more or less direct simplification of the general model. Progress were done by adding some convexity or directedness constraints—see Table 1.

[1]See Steve Finch's website on constants at `http://algo.inria.fr/bsolve/constant/constant.html` for up-to-date datas on the Klarner's constant.

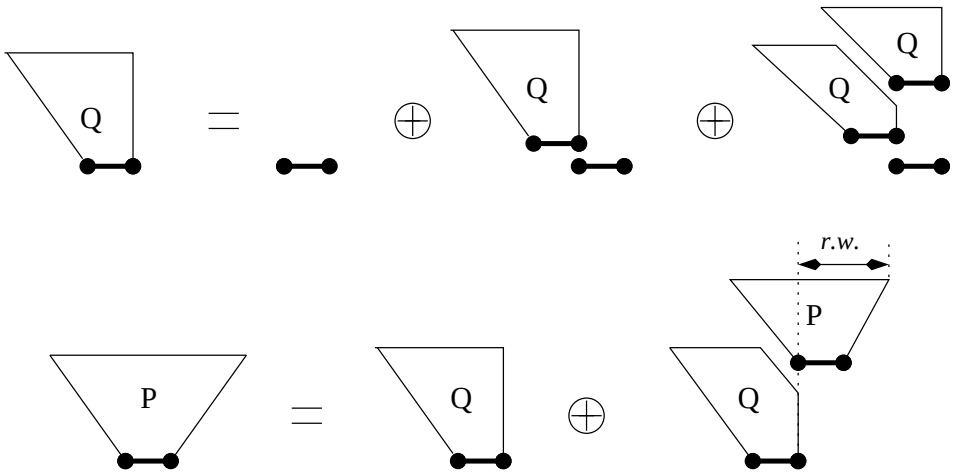


FIGURE 3. The first line shows that a half-pyramid Q factorizes in smaller half-pyramids. The second line shows that a pyramid P can be factorized in terms of half-pyramids Q and a smaller pyramid.

The Schützenberger methodology (also called "the symbolic method") classically gives the generating functions of combinatorial structures which factorize. For pyramids (heap of dimers), the factorization in Figure 3 gives a system of functional equation $P(x) = Q(x) + Q(x)P(x)$ and $Q(x) = x + xQ(x) + xQ(x)^2$, solving it gives the generating functions of half-pyramids

$$Q(x) = \frac{1 - x - \sqrt{(1+x)(1-3x)}}{2x}.$$

From this, the bivariate generating function of pyramids (x encoding the number of dimers and w encoding the right width) is $P(x,w) = Q(x)\big(1 - uQ(x)\big)^{-1}$. This gives $\mu = 3$ for directed animals (see Table 1), and also that their average width (which is given by twice the *right* width plus one) is asymptotically $6\sqrt{3\pi n}$.

We now define in the following figure two new classes of animals: stacked directed animals and multidirected animals. (See examples on Figure 4.)

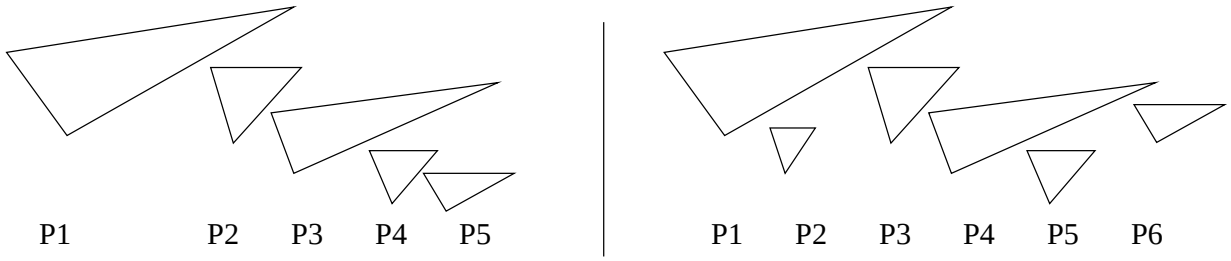


FIGURE 4. Each triangle represents a directed animal. In stacked directed animal (left) each directed animal component, P_i lies below P_{i-1}, whereas in a multi-directed animal (right) P_i lies below P_j for some $j < i$. So the right drawing is not a stacked directed animal as P_3 is *above* P_2.

We have already computed $P(x,v)$, the generating function for pyramids. The generating function $S(x,w,t)$ for stacked directed animals (x enumerates the number of dimers, w the right width and t the number of sources) is algebraic and given by $S(x,w,t) = t\frac{P(x,w)}{1-tP(x,1)^2}$. This comes easily from the functional equation

$$S(x,w,t) = tP(x,w) + tP(x,w)\partial_w S(x,1,t)$$

which reflects the fact that a stacked animal is either a single pyramid or a pyramid with another stacked animals "attached" below it. There are r ways to attach it, if r is the length of the pyramid; this "attachement" (or "pointing") is translated by a differentiation with respect to w in the functional equation.

The generating function for multi-directed animals is

$$M(x) = \frac{Q}{(1-Q)\left(1-\sum_{k\geq 1}\frac{Q^{k+1}}{1-Q^k(1+Q)}\right)}.$$

Consequently, $M(x)$ is not D-finite;[2] this comes from the fact that the zeroes of $1-q^n(1+q)$ accumulate on a part of the circle $|q|=1$, whereas a D-finite function has only finitely many singularities.

The generating function is in fact obtained by $M(x) = C(x,x,1)$, where $C(x,y,w)$ is the generating functions of connected heaps (x encodes the number of dimers, w the width and y the size of the rightmost column) and satisfies the following functional equation

$$C(x,y,w) = \frac{uy}{1-y} + \frac{u}{1-y}C\left(x, \frac{x}{1-y}, w\right) - wC(x,x,w)\,.$$

Iterating this recursive definition leads to

$$1 + C(x,y,w) = \left(\sum_{n\geq 0}\frac{u^n}{F_n(x) - yF_{n-1}(x)}\right) - \left(\sum_{n\geq 1}\frac{u^n}{F_n(x)}\right)\big(1 + C(x,y,w)\big)$$

which is equivalent to

$$C(x,y,w) = -1 + \left(\sum_{n\geq 0}\frac{u^n}{F_n(x) - yF_{n-1}(x)}\right)\left(\sum_{n\geq 0}\frac{u^n}{F_n(x)}\right)^{-1} \tag{1}$$

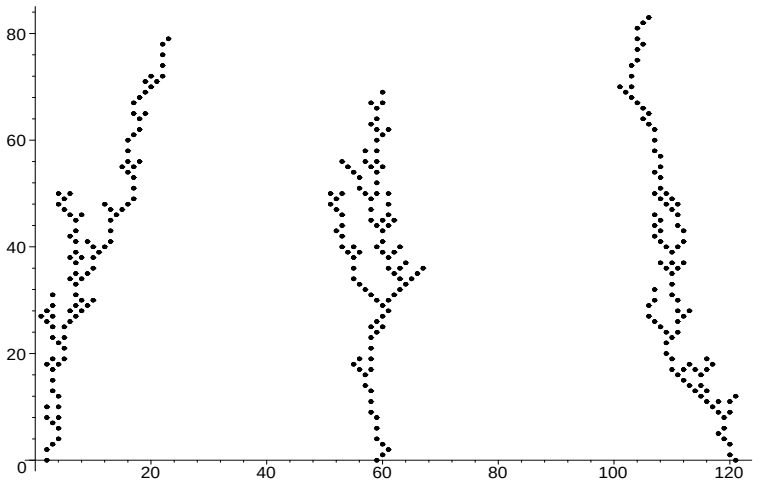
where $F_n(x)$ stands the nth Fibonacci polynomial, defined by $F_0 = F_1 = 1$ and $F_n = F_{n-1} - xF_{n-2}$.

It is interesting to note that from formulas similar to (1), one gets that some other generating functions $R(x,y,w)$ are non D-finite. The proof relies on the fact that these generating functions involve

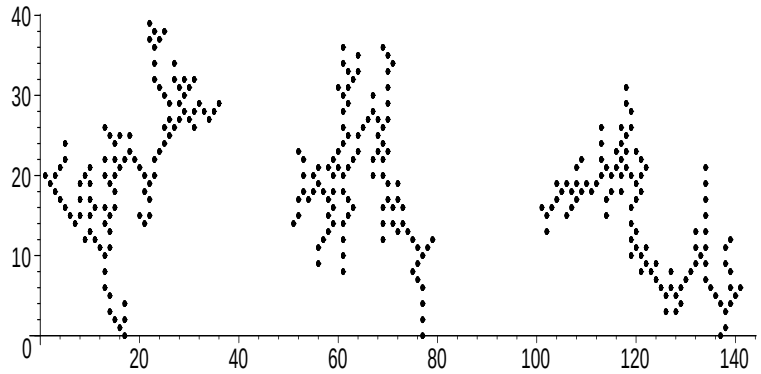
$$\sum_{n\geq 1}\frac{q^n}{1-q^n} = \sum_{n\geq 1} d(n)q^n \qquad \text{or} \qquad \sum_{n\geq 1}\frac{q^n}{(1-q^n)^2} = \sum_{n\geq 1}\sigma(n)q^n$$

where $d(n)$ is the number of divisors of n and $\sigma(n)$ the sum of the divisors of n, two well-known functions of number theory. Evaluating these functions modulo 2 relates them to the generating functions of square numbers $\sum_{n\geq 1} z^{n^2}$, which is not rational (either as a lacunary series, either by p-automatic considerations). But a series with integer coefficients and with radius 1, is either rational or has the circle as natural boundary (Fatou–Pólya–Carlson theorems). So $R(x,y,w)$ does not have finitely many singularities and hence is not D-finite.

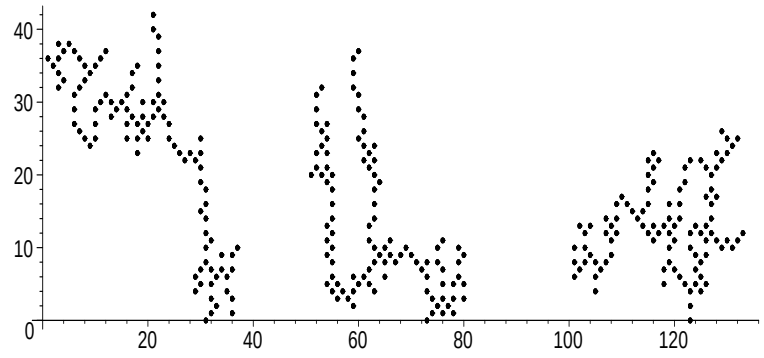
[2]Recall that a function $F(x)$ is called D-finite whenever there are some polynomials $p_i(x)$ and an integer d such that $p_d(x)\partial^d F(x) + \cdots + \partial F(x) + p_1(x)F(x) + p_0(x) = 0$. This is an important class of generating functions, very well suited to computer algebra methods.



3 directed animals.
(width= $O(\sqrt{n})$ in average)



3 stacked directed animals.
(width= $O(n)$ in average)



3 multi-directed animals.
(width = $O(n)$ in average)

FIGURE 5. Pictures of animals drawn uniformly at random amongst animals of size 100 (Mireille's zoo).

From a prospective viewpoint, it is perhaps possible to extend this approach to more sophisticated structures of animals (e.g., partially directed animals). The nature of the generating function of general animals/polyominoes remains an open problem.

This small note is a summary of M. Bousquet-Mélou & A. Rechnitzer article [1], available online at `http://dept-info.labri.u-bordeaux.fr/~bousquet/`.

Bibliography

[1] Bousquet-Mélou (M.) and Rechnitzer (A.). – Lattice animals and heaps of dimers. *Discrete Mathematics*, vol. 258, 2002, pp. 235–274.

[2] Cartier (P.) and Foata (D.). – *Problèmes combinatoires de commutation et réarrangements*. – Springer-Verlag, Berlin, 1969, *Lecture Notes in Mathematics*, vol. 85, iv+88p.

[3] Viennot (Gérard Xavier). – Heaps of pieces. I. Basic definitions and combinatorial lemmas. In *Combinatoire énumérative (Montreal, Que., 1985/Quebec, Que., 1985)*, pp. 321–350. – Springer, Berlin, 1986.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 13–16.

Available online at the URL
http://algo.inria.fr/seminars/.

Counting Domino Tilings of Rectangles via Resultants

Volker Strehl
University of Erlangen–Nürnberg

February 25, 2002

Summary by Sylvie Corteel

Abstract

The classical cosine formula for enumerating domino tilings of a rectangle, due to Kasteleyn, Temperley, and Fisher is proved using a combination of standard tools from combinatorics and algebra. For further details see [4].

1. Introduction

A classical result in combinatorial enumeration, first proved by Kasteleyn [3] gives the number of domino tilings of an $m \times n$ rectangle (mn even) as

$$k_{m,n} = \prod_{j=1}^{\lceil m/2 \rceil} \frac{c_j^{n+1} - d_j^{n+1}}{2b_j}$$

with $b_j = \sqrt{1 + \cos^2 \frac{j\pi}{m+1}}$, $c_j = b_j + \cos \frac{j\pi}{m+1}$, and $d_j = b_j - \cos \frac{j\pi}{2m+1}$.

The result can be written in a nicer way when m and n are even to get the "cosine formula:"

$$k_{2m,2n} = 4^{mn} \prod_{j=1}^{m} \prod_{k=1}^{n} \left(\cos^2 \frac{j\pi}{2m+1} + \cos^2 \frac{k\pi}{2n+1} \right)$$

Here is a new proof of this cosine formula. It uses the following notions:

- the method of determinant evaluation by counting families of non-intersecting paths in a graph,
- the inversion formula relating heaps and trivial heaps in a commutation monoid,
- in the particular case of a line, the interpretation of heaps in terms of lattice paths and their relation to the matching polynomials ,
- the determinant evaluations due to Laplace and Binet–Cauchy
- the Sylvester matrix of two polynomials and its determinant, the resultant.

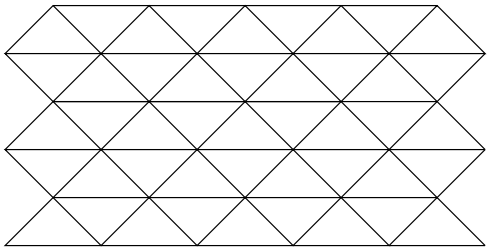
These notions are explained in Section 2 of the full paper [4]. We now concentrate on the proof. The idea is to show that the number of domino tilings of a $(2m \times 2n)$ rectangle can be expressed as a resultant of two matching polynomials from which the cosine formula can be deduced. In Section 3 a multivariate version is given.

2. The Proof

2.1. **From tilings to paths.** Domino tilings of a $2m \times 2n$ rectangle can be coded by systems of vertex-disjoint paths in a particular graph which is part of the Generalized Pascal Triangle. The

graph $\Gamma_{m,n}$ can be defined as a graph whose vertices are the lattice points $(i,j) \in \mathbb{Z}$ for $0 \leq i \leq 2n$, $0 \leq j \leq 2m$, and $i+j$ even, and whose vertex (i,j) has three outgoing edges to vertices $(i+1, j+1)$, $(i+2, j)$, and $(i+1, j-1)$.

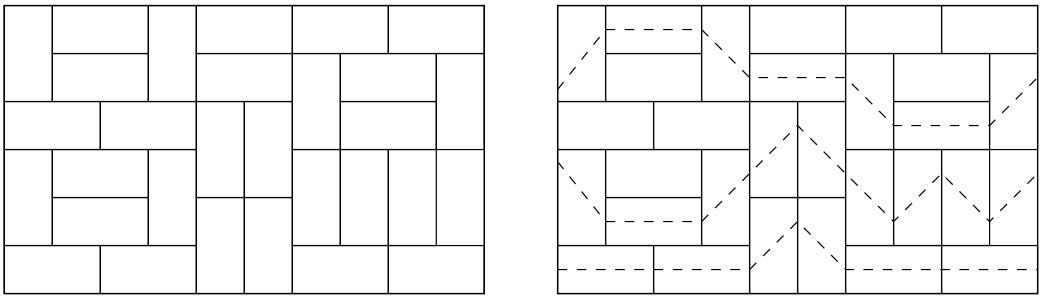
The m sources are the points of abscissa 0 and the m targets are the points of abscissa $2n$. The ith source has coordinates $\big(0, 2(i-1)\big)$ and the ith target has coordinates $\big(2n, 2(i-1)\big)$. An example of the graph $\Gamma_{3,4}$ is given below:



Domino tilings are in bijection with sets of m non-intersecting paths on $\Gamma_{m,n}$. Given a tiling, start on the left side and traverse the tiled rectangle according to the rules:

- if a vertical tile is hit traverse diagonally,
- if a horizontal tile is hit traverse straight.

Starting with a tiling on the 6×8 rectangle an example of the bijection is illustrated:



Using the theory of non-intersecting paths [1], this shows that $k_{2m,2n} = \det H_{m,n}$ where the entry $h_{i,j}$ in $H_{m,n}$ is the number of paths from the ith source to the jth target.

2.2. **Extending the graphs of the path.** Now $\Gamma_{m,n}$ is extended to the left and to the right to create a new graph $\bar{\Gamma}_{m,n}$ by adding to it:

- vertices $(i,j) \in \mathbb{Z}$ for $2n < i < 2n + 2m$, $2n - 2m < i - j < 2n$, and $i + j$ even,
- vertices $(i,j) \in \mathbb{Z}$ for $-2m + 2 \leq i < 0$, $-2m + 2 \leq i - j < 0$, and $i + j$ even,

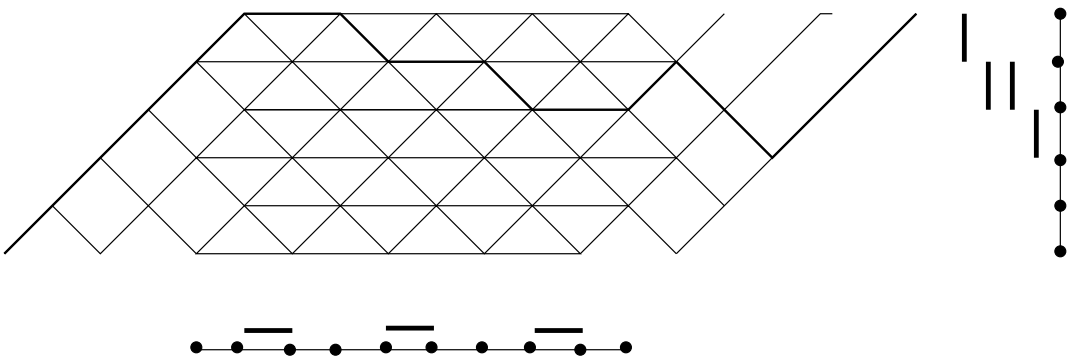
and by connecting among themselves the added vertices and the vertices of $\Gamma_{m,n}$ whenever NE-edges and SE-edges are possible.

An example of the graph $\bar{\Gamma}_{3,4}$ is given in Section 2.3.

In that graph the ith source has coordinates $(-2i + 2, 0)$ and the jth target $(2n + 2m - 2j + 1, 2m - 2)$. It is obvious that the number of systems of vertex-disjoint paths on $\Gamma_{m,n}$ is equal to the number of systems of vertex-disjoint paths on $\bar{\Gamma}_{m,n}$. This shows that $k_{2m,2n} = \det \bar{H}_{m,n}$ where the entry $\bar{h}_{i,j}$ in $\bar{H}_{m,n}$ is the number of paths from the ith source to the jth target on $\bar{\Gamma}_{m,n}$.

2.3. **Splitting the paths.** Let $\mathcal{L}_n$ denote the graph of $(n+1)$ points on a line. Given a path leading from the ith source of $\bar{\Gamma}_{m,n}$ to the jth target, the horizontal steps define a trivial heap of $\mathcal{L}_{2n-1}$ and the up-down steps are equivalent to a heap of $\mathcal{L}_{2m-1}$.

An example is given below:



If the path has k horizontal steps, then the trivial heap has k pieces and the resulting heap has $n+i-j-k$ pieces. Let $f_{n,k}$ (resp. $g_{m,k}$) be the number of trivial heaps (resp. heaps) with k pieces on $\mathcal{L}_{2n-1}$ (resp. $\mathcal{L}_{2m-1}$). Then we define $m \times (m+n)$ matrices

$$F_{m,n} = [f_{n,i-j}]_{0\le i<m, 0\le j<m+n} \text{ and } G_{m,n} = [g_{m,n+i-j}]_{0\le i<m, 0\le j<m+n}.$$

Then $\bar{H}^t_{m,n} = F_{m,n}G^t_{m,n}$.

2.4. **Dualizing path systems.** According to the Binet–Cauchy formula

$$\det F_{m,n}G^t_{m,n} = \sum_{J\in\binom{[m+n]}{n}} \det F_{m,n}\langle J\rangle \det G^t_{m,n}\langle J\rangle.$$

Let $\Phi_{m,n}$ be the graph consisting of $m+n$ horizontal lines joined by vertical edges labeled from 1 to $2n-1$ as follows for $n=3$ and $m=4$. It has $m+n$ sources $\mathbf{u}(u_1,\dots,u_{m+n})$ and $m+n$ targets $\mathbf{v} = (v_1,\dots,v_{m+n})$. The vertical edges are directed from top to bottom. The Gessel–Viennot machinery [1, 2] says that:

– $\det F_{m,n}\langle J\rangle$ = non-intersecting paths in $\Phi_{m,n}$ from $u_{[m]}$ to v_J,
– $\det G^t_{m,n}\langle J\rangle$ = non-intersecting paths $\Phi_{n,m}$ from $u_{[n]}$ to $v_{[n+m]\setminus J}$.

Therefore

$$\det G^t_{m,n}\langle J\rangle = \det F_{m,n}\langle [m+n]\setminus J\rangle.$$

2.5. **The resultant appears.** Having

$$\det F_{m,n}G^t_{m,n}\langle J\rangle = \sum_{J\in\binom{[m+n]}{m}} \det F_{m,n}\langle J\rangle \det F_{m,n}\langle [m+n]\setminus J\rangle = \det\begin{bmatrix} F_{m,n} \\ F'_{m,n}\end{bmatrix}$$

with $F'_{m,n}$ is the matrix $F_{m,n}$ where all the elements are multiplied by $(-1)^{m+n}$.

Now we have a Sylvester matrix and

$$\det\begin{bmatrix} F_{m,n} \\ F'_{m,n}\end{bmatrix} = \text{resultant}\big(f_n(t), f_m(-t)\big)$$

with

$$f_0(t) = 1,\ f_1(t) = 1+t,\ f_{n+1}(t) = (t+2)f_n(t) - f_{n+1}(t).$$

2.6. **The formula.** Now to get the formula, $f_n(t)$ can be written as:

$$f_n(t) = \prod_{j=1}^{n} \left(t + 4\cos^2 \frac{j\pi}{2n+1} \right)$$

and for two monomial polynomials $a(t)$ and $b(t)$ with roots α_i, $1 \le i \le n$ and β_j, $1 \le j \le m$:

$$\text{resultant}_t(a,b) = a_0^n b_0^m \prod_{i=1}^{n} \prod_{j=1}^{n} (\alpha_i - \beta_j).$$

The cosine formula formula follows directly.

3. A Multivariate Refinement

The counting can be refined. To each tiling one can associate a monomial $c_t(x,y)$ in the variables $\mathbf{x} = (x_1, \ldots, x_{2n-1})$ and $\mathbf{y} = (y_1, \ldots, y_{2m-1})$. The information about the positions of horizontal and vertical tiles can be carried over the path systems in the graph $\Gamma_{m,n}$. The edges will get a weight as follows:

- an horizontal edge $(i,j) \to (i+2,j)$ gets weight x_{i+1}.
- an up-edge $(i,j) \to (i+2,j)$ gets weight 1.
- an down-edge $(i,j) \to (i+1,j-1)$ gets weight y_j.

Then generalized matching polynomials $f_n(\mathbf{x};t) = f_n(x_1, \ldots, x_{2n-1}; t)$ are introduced:

$$f_0(-;t) = 1;\ f_1(x_1;t) = t + x_1;\ f_{n+1} = (t + x_{2n} + x_{2n+1}) f_n(\mathbf{x};t) + x_{2n} x_{2n-1} f_{n-1}(\mathbf{x};t).$$

It is easy to check that the proof of Section 2 goes through.

$$k_{2m,2n}(\mathbf{x},\mathbf{y}) = \text{resultant}\big(f_n(\mathbf{x};t), f_m(\mathbf{y};t)\big)$$

This can be also interpreted in terms of 2-tableaux [4].

If we set $x_i = x$ and $y_i = y$, the cosine formula counting horizontal and vertical tiles separately [3]:

$$k_{2m,2n} = 4^{mn} \prod_{j=1}^{m} \prod_{k=1}^{n} \left(y\cos^2 \frac{j\pi}{2m+1} + x\cos^2 \frac{k\pi}{2n+1} \right)$$

Now to consider the tiling of an $2m \times (2n-1)$ rectangle it suffices to set up the counting machinery for a $2m \times 2n$ rectangle and to set $x_{2n-1} = 0$ in order to have the last column of the rectangle covered with vertically oriented dominos. Then in the resultant the polyonial $f_n(t)$ has to be replaced by $\tilde{f}_n(t) = f_n(t) - f_{n-1}(t)$.

If both side lengths are odd, the same idea applies, but the polynomials always have t as a factor. This implies that the resultant vanishes which algebraically reflects the obvious combinatorial fact that a rectangle with an odd area can not be tiled by dominos.

Some other specializations can be find in the full paper [4].

Bibliography

[1] Gessel (Ira) and Viennot (Gérard). – Binomial determinants, paths, and hook length formulae. *Advances in Mathematics*, vol. 58, n° 3, 1985, pp. 300–321.

[2] Gessel (Ira M.) and Viennot (X. G.). – Determinants, paths, and plane partitions. – Preprint. Available online at `http://www.cs.brandeis.edu/~ira/`, 1989.

[3] Kasteleyn (P. W.). – The statistics of dimers on a lattice. *Physica*, vol. 27, 1961, pp. 1209–1225.

[4] Strehl (Volker). – Counting domino tilings of rectangles via resultants. *Advances in Applied Mathematics*, vol. 27, n° 2-3, 2001, pp. 597–626. – Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000).

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 17–18.

Available online at the URL
http://algo.inria.fr/seminars/.

Random Generation from Boltzmann Principles

Philippe Flajolet
Algorithms Project, INRIA Rocquencourt (France)

December 17, 2001

Summary by Maryse Pelletier and Michèle Soria

Abstract

This talk proposes a new framework for random generation of combinatorial configurations based on Boltzmann models. The idea is to perform random generation of possibly complex structured objects by placing an appropriate measure on the whole of a combinatorial class. The resulting algorithms often operate in linear time. This talk refers to a joint work with P. Duchon, G. Louchard, and G. Schaeffer, to appear in ACM STOC 2002.

1. Introduction

The problem considered here is that of generating samples of structured combinatorial objects of a certain size. In the usual setting of combinatorics, the objects should be drawn uniformly at random from the family of all objects of the same size (see, e.g., [1, 2]).

The basic principle of Boltzmann method is to relax the constraint of generating objects of a strictly fixed size, and prefer to draw objects with a somewhat randomly fluctuating size. The algorithms developed make use of a continuous control parameter $x > 0$. One can tune the value of x in order to draw objects of a size in some vicinity of a target size n.

2. Boltzmann Models

Let $\mathcal{C}$ be a combinatorial class, where each object γ has a size denoted by $|\gamma|$, and $\mathcal{C}_n$ the subclass of objects of size n. The class $\mathcal{C}$ is represented by the ordinary generating function $C(x) = \sum_{\gamma\in\mathcal{C}} x^{|\gamma|}$ in the case of unlabelled objects, and in the case of labelled objects, by the exponential generating function $C(x) = \sum_{\gamma\in\mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!}$. Only coherent values of x are to be considered, that is $0 < x < \rho_C$ where ρ_C is the radius of convergence of C.

Definition 1. The *Boltzmann model* of parameter x assigns to any object $\gamma \in \mathcal{C}$ the probability $\mathbf{P}_x(\gamma) = x^{|\gamma|}/C(x)$. A *Boltzmann generator* $\Gamma C(x)$ for a class $\mathcal{C}$ is a process that produces objects from $\mathcal{C}$ according to a Boltzmann model.

Let us point out that the Boltzmann model of parameter x, conditioned by the fact that the size of the object drawn equals n, obviously coincides with the uniform model on $\mathcal{C}_n$. Given a Boltzmann generator $\Gamma C(x)$, we thus have a rejection algorithm $\mu C(n)$, sampling uniformly over $\mathcal{C}_n$, which simply writes as: `repeat` $\gamma := \Gamma C(x)$ `until` $|\gamma| = n$. Random generation of "approximate size" is obtained by weakening the halting condition of the "repeat" loop. For instance, we refer to $\mu C(n, \varepsilon)$, where ε is a certain tolerance, for the sampler halting with condition $|\gamma| \in \big[n(1-\varepsilon), n(1+\varepsilon)\big]$.

3. Constructions for Boltzmann Generators

Let us first deal with unlabelled objects and ordinary counting generating functions. We consider combinatorial classes, constructed from finite classes by means of disjoint union, cartesian product, and sequence construction. It is well known that the corresponding functional operations on generating functions are sum, product, and quasi-inverse.

- If $\mathcal{C} = \mathcal{A} + \mathcal{B}$, a Boltzmann generator for $\mathcal{C}$ is built by calling a Boltzmann generator, either for $\mathcal{A}$ (with probability $A(x)/C(x)$) or for $\mathcal{B}$ (with probability $B(x)/C(x)$).
- If $\mathcal{C} = \mathcal{A} \times \mathcal{B}$, a Boltzmann generator for $\mathcal{C}$ generates a pair of independent elements, the first one drawn by a Boltzmann generator for $\mathcal{A}$, and the second by a Boltzmann generator for $\mathcal{B}$.
- If $\mathcal{C} = \mathrm{Seq}(\mathcal{A})$ then $\mathcal{C}$ is the solution to the symbolic equation $C = 1 + \mathcal{A}\mathcal{C}$ which recursively involves the operations of union and product mentioned above. Equivalently, a Boltzmann generator for $\mathcal{C}$ can be built by drawing K randomly according to the geometric distribution with parameter $A(x)$ and then drawing K independent elements with a Boltzmann generator for $\mathcal{A}$.

Theorem 1. *A Boltzmann generator constructed from specifications and rules above:*

1. *draws correctly from Boltzmann model;*
2. *halts with probability 1 with finite expected time;*
3. *has a complexity linear in the size of output object.*

In the case of labelled objects and exponential generating functions, the exponential Boltzmann generator is built according to similar rules, and even extended to cycle and set combinatorial constructions and the analogue of Theorem 1 holds.

4. Efficiency

Since the size N of the object produced by a Boltzmann model of parameter x has mean value $\mathbf{E}_x(N) = x\frac{C'(x)}{C(x)}$, the tuning parameter is set to the solution x_n of the equation $n = x\frac{C'(x)}{C(x)}$.

Theorem 2. *Under some technical conditions on $C(x)$, the rejection sampler $\mu C(n, \varepsilon)$, equipped with the value $x = x_n$, succeeds in one trial with probability tending to 1 as $n \to \infty$.*

Sometimes, although the technical conditions are not satisfied, drawing with approximate size can still be done in linear-time complexity with adapted halting condition. Let us mention the case of generators of supercritical sequences (a sequence $\mathcal{C} = \mathrm{Seq}(\mathcal{A})$ is said to be supercritical if $\rho_A > \rho_C$).

Theorem 3. *For supercritical sequences, the adapted singular Boltzmann generator produces a random object of size $n + O(1)$ in one trial, with high probability. And it hits n exactly in $A'(\rho_C)$ trials on average.*

Bibliography

[1] Flajolet (Philippe), Zimmerman (Paul), and Van Cutsem (Bernard). – A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, vol. 132, n° 1-2, 1994, pp. 1–35.

[2] Nijenhuis (Albert) and Wilf (Herbert S.). – *Combinatorial algorithms*. – Academic Press, New York, 1978, second edition, *Computer Science and Applied Mathematics*, xv+302p. For computers and calculators.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 19–22.

Available online at the URL
http://algo.inria.fr/seminars/.

A Relaxed Approach to Tree Generation

Philippe Duchon
Labri, Université de Bordeaux I (France)

November 5, 2001

Summary by Marni Mishna

Abstract

An algorithm for the uniform random generation of trees is described. The algorithm is notable for its simplicity and efficiency. These qualities stem largely from the fact that it does not precisely control the size of the final tree, rather, it is "relaxed." The complexity analysis yields that in certain cases the algorithm is linear. A family of variants with multiple parameters is also discussed.

1. Relaxed?

Efficient random tree generation is important in many contexts. Very often one does not specifically require trees of a particular size, but rather within a given size range. The idea at hand is to generate random trees of any possible size and reject those which are not in the given range. The generation is done in such a way to uniformly generate trees within a fixed size and to minimise the number of rejections. This simple algorithm is surprisingly efficient.

2. The Trees in Question

The trees are simple, as in the sense of Meir and Moon. That is, each family is linked to a set D of non-negative integers which dictates the possible number of children $f(s)$ a node s can have. As the trees are finite, zero is always contained in this set. More precisely, the set D corresponds to the family $\mathcal{T}_D$ of trees

$$\mathcal{T}_D = \{\, t \mid \forall s \in T, f(s) \in D \,\}.$$

Classic complete binary trees correspond to $\{0, 2\}$, for example. Other examples include 1–2 trees ($D = \{0, 1, 2\}$), general trees ($D = \mathbb{N}$) and linear? trees ($D = \{1, 0\}$).

They are simple in the sense that they easily admit a generating function decomposition. Consider $F_D(x) = \sum_n a_n x^n$ where a_n is the number of trees of size n. This can be rewritten

$$F(x) = \sum_{d \in D} xF(x)^d = \Phi_D\big(x, F(x)\big).$$

In the weighted model, the size is no longer the number of nodes, rather, a weighted sum where the weight depends on the degree of the node:

$$F(x) = \sum_{(d,w) \in D} x^{w(d)} F(x)^d = \Phi_D\big(x, F(x)\big),$$

where $w(d)$ is the weight of d.

2.1. **Restrictions on D.** We allow repetitions in the set. Further we impose a non-periodicity and a rationality requirement. These are well-described by M. Drmota in [1]. The essential characteristic guaranteed by these conditions is that F has a square root singularity. That is, $F(x_0 - h) = F_0 - C\sqrt{h} + O(h)$ and hence $a_n = C' x_0^{-n} n^{-3/2} \left(1 + O(n^{-1})\right)$. Given that F has a unique singularity x_0 of minimal modulus we set $F_0 = F(x_0) < \infty$. Hence, we have that (x_0, F_0) is a solution of $F_0 = \Phi_D(x_0, F_0)$, and further, $1 = \frac{\partial \Phi_D}{\partial F}(x_0, F_0)$.

3. Generation à la Galton–Watson

Galton–Watson trees (G.–W. trees) are formed recursively with respect to a given probability rule π. Suppose $\pi = (\pi_k)_{k\in\mathbb{N}}$ satisfies the three conditions $\sum \pi_i = 1$, $\pi_0 > 0$, $\pi_i \geq 0$. We form the G.–W. tree T_π recursively by the following method: beginning at the root node, determine the number of children k, with a random process with probability π_k, independently from the other vertices. If $k \neq 0$, recurse on each of the children. Notice that this does not a priori exclude infinite trees. However, a careful selection the probabilities can sufficiently increase the expected number of finite trees, as the following theorem indicates.

Theorem 1. *Let $m = \sum_k k\pi_k$.*

1. *If $m \leq 1$, then T_π is finite with probability 1.*
2. *In particular, if $m = 1$, The size of T_π is not integrable; the average size is infinite. We say the tree is* G.–W. critical.
3. *If $m < 1$ the expected size is $\mathbf{E}\big[|T_\pi|\big] = \frac{1}{1-m}$. We say the tree is* sub-critical.
4. *Otherwise, the tree is of infinite size with a strictly positive probability and we say it is* G.–W. sur-critical.

Example. Let us take a look at what happens in the case of binary trees. Let $\pi = (p, 0, 1-p, 0, \dots)$, $p > 0$. Suppose t is such a tree of size $2n-1$ nodes. Then there are exactly n internal nodes and $n-1$ leaves. Consequently,

$$\mathbf{P}(T_\pi = t) = \prod_{s\in t} \mathbf{P}\big(X = f(s)\big) = \prod_{s\in t} \pi_{f(s)} = p^{n+1}(1-p)^n,$$

which depends only on the size of t.

Example. The binary trees seem perhaps a special case. What can we say about 1–2 trees? Let $\pi = (\alpha, \beta, \gamma, 0, \dots)$, and let $N_i(t)$ be the number of nodes of t with degree i. Then,

$$\mathbf{P}(T_\pi = t) = \prod_{s\in t} \mathbf{P}\big(X = f(s)\big) = \alpha^{N_0(t)} \beta^{N_1(t)} \gamma^{N_2(t)}.$$

The probability depends on number of vertices with each type of degree. However, if we set $\alpha = \beta = \gamma = 1/3$, then the value $\mathbf{P}(T_\pi = t) = 1/3^{|t|}$ depends only on the size of t.

4. Probability that $T_\pi = t$?

The previous examples give us the intuition to answer the following question: under which conditions does the probability rely only on the size of the object?

An answer is found in Theorem 2, but first we motivate it with some observations. Notice that $|t| = 1 + \sum_{d\in D} dN_d(t)$. We choose some $0 < u < 1$ and let π_d be proportional to u^d. Then we get

$$\mathbf{P}(t) = \prod_{d\in D} \left(\frac{u^d}{\Phi_D(u)}\right)^{N_d(t)} = \frac{u^{\sum_d dN_d(t)}}{\Phi_D(u)^{\sum_d N_d(t)}} = \frac{u^{|t|-1}}{\Phi_D(u)^{|t|}} = \frac{1}{u}\left(\frac{u}{\Phi_D(u)}\right)^{|t|},$$

which only depends on $|t|$.

In the weighted case, we say that the size is $|t| = \sum_d w(d)N_d(t)$. We assign to $N(t) = \sum_d N_d(t)$, the number of vertices. In this case, if we select $x,\ u > 0$ and set $\pi_d = x^{w(d)}u^d/\Phi_D(x,u)$, we see that

$$\mathbf{P}(t) = \prod_{d\in D}\left(\frac{x^{w(d)}u^d}{\Phi_D(x,u)}\right)^{N_d(t)} = \frac{x^{|t|}}{u}\left(\frac{u}{\Phi_D(x,u)}\right)^{N(t)}.$$

Thus, if are principally interested by the weighted size, we can generate them uniformly with a careful selection of (x,u).

Theorem 2. *Let $\mathcal{T}_D$ be a simple labelled family of trees for which the generating series is $F(x) = \Phi_D(x,F(x))$. Further suppose that (x,u) is a couple satisfying $u = \Phi_D(x,u)$. Then the G.–W. tree defined by the generation law*

$$\pi_d = \frac{x^{w(d)}u^d}{\Phi_D(x,u)}$$

conditioned to be of weighted size n is a tree of $\mathcal{T}_D$, uniformly generated among those of weighted size n.

5. **Relaxed Random Generation**

Theorem 2 can be exploited for our random generation purposes. It implies the existence of a generation scheme where all trees of a given size are equally probable. This notion converts directly into an algorithm. But first recall the novelty here was to consider a more general situation. Instead of requiring a tree of size *exactly* n we consider an acceptable range. We will examine three ranges $[n_1,n_2]$ here: strict ($n_2 = n_1$), linear ($n_2 = (1+\lambda)n_1$) and geometric ($n_2 = n_1 + \lambda n_1^\alpha$). Consider the following algorithm which seemingly does the most naive thing.

Algorithm. Input: $\Phi_D(x,F)$, $[\,n_1,n_2\,]$.

1. Determine a couple (x,u) satisfying $u = \Phi_D(x,u)$;
2. Generate a G.–W. tree t with the generation rule $\pi_d = x^{w(d)}u^{d-1}$ until the process stops naturally, or until the total size is greater than n_2;
3. If the size of $t \notin [\,n_1,n_2\,]$ reject t and go to 2. Otherwise, output t.

Some questions must be answered here. For example, how simple is it to assure the non-periodicity of Φ_D? How do we solve for u? How many digits would be required in a numerical approximation to avoid bias?

5.1. **Complexity.** For the analysis purposes assume that determining (x,u) is inconsequential to the complexity. We evaluate:

- $E_< = \mathbf{E}\big[|T| \;/\; |T| < n_1\big]$ (average size of rejected small tree),
- $P_< = \mathbf{P}\big(|T| < n_1\big)$ (too small),
- $P_> = \mathbf{P}\big(n_2 < |T|\big)$ (too big),
- $P_= = \mathbf{P}\big(n_1 \le |T| \le n_2\big)$ (just right).

The average number of rejections for being too large is $\frac{P_>}{P_=}$. The average number of rejections for being too small is $\frac{P_<}{P_=}$. Thus, the average cost is at most

$$n_2 + n_2\frac{P_>}{P_=} + E_<\frac{P_<}{P_=}.$$

Consider now a quick calculation of $P_=$ for these three ranges.

The strict case is classic: $\mathbf{P}\big(|T| = n_1\big) = \Theta(n_1^{-3/2})$.

The linear case yields:

$$\sum_{n \le k \le (1+\lambda)n_1} \mathbf{P}\big(|T| = n_1\big) \sim n_1^{3/2} \sum_{0 \le k \le \lambda n_1} C(1 + k/n_1)^{-3/2} = \Theta(n_1^{-1/2}).$$

The geometric case yields:

$$P_= = \sum_{n_1 \le k \le n_1 + \lambda n_1^\alpha} \mathbf{P}\big(|T| = k\big)$$
$$\sim \Theta(n_1^{-3/2}) \sum_{0 \le k \le \lambda n_1^\alpha} (1 + k/n_1)^{-3/2} = C n_1^{-3/2} \lambda n_1^\alpha \Theta(1) = \Theta(n_1^{\alpha - 3/2}).$$

We can summarise the average cost for our three range types:

Case	$P_=$	Average cost
$n_2 = n_1$	$\Theta(n^{-3/2})$	$n_1 + \frac{\Theta(n_1^{1/2})}{\Theta(n_1^{-3/2})} = \Theta(n_1^2)$
$n_2 = (1+\lambda)n_1$	$\Theta(n_1^{-1/2})$	$\Theta(n_1)$
$n_2 = n_1 + \lambda n_1^\alpha$	$\Theta(n_1^{\alpha - 3/2})$	$\Theta(n_1^{2-\alpha})$

Most notably, in the linear case, the algorithm is linear in n_1. This is quite efficient.

6. **The Multivariate Case**

We can extend the allowable families of trees by looking at G.–W. trees with k types of vertices. Now we have k probability laws and for each vertex the probability of a vertex of type i to have d_j children of type j is $\pi_{i,d}$, where d indicates the collection of d_j and the probability is independent of all others, except for its ancestors. It is less clear how to verify the desired properties such as periodicity and rationality.

Theorem 3. *Let $(x, u_1, \dots, u_k)$ be such that $u_i = \Phi(x, u_1, \dots, u_k)$. The multi-type branching process defined by the laws of progeny*

$$\pi_{i,d} = \frac{x^{w_i(\mathbf{d})} u_1^{d_1} \cdots u_k^{d_k}}{\Phi_i(x, u_1, \dots, u_k)}$$

attribute to each tree, of which the root is of type 1, a probability which depends only on its size.

7. **Complications and Restrictions**

We have already discussed some of the problems of implementing such an algorithm. However, in the univariate case, they can be overcome as precise numerical evaluation is possible, and often it is easy to calculate the singularities directly. In the multivariate case there is some difficulty to verify that the rationality and non-periodicity requirements are met.

Bibliography

[1] Drmota (Michael). – Systems of functional equations. *Random Structures & Algorithms*, vol. 10, n° 1-2, 1997, pp. 103–124. – Average-case analysis of algorithms (Dagstuhl, 1995).

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 23–26.

Available online at the URL
http://algo.inria.fr/seminars/.

Symmetric Functions and P-Recursiveness

Marni Mishna
Lacim, Université du Québec à Montréal (Canada)

October 15, 2001

Summary by Henry Crapo

In his 1990 paper [1], Ira Gessel introduced a notion of D-finite for symmetric functions, and showed how it could be used to determine D-finiteness of combinatorial generating functions.

In this context, a *symmetric function* is a polynomial function of finite degree (here, n) in infinitely many variables x_1, x_2, ..., invariant with respect to arbitrary permutations of finite subsets of the variables. Typical symmetric functions are indexed either by the degree n itself, or by a partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ of n, and include the following (by way of illustration, we set $n = 3$, and take $(2, 1)$ as a typical partition of n):

1. the homogeneous symmetric functions,
$$h_n = \sum_{1 \le i_1 \le \cdots \le i_k} x_1^{i_1} \ldots x_1^{i^k}$$
so
$$h_3 = x_1^3 + x_1^2 x_2 + x_1 x_2^2 + \cdots + x_1 x_2 x_3 + \cdots ,$$
$$h_{[2,1]} = h_2 h_1 = x_1^3 + 2x_1^2 x_2 + \cdots + 3x_1 x_2 x_3 + \cdots .$$
2. the elementary symmetric functions,
$$e_n = \sum_{1 \le i_1 < \cdots < i_k} x_1^{i_1} \ldots x_1^{i^k}$$
so
$$e_3 = x_1 x_2 x_3 + x_1 x_2 x_4 + \cdots ,$$
$$e_{[2,1]} = e_2 e_1 = x_1^2 x_2 + \cdots + 3x_1 x_2 x_3 + \cdots .$$
3. the power symmetric functions,
$$p_n = \sum_{i>0} x_i^n$$
so
$$p_3 = x_1^3 + x_2^3 + \cdots ,$$
$$p_{[2,1]} = p_2 p_1 = x_1^3 + x_1^2 x_2 + \cdots + x_2^2 x_1 + \cdots .$$
4. the monomial symmetric functions,
$$m_\lambda = \sum_\sigma x_{\sigma(1)}^{\lambda_1} \ldots x_{\sigma(n)}^{\lambda_n}$$
where the sum ranges over all permutations of $\{1, \ldots, n\}$, so
$$m_{[3]} = x_1^3 + x_2^3 + \cdots ,$$
$$m_{[2,1]} = x_1^2 x_2 + x_1^2 x_3 + \cdots + x_2^2 x_1 + \cdots .$$

5. the Schur functions,

$$s_\lambda = \det(h_{j-i+\lambda_i}) \text{ where } 1 \le i, j \le k$$

so

$$\begin{aligned} s_{[3]} &= h_3 = x_1^3 + x_1^2 x_2 + x_1 x_2^2 + \cdots + x_1 x_2 x_3 + \cdots , \\ s_{[2,1]} &= \begin{vmatrix} h_2 & h_3 \\ h_0 & h_1 \end{vmatrix} \\ &= m_{[1]}(m_{[2]} + m_{[1,1]}) - (m_{[3]} + m_{[2,1]} + m_{[1,1,1]}) \\ &= m_{[2,1]} + 2m_{[1,1,1]}. \end{aligned}$$

The sets $\{p_\lambda\}$, $\{h_\lambda\}$, $\{e_\lambda\}$, $\{s_\lambda\}$, where λ ranges over all partitions of n, are bases for the vector space Λ^n of symmetric functions of degree n. The basic tool in what follows is a scalar product $\langle \cdot, \cdot \rangle$ introduced by Redfield, and characterized by the condition that the monomial and homogeneous symmetric functions be dual bases for Λ^n, that is

$$\langle m_\lambda, h_\mu \rangle = \delta_{\lambda\mu}$$

and with the property

$$\langle p_\lambda, p_\mu \rangle = z_\lambda \delta_{\lambda\mu}, \quad \text{where } z_\lambda = n!/(\lambda_1! \, 1^{\lambda_1} \ldots \lambda_n! \, n^{\lambda_n})$$

A formal power series $y \in K[[x]]$ in one variable x is *differentiably finite* or simply *D-finite*, if y and all its derivatives $y^{(n)} = \frac{d^n y}{dx^n}$ span a finite dimensional subspace over the field of rational functions over K, that is, if and only if they satisfy a non-trivial polynomial relation of the form:

$$p_n(x) y^{(n)} + p_{n-1}(x) y^{(n-1)} + \cdots + p_0(x) y = 0.$$

Rational functions, algebraic functions, and the exponential function are D-finite. D-finite functions are closed under addition, multiplication, and the Hadamard product. If f is D-finite and g is algebraic, then the composite function $f(g)$ is D-finite. (Are there weaker conditions on g that guarantee $f(g)$ D-finite?)

A function $f : N \to K$ defined on the positive integers is *polynomially recursive*, in short *P-recursive*, if it satisfies a homogeneous linear recurrence of finite degree. For example, $f(n) = n!$ satisfies $f(n) - nf(n-1) = 0$, with polynomial coefficients 1 and $-n$. A power series $\sum_n f(n)x^n$ with coefficient sequence $f(n)$ is D-finite if and only if the sequence f is P-recursive. The speaker was interested in conditions for the existence of such recurrences, rather than in their precise construction.

The concept of D-finite was extended to several variables by Zeilberger and Lipschitz, and to infinitely many variables by Gessel. Viewing f as a formal power series in infinitely many variables, p_1, p_2, ..., the power sum symmetric functions, one applies the Gessel theory to the algebra of symmetric functions. In this way we find that h_n, e_n and $\sum_\lambda s_\lambda$ are D-finite. If $f(x_1, \ldots, x_n)$ is D-finite in $x_1, \ldots, x_n$ and for each i, r_i is a polynomial in the variables $y_1, \ldots, y_m$, then $f(r_1, \ldots, r_n)$ is D-finite in $y_1, \ldots, y_m$, as long as it is well defined as a power series. In particular, if $P(x)$ is a polynomial in $p_1, \ldots, p_n$ then $e^{P(x)}$ is D-finite. In the case of infinitely many variables, Gessel proved:

Theorem 1 ([1, Theorem 8])**.** *Let f and g be symmetric functions D-finite in the p_i and t, and suppose that g involves only finitely many p_i. Then $\langle f, g \rangle$ is D-finite in t as long as it is well-defined as a power series.*

Symmetric functions f and g can be *composed* by substituting the infinite set of monomials of g for the variables of f. Thus

$$e_2(h_2) = e_2(x_1^2, x_2^2, \dots, x_1x_2, \dots) = x_1^2x_2^2 + x_1^3x^2 + x_1x_2^3 + \cdots$$
$$p_k(g) = g(p_k), \quad g(p_1) = g, \quad p_m(p_n) = p_{mn}, \quad (f_1f_2)(g) = f_1(g)f_2(g)$$

Such symmetric functions are said to arise by *plethysm*.

The speaker asks, under what conditions does composition preserve D-finiteness? Are plethysms and D-finiteness friends or enemies? (Apparently, these days, it is necessary to choose.) Gessel proved

Theorem 2 ([1, Theorem 10])**.** *If g is a polynomial in the power sum symmetric functions p_i, then $h(g)$ and $e(g)$ are D-finite.*

What are the weakest conditions for f and g that retain D-finiteness?

The inner product can be used to extract coefficients of specified monomials: the coefficient of $x_1^{\lambda_1} \dots x_k^{\lambda_k}$ in f is $\langle f, h_\lambda \rangle$. To evaluate this inner product, expand both f and h_λ in power sum symmetric functions. Gessel's following resultshows how certain sums of coefficients are D-finite.

Theorem 3 ([1, Corollary 9])**.** *Let f be a D-finite symmetric function and S a finite set of integers. Define a sequence s_n to be the sum, for all n-tuples $(\lambda_1, \dots, \lambda_n)$ in S^n, of the coefficient of $x_1^{\lambda_1} \dots x_n^{\lambda_n}$ in f. Then $s(t) = \sum_n s_n t^n$ is D-finite.*

As an application of this method:

Theorem 4 ([1, Theorem 1])**.** *Define $\theta : \Lambda \to K[[x]]$ by $\theta(p_k) = \delta_{1,k}X$. Then for any symmetric function f,*

$$\theta(f) = \sum_n a_n \frac{X^n}{n!},$$

where a_n is the coefficient of $x_1 \dots x_n$ in f. In particular, $\theta(h_n) = X_n/n!$.

The speaker provided numerous applications to graph theory, Young tableaux, and suggested further applications to nonnegative integer matrices and to permutations with forbidden sequences. The applications to graph enumeration begin with the generating function

$$G = \prod_{i<j} (1 + x_ix_j) = e(e_2),$$

which is D-finite. The coefficient of $x_1^{\lambda_1}, \dots, x_n^{\lambda_n}$ in G is the number of graphs on n vertices with specified degrees $\lambda_1, \dots, \lambda_n$. By Theorem 3 above, the generating function for graphs on n vertices with certain specified classes of degree sequences are D-finite, settling a problem of Goulden and Jackson. For example, taking $S = \{1\}$ counts matchings, $S = \{2\}$ counts (disjoint unions of) circuits, $S = \{k\}$ counts k-regular graphs.

Since the coefficient of $x_1^{\lambda_1}, \dots, x_n^{\lambda_n}$ in the Schur function s_μ is the number of tableaux of shape μ and content λ, when λ is equal to 1^n this counts the number of standard tableaux of size n. Using the fact that $\sum_\lambda s_\lambda = h(e_1 + e_2)$, we have

Theorem 5. *The number y_n of standard tableaux with n entries is P-recursive.*

A similar approach yields:

Theorem 6. *Let $B_k = \sum_\lambda s_\lambda$, the sum over all partitions λ with at most k parts. Then B_k is D-finite.*

This leads us to the conclusion that the number $y_k(n)$ of standard tableaux with n entries and at most k rows is P-recursive.

The speaker presented a long and interesting list of suggestions for further work. She suggested extending the inner product to functions of several sets of variables, in order, for instance, to handle problems concerning directed graphs. She suggested the study of *morphisms* arising in the context of Theorem 3 above, and the definition of other D-finiteness-preserving morphisms.

Further: what are the q-analogues of D-finiteness? Do these concepts make sense in other symmetry classes of functions (skew-symmetric, quasi-symmetric)? When such morphisms exist, what are their corresponding differential equations?

We admire the speaker's skill and courage in undertaking new work in a field already harvested by Gessel, Stanley, and Zeilberger.

Bibliography

[1] Gessel (Ira M.). – Symmetric functions and P-recursiveness. *Journal of Combinatorial Theory, Series A*, vol. 53, n° 2, 1990, pp. 257–285.

Part II

Symbolic Computation

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 29–32.

Available online at the URL
http://algo.inria.fr/seminars/.

Computation of the Inverse and Determinant of a Matrix

Gilles Villard
CNRS and LIP, ENS Lyon (France)

May 27, 2002

Summary by Emmanuel Thomé

Abstract

We investigate here the complexity of different computational problems related to linear algebra, under several models. We see how these complexities are related to each other, and how in most cases they can be shown to be very closely related to the compexity of matrix multiplication.

1. Introduction

Most algorithms advertised here are of probabilistic nature, and the reader must be aware of this fact. More precisely, no distinction will be made between Monte Carlo- or Las Vegas-type probabilistic algorithms. Our major concern is that these algorithms are usable in practice (implementations have been made for most of them). When comparing algorithms, we are only interested in the major contribution to the complexity. We shall ignore constants as well as logarithmic factors in complexity estimates. The notation $O\tilde{}(x)$ will be used to reflect this consideration. Furthermore, the use of fast Fourier transform arithmetic is always assumed when applicable.

1.1. **Definitions.** We fix notation for the complexities of different computations that can be done on matrices of size n over a domain R (which will be either a Euclidean domain or a finite field), where the meaning of "complexity" depends on the computational model that will be chosen later on:

- $\mathsf{DET}(n)$: computing the determinant of a matrix of size n;
- $\mathsf{INV}(n)$: computing the inverse (when it exists) of a matrix of size n;
- $\mathsf{LINSYS}(n)$: solving a linear system of n equations with n unknowns;
- $\mathsf{MM}(n)$: multiplying two matrices of size n.

As the complexity for multiplying two matrices expressed in terms of arithmetic operations is crucial to many respects, it will appear in several places. We will use w for the exponent associated to this problem. Currently, the algorithm of Coppersmith–Winograd for matrix multiplication has the lowest aymptotical estimate, $O\big(n^w\big)$ with $w \approx 2.38$.

2. Classical Complexity Results under Different Models

2.1. **Algebraic model.** Here we assume that R is a field (therefore division is possible) and we express complexities in terms of arithmetic operations in R. The following facts are known:

Proposition 1 (Strassen, 1969)**.** $\mathsf{DET}(n) \prec \mathsf{MM}(n)$ and $\mathsf{LINSYS}(n) \prec \mathsf{MM}(n)$.

Proposition 2 (Strassen, 1973, Baur & Strassen, 1983)**.** $\mathsf{MM}(n) \prec \mathsf{DET}(n)$.

An easy consequence of this fact is that under the arithmetic model, the complexities of DET and MM are the same. Using the asymptotically fastest matrix multiplication algorithms, a complexity of n^w is therefore possible. Furthermore, the results above also imply that LINSYS(n) can be solved in at most the same complexity. It is not known, however, whether one can do better than n^w for LINSYS(n), and this has been an open problem for soon thirty years.

2.2. **Bit complexity.** All bit operations are equally counted under that model. For instance, this is the relevant model when computations are carried over the integers. The size of the input becomes important beyond the only consideration of the dimension of the matrix. The length of the coefficients is important too, therefore we introduce $\|A\|$, the norm of A, as the biggest of its coefficients (in absolute value). The determinant of an $n \times n$ matrix A is an integer of size $O\big(n \log \|A\|\big)$ and can be computed in time:

$$\mathsf{DET}(n) = \tilde{O}\big(n^{w+1} \log \|A\|\big).$$

The above complexity result is obtained by using the Chinese remainder theorem: the result is evaluated modulo a sufficiently large set of primes, and then recovered by interpolation.

As for linear systems, we have a much better result, since LINSYS(n) can be solved in complexity $\tilde{O}\big(n^w \log^2 \|A\|\big)$ using p-adic (Hensel) lifting. This result is extensively explained in Storjohann's thesis [6] as well as in [7].

2.3. **Division-free complexity.** The complexity results given in the algebraic model above assume that division is allowed and the results of Strassen quoted above do make use of this fact. What happens if we remove the possibility of computing inverses? For instance, if R is a Euclidean domain and not a field, inverses cannot be computed whereas the determinant is well-defined.

The following trick might help to carry the results concerning the arithmetic model to the division-free model, with an impact on the complexity. Suppose for instance that we want to compute the determinant of the matrix A over the domain R. The idea is to work in $R[[u]]$ (the ring of formal power series over R), setting $B(u) = I + u(A - I)$. Therefore $B(0) = I$ and $B(1) = A$. The computation of the determinant of B using the recursive factorization algorithm of Strassen is possible, and requires only the computation of inverses of elements in $1 + uR[[u]]$. Since

$$\frac{1}{1-z} = 1 + z + z^2 + \ldots = (1+z)(1+z^2)(1+z^4)\ldots,$$

it is possible to compute this quantity without inverting elements of R. Since the result obtained is necessarily a polynomial in $R[u]$ of degree n, we can afford to carry computations only modulo u^{n+1}. Final evaluation of this polynomial at $u = 1$ yields $\det A$. The consequence of this is that we obtain a complexity of $\tilde{O}\big(n^{w+1}\big)$ for computing a determinant in the division-free model.

2.4. **Black-box (sparse) complexity.** We say that we do black-box computations with a given matrix when no investigation is made on the inner structure of the matrix and the matrix is merely used in one single operation: multiplication by a vector. In theory, this operation requires $O\big(n^2\big)$ multiplications, but when the matrix has the property of being *sparse* (as few as $O(\log n)$ non-zero coefficients per row), then the cost can be for instance $\tilde{O}(n)$.

Under this model, the computation of the determinant or the characteristic polynomial is done in $\tilde{O}\big(n^w\big)$, which is no better than the arithmetic complexity. However, the computation of the minimal polynomial, and therefore the solution of a linear system, can be obtained much more efficiently. Algorithms gathered under the name of "Krylov subspace techniques" are quite successful for this purpose. The Lanczos algorithm can be regarded as the Gram–Schmidt process for finding

a self-orthogonal vector. The Wiedemann algorithm is another method that can be quickly described. Choose two random vectors u and v. Then compute the minimum generating polynomial of the scalar sequence $v^T u$, $v^T Au$, ..., $v^T A^k u$, ... With high probability (on u, v, and A), this polynomial is also the minimum polynomial of the sequence I, A, ..., A^k, ..., which is also the minimum polynomial of A. Since the minimum generating polynomial of a scalar sequence can be computed in time $O(n^2)$ using the Berlekamp–Massey algorithm, using the first $2n$ coefficients of the sequence, it follows that the minimum polynomial can be computed in time $O^{\sim}(n^2)$.

We will see later that in fact the relative easiness of the computation of the minimum polynomial can be well understood in terms of the invariant factors of the matrix, of which the minimum polynomial is the biggest, while the characteristic polynomial is their product.

2.5. **Overview of classical costs.** If we summarize these results, focusing only on the dominating term and taking the size of the inputs constant, we obtain:

	Arithmetic	Division free	Binary	Sparse
LINSYS	n^w	–	n^w	n^2
DET	n^w	n^{w+1}	n^{w+1}	n^w

3. Recent Progress (2000–2002)

Recent works in the field have contributed to the following improvements:

	Division free	Binary (general)	Binary (polynomial)	Sparse
LINSYS	–	n^w	$n^w d$	n^2
DET	n^{w+1} ↓ $n^{2.698}$ [4, 5]	n^{w+1} ↓ n^w [2, 4, 7]	$n^{w+1}d$ ↓ $n^3 d$ (INV) [3]	n^w ↓ $n^{2.25}$ [8, 10]

Two examples will be presented here.

3.1. **Computing the inverse of a polynomial matrix.** Given a polynomial $n \times n$ matrix A of degree d over a field K, we present here an algorithm that computes the inverse of A in time $O^{\sim}(n^3 d)$, which is in fact the size of the output.

It should be noted that a closely related result has been obtained recently by Storjohann [7], who shows that the determinant of A can be computed in time $O^{\sim}(n^w d)$. It is not known, however, whether the computation of the inverse reduces to the computation of the determinant. On the other hand, our algorithm can be derived to obtain another method for computing the determinant (although not as efficient as Storjohann's).

The polynomial inverse computation algorithm described here is a block divide-and-conquer procedure. If the input matrix has the form $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$, assuming that the input is generic, we proceed through the following steps for computing the left inverse of the matrix:

1. Compute cofactors $U(X)$ and $V(X)$ such that the equality $U(X)A(X) + V(X)C(X) = 0$ holds. This can be achieved by a matrix Euclidean algorithm, using [1] or [9] for instance, in time $O^{\sim}(n^3 d)$. With generic input, the degrees of $U(X)$ and $V(X)$ are less than or equal to d.
2. Similarly, obtain cofactors $S(X)$ and $T(X)$ such that we have $S(X)B(X) + T(X)D(X) = 0$.
3. Multiply the input matrix on the left by $\left(\begin{smallmatrix} S & T \\ U & V \end{smallmatrix}\right)$. The resulting matrix is block-diagonal, and the blocks have degree $2d$ at most.
4. Compute the inverse of the blocks on the diagonal recursively.

All of these steps can be performed in time $O^{\sim}(n^3 d)$. There are two recursive calls for the inversion of matrices on the diagonal. These are inverses of $\frac{n}{2} \times \frac{n}{2}$ matrices of degree $2d$. The resulting complexity can therefore easily be shown to be equal to $O^{\sim}(n^3 d)$.

We have assumed that the degrees of $U(X)$ and $V(X)$ were balanced, which is true for most inputs, but might fail for some particular matrices. In such a case, it is possible to precondition (multiply on the left by some matrix whose inverse is known) the input matrix so that the required conditions are met.

3.2. **Computing the characteristic polynomial and invariant factors of a sparse matrix.** If A is a sparse matrix, we have seen already that its minimal polynomial can be easily computed in time $O^{\sim}(n^2)$. Using this, we will now see that much more can be obtained. We define the invariant factors of A (denoted $f_1, \ldots, f_n$) to be the coefficients of the diagonal on the Smith normal form $S(X)$ of the polynomial matrix $A - XI$. Recall that the Smith normal form is the unique diagonal matrix such that $S(X) = U(X)(A - XI)V(X)$, where U and V have determinant 1, and the coefficients on the diagonal divide each other.

Obviously, the characteristic polynomial of A is the product of its invariant factors. Since these invariant factors divide each other, it is easy to see that at most $\sqrt{n}$ of them are distinct. We show that it is possible to discover the chain of distinct invariant factors and their multiplicities by a divide-and-conquer procedure, with at most $O(\sqrt{n})$ computations of individual invariant factors. The crux of this algorithm is the ability to compute f_k just as easily as we are already able to find the minimal polynomial f_n. To achieve this, we simply compute the gcd of f_n with the minimal polynomials of $A+B$, where B is a matrix of rank $n-k$. This gives the desired result. The matrix B can even be chosen to be a product of two Toeplitz matrices, which eases the computations. The design of the recursive discovery procedure is then straightforward. This yields a total complexity of $O^{\sim}(n^{2.5})$. The latter can be lowered to $O^{\sim}(n^{2.25})$ using block techniques.

Bibliography

[1] Beckermann (Bernhard) and Labahn (George). – A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM Journal on Matrix Analysis and Applications*, vol. 15, n° 3, 1994, pp. 804–823.

[2] Eberly (Wayne), Giesbrecht (Mark), and Villard (Gilles). – On computing the determinant and Smith form of an integer matrix. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pp. 675–685. – IEEE Computing Society, Los Alamitos, CA, 2000.

[3] Jeannerod (Claude-Pierre) and Villard (Gilles). – Computing the inverse of a polynomial matrix. – 2002. In preparation.

[4] Kaltofen (Erich) and Villard (Gilles). – On the complexity of computing determinants (extended abstract). In *Computer mathematics (Matsuyama, 2001)*, pp. 13–27. – World Scientific, River Edge, NJ, 2001.

[5] Kaltofen92 (Erich). – On computing determinants without divisions. In Wang (Paul S.) (editor), *International Symposium on Symbolic and Algebraic Computation 92*. pp. 342–349. – ACM Press, New York, 1992. Proceedings of ISSAC'92, Berkeley, California (July 27-29, 1992).

[6] Storjohann (Arne). – *Algorithms for matrix canonical forms*. – Dissertation, Swiss Federal Institute of Technology, December 2000. Diss. ETH No. 13922.

[7] Storjohann (Arne). – High-order lifting [extended abstract]. In Mora (Teo) (editor), *ISSAC 2002 (July 7-10, 2002. Université de Lille, Lille, France)*. pp. 246–254. – ACM Press, New York, 2002. Conference proceedings.

[8] Storjohann (Arne) and Villard (Gilles). – Computing the characteristic polynomial of a sparse matrix. – 2002. In preparation.

[9] Thomé (Emmanuel). – Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. *Journal of Symbolic Computation*, vol. 33, n° 5, 2002, pp. 757–775. – Computer algebra (London, ON, 2001).

[10] Villard (Gilles). – Computing the Frobenius normal form of a sparse matrix. In *Computer algebra in scientific computing (Samarkand, 2000)*, pp. 395–407. – Springer, Berlin, 2000.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 33–36.

Available online at the URL
http://algo.inria.fr/seminars/.

Fast Algorithms for Polynomial Systems Solving

Alin Bostan

GAGE, École polytechnique (France)

November 19, 2001

Summary by Frédéric Chyzak

Abstract

Solving a system of polynomial equations with a finite number of solutions can be reduced to linear algebra manipulations in an algebra A of finite type. We show how to accelerate this linear algebra phase in order to compute a "rational parameterization" of the zeros of the polynomial system. We propose new algorithmic solutions by extending ideas introduced by V. Shoup in the context of the factorization of polynomials over finite fields. The approach is based on the A-module structure of the dual of A, which translates algorithmically to techniques of the type "baby steps / giant steps." This is joint work with B. Salvy and É. Schost [1].

Given a zero-dimensional ideal I in some polynomial ring $k[X_1, \ldots, X_n]$, a nice form for a parameterization of the solution set $V(I)$ of I is of the type

$$V(I) = \left\{ \left(\frac{g_1(a)}{g(a)}, \ldots, \frac{g_n(a)}{g(a)} \right) \;\middle|\; m(a) = 0 \right\} \tag{1}$$

for polynomials m, g, and g_i. In other words, solutions are indexed by the zeros of a univariate polynomial m, and the ith coordinate of the solutions is the evaluation of the fixed rational function g_i/g at those zeros. With additional technical constraints, (1) is called a *rational parameterization* of the variety $V(I)$. Note that by the algebraic nature of the problem, polynomials could be considered in place of rational functions with common denominators, but the choice of rational parameterizations proves useful to obtain compact expressions and algorithms with low complexity.

An algebraic quantity needed in several works to solve polynomial systems is the *minimal polynomial* of suitable elements u of the quotient algebra $A = k[X_1, \ldots, X_n]/I$. (By the minimal polynomial m_u of $u \in A$, we mean the unique monic polynomial of minimal degree such that $m(u) = 0$.) For example, in the algorithms below, obtaining the polynomials g and g_i of rational parameterizations indirectly requires to compute minimal polynomials.

The goal of this work is to accelerate the computation of minimal polynomials in A and of rational parameterizations of the variety $V(I)$. Additional motivation is given by the need for such calculations for polynomial factorization, in cryptography, effective Galois theory, effective theory of $\mathcal{D}$-modules, when counting and approximating zeros, etc. We present several probabilistic algorithms with several types of inputs and different complexity. A first class of algorithms takes as input one or a few matrices of multiplication by selected elements of A. In the case of the calculation of minimal polynomials, the result is that our approach gains when the degree δ of the minimal polynomial is relatively small, compared to the dimension D of A as a k-vector space. In the same way, our algorithm for the calculation of rational parameterizations gains when an a priori bound δ for the degree of minimal polynomials to be computed as a subtask is sufficiently smaller

than D. An additional gain by an order of $2^{-n}\sqrt{\delta}$ is made available by algorithms that take the whole multiplication table of A.

The algorithms presented here have been implemented in the computer algebra system Magma.

1. Computation of Minimal Polynomials

We focus on the computation of the minimal polynomial of an element u of the algebra A. The cost of the naive algorithm—express the powers of u in terms of a k-basis of A before looking for a linear dependency—is dominated by the calculations of the successive powers.

A first ingredient to improve the calculation of a minimal polynomial is by projection of powers, an idea already used in other contexts by Wiedemann and Shoup. Indeed, observe that when u satisfies an algebraic relation $a_d u^d + \cdots + a_0 = 0$, then

$$a_d \ell(u^{d+i}) + \cdots + a_0 \ell(u^i) = 0$$

for any k-linear form ℓ on A and any integer i, making the sequence of the $\ell(u^i)$ linear recurrent. One thus looks for the minimal polynomial $m_{u,\ell}$ of the sequence $\mathcal{L}_u = \big(\ell(u^i)\big)_{i\geq 0}$, which for "generic" ℓ is equal to m_u. (For unlucky choices of ℓ, it is only a divisor of m_u.)

Specifically, the algorithm chooses a k-linear form ℓ from the dual A^* of A, then determines the first 2δ terms of the scalar sequence $\mathcal{L}_u$. Using the Berlekamp–Massey algorithm, it next determines $m_{u,\ell}$, which merely amounts to computing a Padé approximant for the (truncated) series $\sum_{i=0}^{2\delta-1} \ell(u^i)U^i$, as one can prove the existence of a relation

$$\sum_{i=0}^{2\delta-1} \frac{\ell(u^i)}{U^{i+1}} = \frac{G_{u,\ell}}{m_u} \tag{2}$$

for a polynomial $G_{u,\ell}$ of degree at most δ. Apart from the projection step, the complexity of this algorithm decreases from the complexity of linear algebra, in $O(\delta^\omega)$ for $2 < \omega \leq 3$, to $O(\delta^2)$. The overall complexity thus remains dominated by the calculations of the successive powers.

A second improvement consists in a better calculation of the powers, and follows a "baby steps / giant steps" approach: instead of computing the $\ell(u^i)$ in sequence for i to δ, one only computes $t = O\big(\sqrt{\delta}\big)$ powers u^i for $1 \leq i \leq t$ and evaluates them at $O\big(\sqrt{\delta}\big)$ forms of the form $x \mapsto \ell(u^{it}x)$. Computing those forms efficiently requires a better understanding of the structure of A^*. Specifically, for any $a \in A$, we consider the k-linear application of multiplication by a from A to itself, which by transposition induces a k-linear map of A^* to itself: this *transposed product* $a \cdot \ell$ maps an element $x \in A$ to $\ell(xa)$. The dual A^* thus turns out to be an A-module. Since (on suitable bases) the matrix of the (transposed) product by a in A^* is the transposed of the matrix of the multiplication by a in A, Tellegen's transposition principle predicts that the complexity of computing the transposed product by a is that of computing the multiplication by a. Guided by this heuristic, Bostan et al. have obtained an algorithm to compute all the projections and have gained essentially a factor of $\sqrt{\delta}$ on the naive complexity. For the sake of exposition, the description of this algorithm is postponed to Section 3.

Detailed complexity analysis leads to the following result.

Theorem 1. *Given $u \in A$, the minimal polynomial m_u (together with the polynomial $G_{u,\ell}$ in (2)) can be computed by a probabilistic algorithm:*

1. *in $O(\delta D^2)$ operations in k if the matrix of multiplication by u is known;*
2. *in $O(2^n\sqrt{\delta}\, D^2)$ operations in k if the multiplication table of A is known (and described in some specific way, see Section 3).*

This has to be compared with classical algorithms, respectively in $O(\delta D^2 + D^\omega)$ and $O_{\log}(D^\omega)$.

2. Computations of Rational Parameterizations

Elements of the quotient algebra $A = k[X_1, \ldots, X_n]/I$ can be viewed as functions on the variety $V(I)$. The idea behind the representation (1) is to distinguish two points by *distinct* values of a suitable polynomial function on $V(I)$. To this end, we introduce the notion of a *separating element* of A to refer to polynomial functions with this property.

The following central and, to the eye of the author of this summary, surprizing result from [1] provides rational parameterizations as a by-product of computations of minimal polynomials.

Theorem 2. *Let u be a separating element of A of generic degree and ℓ be a linear form on A such that $m_{u,\ell} = m_u$. Then, a rational parameterization of $V(I)$ is given as*

$$V(I) = \left\{ \left(\frac{G_{u,x_1\cdot\ell}(a)}{G_{u,\ell}(a)}, \ldots, \frac{G_{u,x_n\cdot\ell}(a)}{G_{u,\ell}(a)} \right) \;\middle|\; m_u(a) = 0 \right\}.$$

Then, an algorithm for rational parameterizations is the following. An element u is chosen in A, as well as a k-linear form ℓ in A^*. The minimal polynomial m_u and the related polynomial $G_{u,\ell}$ are computed by the algorithm of the previous section. The forms $x_j \cdot \ell$ are then computed. By projecting powers like in the previous section, the series $R_i = \sum_{i\geq 0} \ell(x_j u^i)/U^{i+1}$ are computed with precision $O(U^{1-\delta})$. The polynomials $G_{u,x_i\cdot\ell}$ are then obtained by mere polynomial products from the formula $G_{u,x_i\cdot\ell} = m_u R_i$, which assumes the generically verified identity $m_u = m_{u,\ell} = m_{u,x_i\cdot\ell}$. Again, the bottleneck of the calculation is the projection step.

Detailed complexity analysis leads to the following result. For non-zero characteristic, a technical condition is given in terms of the *radical* $\sqrt{I}$ of the ideal I, in other words, the set of polynomials that, when raised to some power, lie in the ideal I.

Theorem 3. *Given a separating element $u \in A$ of generic degree. Assuming that the field k is a perfect field of characteristic zero or at least $\min\{ s \mid \sqrt{I}^{\,s} \subset I \}$, a rational parameterization of $V(I)$ can be computed by a probabilistic algorithm:*

1. *in $O(\delta D^2 + nD^2)$ operations in k if the matrices of multiplication by u and the x_i are known;*
2. *in $O(n2^n\sqrt{\delta}\, D^2)$ operations in k if the multiplication table of A is known (and described in some specific way, see Section 3).*

This has to be compared with Rouillier's RUR algorithm in $O(D^3 + nD^2)$.

3. Algorithm for Effective Transposed Product

In this section, we show that both multiplications in A and transposed products in A^* can be performed in $O(2^n D^2)$ operations in k when the multiplication table of A is known. More specifically, we require that the multiplication table be described in terms of a special vectorial basis of A. A basis $\{\omega_i\}_{i=1,\ldots,D}$ of the quotient algebra $A = k[X_1, \ldots, X_n]/I$ is called a *monomial basis* when the ω_i are given as $\omega_i = m_i + I$ for a collection M of monomials m_i "under the stairs" of a Gröbner basis for I. (In particular, if m is a monomial in this collection, all its monomial divisors are there as well; this property is in fact sufficient to obtain the subsequent results.) The property of being a monomial basis, not just any basis, has a strong consequence on products: the set $M \cdot M$ of the products $m_i m_j$ has cardinality bounded above by $2^n D$. After fixing orders on M and $M \cdot M$, the multiplication table of A is given as a $|M| \times |M \cdot M|$ matrix T.

In order to compute the multiplication of $u = \sum_{i=1}^D u_i\omega_i \in A$ and $v = \sum_{i=1}^D v_i\omega_i \in A$, just compute the product of $\sum_{i=1}^D u_i m_i$ with $\sum_{i=1}^D v_i m_i$ in $k[X_1, \ldots, X_n]$, write the vector V of the

coefficients of this product with respect to the basis $M \cdot M$, and compute the product TV to get the coefficients with respect to the basis M of the product uv. This uses $O(D^2)$ operations for the first step and $O(2^n D^2)$ for the second, so in total $O(2^n D^2)$ operations in k for multiplication in A.

We now turn to the computation of transposed products, in which certain truncations of generating series play a crucial role. We introduce the notation

$$S(\ell, C) = \sum_{m \in C} \ell(m + I)m \in k[X_1, \ldots, X_n]$$

for any form $l \in A^*$ and any collection C of monomials. One readily verifies that when $M = \{m_i\}_{i=1}^D$ corresponds to a monomial basis of A and with $u = \sum_{i=1}^D u_i m_i + I$, the polynomial $S(u \cdot \ell, M)$ is given as the part with support in M of the product

$$\left(\sum_{i=1}^{D} u_i m_i^{-1} \right) S(\ell, M \cdot M). \tag{3}$$

An algorithm to compute a transposed product $u \cdot \ell$ is thus the following. Write $\ell = \sum_{i=1}^D \ell_i \omega_i^*$ in terms of the dual basis $\{\omega_i^*\}_{i=1}^D$ of the monomial basis of A. Multiply this vector by the transposed of the matrix T that encodes the multiplication table of A. The resulting vector gives the coefficients of $S(\ell, M \cdot M)$ with respect to the basis $M \cdot M$. Compute the product in (3) and read off the coefficients of the form $u \cdot \ell$ with respect to the dual basis $\{\omega_i^*\}_{i=1}^D$. Again, this uses $O(2^n D^2)$ operations in k in total.

4. **A Note on the Probabilistic Nature of the Algorithms**

The algorithms of Sections 1 and 2 are randomized by the choice of the linear form ℓ. If the coordinates of ℓ are chosen in a finite subset F of k, then the probability of failure of the algorithms is bounded above by $\delta/|F|$ (uniformly in the input u in the case of the minimal polynomial computation).

Bibliography

[1] Bostan (Alin), Salvy (Bruno), and Schost (Éric). – Fast algorithms for zero-dimensional polynomial systems using duality. – To appear in *Applicable Algebra in Engineering, Communication and Computing*.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 37–42.

Available online at the URL
http://algo.inria.fr/seminars/.

Transseries Solutions of Algebraic Differential Equations

Joris van der Hoeven
CNRS, Université Paris-Sud (France)

May 27, 2002

Summary by Anne Fredet

Abstract

Transseries are series defined using exponential and logarithmic variables. They were first introduced to describe very general types of strongly monotonic asymptotic behaviour. The functions that are considered do not present any oscillatory phenomenon. An algorithm is presented that computes transseries solutions of algebraic differential equations with transseries coefficients.

1. Introduction to Transseries

1.1. **Well-ordered and grid-based transseries.** The transseries are a generalization of the usual formal power series, allowing the recursive introduction of exponential and logarithmic variables (see [1] or [3] and references).

Example. The following series are transseries:

- $1 + x^{-1} + x^{-2} + x^{-e} + x^{-3} + x^{-e-1} + \cdots = \frac{1}{1-x^{-1}-x^{-e}}$,
- $1 + \frac{1}{x} + \frac{1}{x^2} + \cdots + e^{-x} + \frac{e^{-x}}{x} + \cdots + e^{-2x} + \cdots$,
- $\frac{1}{x} + \frac{1}{x^2} + \cdots + \frac{1}{e^{\log^2 x}} + \frac{1}{e^{2\log^2 x}} + \cdots + \frac{1}{e^{\log^4 x}} + \frac{1}{e^{2\log^4 x}} + \cdots$ solution of $f(x) = \frac{1}{x} + f(x^2) + f(e^{\log^2 x})$,
- $1 + 2^{-x} + 3^{-x} + 4^{-x} + \cdots$,
- $x^{-1} + x^{-\pi} + x^{-\pi^2} + x^{-\pi^3} + \cdots$,
- $x + \sqrt{x} + \sqrt{\sqrt{x}} + \sqrt{\sqrt{\sqrt{x}}} + \cdots$,
- $e^{e^x + \frac{e^x}{x} + \frac{e^x}{x^2} + \cdots} + x^{-1} e^{e^x + \frac{e^x}{x} + \frac{e^x}{x^2} + \cdots} + \cdots$,
- $\Gamma(x - \pi) + \log\Gamma(e^{\Gamma(x^2)}) x^{x^{x^x}}$,
- $e^{\sqrt{x} + e^{\sqrt{\log x} + e^{\sqrt{\log\log x} + \cdots}}}$.

An *ordered ring* is a ring A, together with an order $\leq$ which is compatible with the ring structure. This means that: *(i)* $(x \leq y$ and $x' \leq y') \Rightarrow x + x' \leq y + y'$, *(ii)* $0 \leq 1$, and *(iii)* $(0 \leq x$ and $0 \leq y) \Rightarrow 0 \leq xy$. The *absolute value* $|x|$ of $x \in A$ is defined by $|x| = x$ if $x \geq 0$ and $-x$ otherwise. One writes $x \prec y$ if $|\lambda x| \leq |\mu y|$ for some $\mu \in A$ and all $\lambda \in A$ and one says that x is *negligible* with respect to y.

More generally, let C be a constant field with a total order (i.e., either $\alpha = \beta$ or $\alpha < \beta$ or $\alpha > \beta$ for all α and β in C—see [7]) and $\mathfrak{M}$ be a group with a total order $\succcurlyeq$. A *well-ordered* transseries is a mapping $f : \mathfrak{M} \to C$ with well-ordered support (this means that every nonempty subset of the support of f has a least element—see [7]). The elements of C are called *coefficients* and the elements of $\mathfrak{M}$ are *monomials*. If $f = \sum_{\mathfrak{m} \in \mathfrak{M}} f_{\mathfrak{m}} \mathfrak{m}$ is a well-ordered transseries and $\mathfrak{m} \in \mathfrak{M}$ then

one says that $f_{\mathfrak{m}}$ is the *coefficient* of $\mathfrak{m}$ in f and $f_{\mathfrak{m}}\mathfrak{m}$ is a *term* occurring in f. Since the support of f is well-ordered, it admits a maximal element $\mathfrak{d}_f$ which is called the *dominant monomial*. If $f = c_f\mathfrak{d}_f(1+\delta_f)$ then $c_f\mathfrak{d}_f$ is the *leading term* of f and one denotes that $f \preccurlyeq g$ if and only if $\mathfrak{d}_f \preccurlyeq \mathfrak{d}_g$. One decomposes

$$f = f^{\uparrow} + f^{=} + f^{\downarrow}$$

with

$$f^{\uparrow} = \sum_{\mathfrak{m}\succ 1} f_{\mathfrak{m}}\mathfrak{m}, \qquad f^{=} = f_1, \qquad f^{\downarrow} = \sum_{\mathfrak{m}\prec 1} f_{\mathfrak{m}}\mathfrak{m}.$$

One focuses on particular transseries:

Definition 1. A series f is *grid-based* if there are $\mathfrak{m}_1, \dots, \mathfrak{m}_k \prec 1$ and $\mathfrak{n} \in \mathfrak{M}$ such that

$$\operatorname{supp} f \subset \{\mathfrak{m}_1, \dots, \mathfrak{m}_k\}^*\mathfrak{n}$$

One denotes by $C[\![\mathfrak{M}]\!]$ the set of mappings from $\mathfrak{M}$ to C with grid-based support and one calls it the set of grid-based transseries. One remarks that $C[\![\mathfrak{M}]\!] \neq C[[\mathfrak{M}]]$.

Example. If $f = x^2 + x + 1 + x^{-1} + \cdots$ then $\operatorname{supp} f \subseteq \{x^{-1}\}^* x^2$

The field of the grid-based transseries in x over C is denoted by $\mathbb{T}$ and is stable under derivation, composition and functional inversion as proved in [3]. Ways to construct $\mathbb{T}$ are presented in [3].

1.2. **Transbasis.**

Definition 2. An ordered set of transseries $\mathfrak{B} = (\mathfrak{b}_1, \dots, \mathfrak{b}_n)$ is a *transbasis* if the following conditions are satisfied:

1. $\mathfrak{b}_1$ is an iterated logarithm or exponential: $\mathfrak{b}_1 = \exp_l x$ for some $l \in \mathbb{Z}$ (l is the *level* of the transbasis);
2. $1 \prec \mathfrak{b}_1 \prec \cdots \prec \mathfrak{b}_n$;
3. $\mathfrak{b}_i \in \exp C[\![\mathfrak{b}_1, \dots, \mathfrak{b}_{i-1}]\!]$ for $i > 1$.

Example. The sets $\mathfrak{B}_1 = \{x^{-1}, e^{-x}, e^{-x^2}, e^{-x^3}\}$ and $\mathfrak{B}_2 = \{\log^{-1} x, x^{-1}, e^{-\log^2 x}, e^{-x}, e^{-e^x/(1+x^{-1})}\}$ are transbasis but $\mathfrak{B}_3 = \{x^{-1}, e^{-x+e^{-x}}\}$ is not because $e^{-x+e^{-x}}$ is not in $\exp C[\![x^{-1}]\!]$.

One says that a transseries f can be expanded with respect to $\mathfrak{B}$ if $f \in C[\![\mathfrak{b}_1, \dots, \mathfrak{b}_n]\!]$. Equivalently, one says that $\mathfrak{B}$ is a transbasis for f.

Example. $\log(x + e^{\frac{-x^2}{1-x^{-1}}}) \in C[\![\log x; x; e^{x^2+x}]\!]$ and then $\mathfrak{B} = \{\log x; x; e^{x^2+x}\}$ is a transbasis for f.

For any $f \in C[\![\mathfrak{b}_1, \dots, \mathfrak{b}_n]\!]$, one can recursively expand f: $f = \sum_{\alpha_n} f_{\alpha_n}\mathfrak{b}_n^{\alpha_n}$ where $f_{\alpha_n} = \sum_{\alpha_{n-1}} f_{\alpha_n,\alpha_{n-1}}\mathfrak{b}_{n-1}^{\alpha_{n-1}}$, where…, where $f_{\alpha_n,\dots,\alpha_2} = \sum_{\alpha_1} f_{\alpha_n,\dots,\alpha_1}\mathfrak{b}_1^{\alpha_1}$.

Theorem 1. *Let f be a transseries and let $\mathfrak{B}_0$ be a transbasis. Then there exists a transbasis $\mathfrak{B}$ for f which contains $\mathfrak{B}_0$.*

1.3. **Differentiation and shifting.** Right compositions by exp (resp. log) are referred to by *upward shifting* (resp. *downward shifting*). The upward (resp. downward) shifting of $f \in \mathbb{T}$ is denoted by $f \circ \exp = f\uparrow$ (resp. $f \circ \log = f\downarrow$). One observes that $\uparrow$ and $\downarrow$ are scale changes which preserve the set of transmonomials. Note that $f\uparrow \neq f^{\uparrow}$ and $f\downarrow \neq f^{\downarrow}$. These compositions are used to consider transbasis starting with level one ($\mathfrak{b}_1 = x$) which is particularly useful for differential calculus (see below).

1.4. **A conjecture of Hardy.** In [2] a conjecture states that the functional inverse of $\log x \log\log x$ is not equivalent to any exp-log function over $\mathbb{R}$ for $x \to \infty$. Theorem 1.2 of [3] illustrate the interest of transseries by a proof of this conjecture.

2. Differential Algebraic Polynomials

Let $P = \sum_d P_d$ be a differential algebraic polynomial where

$$P_d = \sum_{i_0+\cdots+i_r=d} P_{i_0,\ldots,i_r} f^{i_0} \ldots f^{(r)^{i_r}}.$$

One defines:

- the *degree* of P, $\deg P = \max\{\, i_0 + \cdots + i_r \mid P_{i_0,\ldots,i_r} \neq 0 \,\}$,
- the *additive conjugate*, $P_{+h}(f) = P(f+h)$,
- the *multiplicative conjugate*, $P_{\times h}(f) = P(fh)$,
- the *upward shifting*, $P\uparrow(f\uparrow) = P(f)\uparrow$,
- the *dominant monomial*, $\mathfrak{d}_P = \max\{\, \mathfrak{d}_{P_{i_0,\ldots,i_r}} \mid \mathfrak{d}_{P_{i_0,\ldots,i_r}} \neq 0 \,\}$,
- the *dominant coefficient*, $D_P = \sum_{i_0,\ldots,i_r} P_{i_0,\ldots,i_r,\mathfrak{d}_P} c^{i_0} \ldots c^{(r)^{i_r}}$, where c is a variable.

2.1. **Differential Newton polynomials.** One now describes an algorithm for the resolution of algebraic differential equations with transseries coefficients like

$$P(f) = 0 \qquad (f < \mathfrak{v}) \tag{1}$$

where $P \in \mathbb{T}[f, f', \ldots, f^{(r)}]$ is a differential polynomial with transseries coefficients and $\mathfrak{v} \in \mathfrak{M}$ a transmonomial. The first step is to construct an analogue of the Newton polygon and polynomial method in this setting, enabling us to compute the successive terms of solutions one by one.

The following theorem shows how D_P looks like after sufficiently many upward shifting.

Theorem 2. *There exist an integer $k \leq \deg(P)$ and a polynomial N_P depending only on the variables c and c' such that for any $l \geq k$, $D_{P\uparrow_l} = N_P$.*

Example. If one considers $P = ff'' - f'^2$ and one denotes $\tilde{f}(x) = f\uparrow = f(e^x)$ then one has $\tilde{f}'(x) = e^x f'(e^x)$ and $\tilde{f}''(x) = e^x f'(e^x) + e^{2x} f''(e^x)$. This implies that $f(e^x) = \tilde{f}(x)$, $f'(e^x) = e^{-x}\tilde{f}'(x)$, $f''(e^x) = e^{-2x}\left(\tilde{f}''(x) - \tilde{f}'(x)\right)$ and $P(f)\uparrow = e^{-2x}\left(\tilde{f}\tilde{f}'' - \tilde{f}\tilde{f}' - \tilde{f}'^2\right) = P\uparrow(\tilde{f}) = P\uparrow(f\uparrow)$. So one deduces that

$$P\uparrow = e^{-2x}\left(ff'' - ff' - f'^2\right).$$

Using the same method, one finds that $P\uparrow\uparrow = e^{-2e^x - x}(ff') + e^{-2e^x - 2x}(ff'' - ff' - f'^2)$. This implies that $N_P = cc'$.

N_P is the *differential Newton polynomial of P*. More generally, given a monomial $\mathfrak{m}$, $N_{P\times\mathfrak{m}}$ is the *differential Newton polynomial of P associated to $\mathfrak{m}$*. The *Newton degree* of (1) is the largest possible degree of $N_{P_{\times\mathfrak{m}}}$ for all the monomials $\mathfrak{m} \prec \mathfrak{v}$. In the algebraic case, the Newton degree measures the number of solutions to the asymptotic equation when counting with multiplicities. In the differential case, it only gives a lower bound (see Theorem 1 of [6]). Also, an equation of degree zero does not admit any solutions.

2.2. **Potential dominant monomials of solutions.** One is now interested by the le ading terms of a solution f to the asymptotic differential equation (1). One calls $\mathfrak{m} \prec \mathfrak{v}$ a *potential dominant monomial* if $N_{P_{\times\mathfrak{m}}} \notin C$. If $c \in C$ is such that $c\mathfrak{m} \prec \mathfrak{v}$ and $N_{P_{\times\mathfrak{m}}}(c) = 0$ then the corresponding term $c\mathfrak{m}$ is called a *potential dominant term*.

A potential dominant monomial $\mathfrak{m}$ is said to be *algebraic* if $N_{P_{\times\mathfrak{m}}} \in C[c] \setminus C$, *differential* if $N_{P_{\times\mathfrak{m}}} \in C[c']$. A potential dominant monomial involving both c and c' in $C[c, c'] \setminus (C[c] \cup C[c'])$ is said to be *mixed*.

The algebraic potential dominant monomials correspond to the slopes of the Newton polygon in a non differential setting. However, they can not be determined directly as a function of the dominant monomials of the P_i, because there may be some cancellation of terms in the different homogeneous parts during multiplicative conjugation. The algebraic potential dominant monomials are determined by successive approximation:

Proposition 1. *Let i, j be such that $P_i \neq 0$ and $P_j \neq 0$. There exists a unique monomial $\mathfrak{m}$ such that $N_{(P_i+P_j)_{\times\mathfrak{m}}}$ is non homogeneous.*

This unique monomial is called an *equalizer* or the (i, j)-equalizer for P. An algebraic potential dominant monomial is necessarily an equalizer (see [5]). Consequently, there are only a finite number of algebraic potential dominant monomials. In the proof of proposition 5.3 in [5], the author gives a method to compute such monomials.

Example. Consider the algebraic differential equation

$$P(f) = f + ff'' - f'^2 \tag{2}$$

One starts by computing the potential dominant monomials of f. One first has to find the $(1, 2)$-equalizer relative to 2. Since D_{P_2} must be in $c^{\mathbb{N}}(c')^{\mathbb{N}}$ one cannot have $N_{P_2} = P_2$ so one has to compute

$$P\uparrow = f + e^{-2x}\left(-ff' + ff'' - f'^2\right)$$

In order to equalize $P\uparrow_1$ and $P\uparrow_2$ one conjugates P multiplicatively with e^{2x}:

$$\begin{aligned} P\uparrow_{\times e^{2x}} &= fe^{2x} + e^{-2x}\left(-fe^{2x}(fe^{2x})' + fe^{2x}(fe^{2x})'' - (fe^{2x})'^2\right) \\ &= e^{2x}\left(f - 2f^2 - ff' + ff'' - f'^2\right) \end{aligned}$$

One has

$$P\uparrow_{\times e^{2x}\uparrow} = e^{2x}(f - 2f^2) - e^x(ff') + \left(ff'' - ff' - f'^2\right)$$

One observes that $D_{P\uparrow_{\times e^{2x}\uparrow}} = c - 2c^2 \in C[c]$ so one has found the $(1, 2)$-equalizer which is $\mathfrak{e} = e^{2x}\downarrow = x^2$. Since $N_{P_{\times\mathfrak{e}}} = c - 2c^2$ the corresponding algebraic potential dominant term of f is $\tau^{\mathrm{alg}} = \frac{1}{2}x^2$.

In order to find the differential potential dominant monomials, it suffices to consider P_i since $N_{P_{\times\mathfrak{m}},i} = N_{P_{i,\times\mathfrak{m}}}$ if $c'|N_{P_{\times\mathfrak{m}}}$ and $N_{P_{\times\mathfrak{m}}} \neq 0$. One rewrites $P_i = R_{P,i}(f^\dagger)f^i$ where the order of $R_{P,i}$ in $f^\dagger = f'/f$ is less than or equal to 1 and calls $R_{P,i}$ the *ith Riccati equation associated to P*.

Proposition 2. *The monomial $\mathfrak{m} \prec \mathfrak{v}$ is a potential dominant monomial of f w.r.t. $P_i(f) = 0$ if and only if*

$$R_{P,i,\frac{\mathfrak{m}'}{\mathfrak{m}}}(f^\dagger) \qquad \left(f^\dagger \prec \frac{1}{x \log x \log\log x \cdots}\right)$$

has strictly positive Newton degree.

Example. Consider the algebraic differential equation (2) given in the previous example. One has

$$R_{P,1} = 1, \qquad R_{P,2} = f^{\dagger\prime}.$$

$R_{P,1}$ has no roots and $R_{P,2}(f^\dagger) = 0$ has all constants $\lambda \in C$ as its solutions modulo $\frac{1}{x \log x \log\log x \cdots}$. Consequently $e^{\lambda x}$ is a potential dominant monomial of f for all $\lambda \in C$ such that $e^{\lambda x} \succ 1$. The corresponding differential dominant terms are of the form $\tau^{\mathrm{diff}}_{\mu,\lambda} = \mu e^{\lambda x}$ with $\mu \neq 0$ and $e^{\lambda x} \succ 1$.

2.3. **Quasi-linear differential operators and distinguished solutions.** The equation (1) is *quasi-linear* if its Newton degree is one. A solution f to such an equation is said to be *distinguished* if $f_{\mathfrak{d}_{\tilde{f}-f}} = 0$ for all other solutions $\tilde{f}$ to (1).

Theorem 3 (Theorem 6.3 of [5])**.** *Assume that the equation (1) is quasi-linear. Then it admits a distinguished transseries solution.*

2.4. **Other terms of solutions.** Using the previous results, one knows how to determine the potential dominant terms of solutions to (1). One is now interested in obtaining more terms. A *refinement* is a change of variables together with an asymptotic constraint $f = \phi + \tilde{f} \quad (\tilde{f} \prec \mathfrak{v})$. Such refinement transforms (1) into

$$P_{+\phi}(\tilde{f}) = 0 \qquad (\tilde{f} \prec \mathfrak{v}). \tag{3}$$

Proposition 3. *Let τ be the dominant term of ϕ. The Newton degree of (3) is the multiplicity of τ as potential dominant term in (1).*

Example. In order to find more terms of the solution to (2) one has to refine the equation. First of all, consider the refinement associated to the algebraic potential dominant term,

$$f = \tau^{\mathrm{alg}} + \tilde{f} \qquad (\tilde{f} \prec \tau^{\mathrm{alg}}),$$

which transforms (2) into

$$2\tilde{f} - 2x\tilde{f}' + \frac{1}{2}x^2\tilde{f}'' + \tilde{f}\tilde{f}'' - \tilde{f}'^2 = 0 \qquad (\tilde{f} \prec x^2). \tag{4}$$

Since $P_0 = 0$ one first observes that $f = \frac{1}{2}x^2$ is actually a solution of (2). Since $\frac{1}{2}x^2$ is a potential dominant term of multiplicity 1 of f, the Newton degree of (4) is one. The only potential dominant monomials of $\tilde{f}$ therefore necessarily correspond to solutions modulo $\frac{1}{x \log x \log\log x}$ of the Riccati equation

$$2 - 2xf^\dagger + \frac{1}{2}x^2\left(f^{\dagger 2} + f^{\dagger\prime}\right) = 0$$

These solutions are of the form $f^\dagger = \frac{1}{x} + \cdots$ and $f^\dagger = \frac{4}{x} + \cdots$ which leads to the potential dominant monomials x and x^4 from which one removes x^4 since $x^4 \not\prec x^2$. Expanding one term further, one sees that the generic solution to (4) is

$$\tilde{f} = \lambda x + \frac{\lambda^2}{2}$$

with $\lambda \in C$ where the case $\lambda = 0$ recovers the previous solution. So

$$f = \frac{1}{2}x^2 + \lambda x + \frac{\lambda^2}{2}$$

is the first type of generic solution to (2). As to the second case, we consider the refinement

$$f = \tau_{\mu,\lambda}^{\mathrm{diff}} + \tilde{f} \qquad (\tilde{f} \prec \tau_{\mu,\lambda}^{\mathrm{diff}})$$

which transforms (2) into

$$\mu e^{\lambda x} + \left(\lambda^2 f - 2\lambda f' + f''\right)\mu e^{\lambda x} + f + \tilde{f}\tilde{f}'' - \tilde{f}'^2 = 0 \qquad (\tilde{f} \prec \mu e^{\lambda x}) \tag{5}$$

This equation has Newton degree one and one observes that the linear part of this equation only admits solutions with dominant monomial $e^{\lambda x}$ or $xe^{\lambda x}$. Consequently (5) admits at most one solution. By Theorem 3 one knows that quasi-linear equations always admit at least one solution. This leads to the following second type of generic solution to (2):

$$f = \mu e^{\lambda x} - \frac{1}{\lambda^2} + \frac{1}{4\mu\lambda^4} e^{-\lambda x}$$

For this example, we found exact solutions but the expansion are infinite in general.

3. A Differential Intermediate Value Theorem

Theorem 4 ([4]). *Let P be a differential polynomial with coefficients in $\mathbb{T}$. Given $\varphi < \psi$ in $\mathbb{T}$ such that $P(\varphi)P(\psi) < 0$, there exists an $f \in (\varphi, \psi)$ with $P(f) = 0$.*

If there exists a differential polynomial with coefficients in $\mathbb{T}$ which admits a sign change on a non empty interval (φ, ψ) of transseries, one uses the differential Newton polygon method to shrink the interval further and further while preserving the sign change property. Ultimately, one ends up with an interval which is reduced to a point which will be seen as a zero of P.

Corollary 1. *Any algebraic equation of odd degree has at least one transseries solution.*

4. Conclusion

In [3] this approach of transseries was introduced, based on Écalle's works (see [1]). In [5] the approach is generalized to complex transseries. In particular, some results on the factorization of linear differential equation are presented. There remains some difficulties in this generalization, as to determine the differentially algebraic closure.

The transseries formalism could also be used to solve functional equations, and the multiple results should be extended to such operators.

Bibliography

[1] Écalle (Jean). – *Introduction aux fonctions analysables et preuve constructive de la conjecture de Dulac.* – Hermann, Paris, 1992, *Actualités Mathématiques. [Current Mathematical Topics]*, ii+340p.

[2] Hardy (G. H.). – Properties of logarithmico-exponential functions. *Proceedings of the London Mathematical Society*, vol. 10, 1911, pp. 54–90.

[3] van der Hoeven (Joris). – *Asymptotique automatique.* – PhD thesis, Université Paris VII, 1997.

[4] van der Hoeven (Joris). – *A differential intermediate value theorem.* – Technical Report n° 2000-50, Université d'Orsay, 2000.

[5] van der Hoeven (Joris). – *Complex transseries solutions to algebraic differential equations.* – Technical Report n° 2001-34, Université d'Orsay, 2001.

[6] van der Hoeven (Joris). – A new zero-test for formal power series. In Mora (Teo) (editor), *ISSAC 2002 (July 7-10, 2002. Université de Lille, Lille, France).* pp. 117–122. – ACM Press, New York, 2002. Conference proceedings.

[7] von zur Gathen (Joachim) and Gerhard (Jürgen). – *Modern computer algebra.* – Cambridge University Press, New York, 1999, xiv+753p.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 43–46.

Available online at the URL
http://algo.inria.fr/seminars/.

Recent Algorithms for Solving Second-Order Differential Equations[†]

Jacques-Arthur Weil
LACO, Université de Limoges — CAFE Project, INRIA

January 14, 2002

Summary by Michèle Loday-Richaud

We consider a second order ordinary linear differential equation

$$\tilde{L}y \equiv \partial^2 y + A_1(x)\partial y + A_2(x)y = 0 \tag{1}$$

with rational coefficients A_1, $A_2 \in C(x)$, over a constant field C which is assumed to be of characteristic zero and algebraically closed. We denote $\partial = d/dx$ and $K = C(x)$.

After a suitable change of variable $y \mapsto ye^{\int -a/2}$ Equation (1) is changed into the *reduced form*

$$L_r y \equiv \partial^2 y - r(x)y = 0 \qquad \text{where} \qquad r(x) = \frac{A_1^2}{4} + \frac{A_1'}{2} - A_2. \tag{2}$$

Given two linearly independent solutions of (1), say y_1, y_2, either formal or actual, the differential field generated by K, y_1, and y_2 is called a *Picard-Vessiot extension* of (1). The group of K-differential automorphisms (i.e., of field automorphisms leaving K pointwise fixed and commuting with ∂) is called the *differential Galois group of* (1) *over* K. We denote it by $G(\tilde{L}) = \mathrm{Gal}_K(\tilde{L})$ and by $PG(\tilde{L}) = G(\tilde{L})/\mathcal{Z}(G(\tilde{L})) \simeq G(\tilde{L})/(G(\tilde{L}) \cap C^*)$ the corresponding projective group.

A differential Galois group is a linear algebraic group over C; it can then be represented as a subgroup of $GL(2, C)$. In the case of an operator in reduced form L_r the differential Galois group is a special linear algebraic group over C and it can thus be represented as a subgroup of $SL(2, C)$.

The Galois correspondence states the link between properties of solutions and the form of the differential Galois group. Equation (2), hence (1), has no *Liouvillian solutions* (also called solutions *in closed form*) if and only if the differential Galois group $G(L_r)$ is isomorphic to $SL(2, C)$. At the opposite end, all solutions of (2) are algebraic if and only if the differential Galois group $G(L_r)$ is a finite group. In the case when $G(L_r) \neq SL(2, C)$ since the order is only 2, then all solutions are Liouvillian.

The Kovacic algorithm [4] is an algorithm to effectively determine whether or not a second-order linear differential equation has Liouvillian solutions with a computation of those. It can be extended to the calculation of the differential Galois group of the equation in reduced form. What follows applies to general second order differential equations in form (1) as well as form (2).

This talk is concerned with the case when the solutions are algebraic and an explicit direct computation of those. The idea consists in refering to a small amount of standard equations whose solutions have been computed once for all. Using a theorem of Klein each equation is seen as an adequate pullback of one of the standard equations. Our aim is to make this pullback explicit.

[†]Joint work with Maint Berkenbosch and Mark Van Hoeij.

1. Standard Equations

The possible projective differential Galois groups in this case are the dihedral groups $\mathbf{D}_n$ for all $n \in \mathbb{N}$, the tetrahedral group $\mathbf{A}_4$, the octahedral group $\mathbf{S}_4$, and the icosahedral group $\mathbf{A}_5$.

The standard equations in reference are hypergeometric equations

$$\mathrm{St}_G\, y \equiv \partial^2 y + \left(\frac{a}{x^2} + \frac{b}{(x-1)^2} + \frac{c}{x(x-1)} \right) y = 0$$

where the coefficients a, b, c are related to the differences λ, μ, ν of the exponents at 0, 1, and ∞ by the relations $4a = 1-\lambda^2$, $4b = 1-\mu^2$, and $4c = 1-\nu^2+\lambda^2+\mu^2$. We choose $(\lambda, \mu, \nu) = (1/2, 1/2, 1/n)$ for $PG = \mathbf{D}_n$, $(1/3, 1/2, 1/3)$ for $PG = \mathbf{A}_4$, $(1/3, 1/2, 1/4)$ for $PG = \mathbf{S}_4$, and $(1/3, 1/2, 1/5)$ for $PG = \mathbf{A}_5$.

The index G refers to the differential Galois group of the equation $\mathrm{St}_G\, y = 0$ corresponding to the chosen values of a, b, c. The solutions of these hypergeometric equations are Legendre functions.

2. Klein's Theorem

Definition 1. Let $L_1 \in C(z)[d/dz]$ and $L_2 \in C(x)[d/dx]$ be linear differential operators.

1. L_2 is a *proper pullback* of L_1 under $f \in C(x)$ if the change of variable $z = f(x)$ changes L_1 into L_2.
2. L_2 is a *pullback* of L_1 under $f \in C(x)$ if there exists $v \in C(x)$ such that $L_2 \otimes (\partial + v)$ is a proper pullback of L_1 under f.

Theorem 1. *Let L be a second order linear differential operator over $C(x)$ in reduced form with projective differential Galois group PG. Then, $PG \in \{\mathbf{D}_n, \mathbf{A}_4, \mathbf{S}_5, \mathbf{A}_5\}$ if and only if L is a pullback of St_G.*

Let L have a finite projective differential Galois group PG and suppose the standard equation with differential Galois group G has (H_1, H_2) as a C-basis of solutions. The theorem of Klein says that L is a pullback of St_G. Suppose we know f and v. Then, a C-basis of solutions of $Ly = 0$ is given by $H_1\big(f(x)\big)e^{\int v}$ and $H_2\big(f(x)\big)e^{\int v}$.

H_1 and H_2 are known for all standard equations. To get the solutions in explicit form one should then determine the projective differential Galois group and, in case it is finite, determine the pullback functions f and v. The idea is to build these quantities using semi-invariants of the equation.

3. Invariants and Semi-Invariants

Definition 2.

1. A polynomial $I(Y_1, Y_2) \in C[Y_1, Y_2]$ is called an *invariant* with respect to a differential operator L if its evaluation on a C-basis (y_1, y_2) of solutions is invariant under the action of the differential Galois group $G(L)$ of L. The rational function $h(x) = I\big(y_1(x), y_2(x)\big)$ is called the *value* of the invariant polynomial I.
2. A polynomial $I(Y_1, Y_2) \in C[Y_1, Y_2]$ is called a *semi-invariant* with respect to a differential operator L if the logarithmic derivative h'/h of its evaluation $h(x) = I\big(y_1(x), y_2(x)\big)$ on any C-basis (y_1, y_2) of solutions is rational, i.e., in $C(x)$.

The invariant polynomials (in short invariants) of degree m of a differential equation $Ly = 0$ are elements of the mth symmetric power $\mathrm{Sym}^m\big(\mathrm{Sol}(L)\big)$. Their values are elements of the space $\mathrm{Sol}(\mathrm{Sym}^m(L))$. An isomorphism between these two spaces preserving the Galois representations allows to identify an invariant to its value. As a consequence, determining the invariants or the semi-invariants of degree m of L is equivalent to determining the rational solutions of the mth

symmetric power $\mathrm{Sym}^m(L)$ of L. On the other hand, we know the full set of possible m since we know the list of invariants and semi-invariants of the finite groups $\mathbf{D}_n, \mathbf{A}_4, \mathbf{S}_4, \mathbf{A}_5$.

This provides us with a perfectly effective procedure to determine the invariants or semi-invariants of L and consequently the type of its differential Galois group.

Now, suppose L has a differential Galois group G with semi-invariant $\mathcal{S}$ of degree m and value $\sigma(x)$. And suppose the value of $\mathcal{S}$ with respect to the standard operator St_G equals σ_0 (modulo C^*). Then, the value of $\mathcal{S}$ with respect to both differential operators

$$S_G = \mathrm{St}_{PG} \otimes \left(\partial_z + \frac{\sigma_0'}{m\sigma_0}\right) \qquad \text{and} \qquad L = \tilde{L} \otimes \left(\partial_x + \frac{\sigma'}{m\sigma}\right)$$

is equal to 1 and the following property holds.

Proposition 1. *L is a proper pullback of S_G under $z = f(x)$.*

A direct examination in each case will provide the pullback function f.

4. Pullback Formulas

Primitive case: $PG \in \{\mathbf{A}_4, \mathbf{S}_4, \mathbf{A}_5\}$. The standard equation in reference is $\mathrm{St}_G\, y = 0$ where the differences of exponents are $\lambda = 1/3$ at $x = 0$, $\mu = 1/2$ at $x = 1$, and $\nu = 1/3$ for $\mathbf{A}_4$, $1/4$ for $\mathbf{S}_4$, and $1/5$ for $\mathbf{A}_5$ at infinity.

The differential Galois group of this equation has a semi-invariant S of degree $m = 4$ in the case of $\mathbf{A}_4$, $m = 6$ in the case of $\mathbf{S}_4$ and $m = 12$ in the case of $\mathbf{A}_5$ with value $s(x) = x^{-m/3}(x-1)^{-m/4}$. The new standard equation

$$S_G = \mathrm{St}_G \otimes \left(\partial + \frac{1}{3z} + \frac{1}{4(z-1)}\right)$$

reads

$$S_G = \partial^2 + \frac{7z-4}{6z(z-1)}\partial - \frac{1}{144}\frac{(6\nu-1)(6\nu+1)}{z(z-1)}.$$

It has exponents $(0, 1/3)$ at 0, $(0, 1/2)$ at 1, and $\left(\frac{6\nu+1}{12}, \frac{-6\nu+1}{12}\right)$ at infinity where ν has the previous value in each case. The semi-invariant $\mathcal{S}$ of degree m now has value 1. The coefficients of the pullback equation $\partial^2 y + a_1 \partial y + a_0 y = 0$ satisfy

$$a_1 = \frac{f''}{f'} + f' \frac{7f-4}{6f(f-1)} \qquad \text{and} \qquad a_0 = -\frac{(6\nu-1)(6\nu+1)f'^2}{144 f(f-1)}.$$

Algorithm. Input: $\tilde{L}$ with finite primitive group.

1. For $m \in \{4, 6, 12\}$ check for a semi-invariant of degree m and call v its logarithmic derivative.
2. If successful, let $L = \tilde{L} \otimes (\partial + \frac{1}{m} v)$ be a proper pullback of S_G with invariant value 1. Denote $L = \partial^2 + a_1\partial + a_0$.
3. Let $s = \frac{(6\nu-1)(6\nu+1)}{144}$, where $\nu \in \{1/3, 1/4, 1/5\}$ is known.

Output: the pullback function $f = \left(1 + \frac{s}{a_0}(6a_1 + 3\frac{a_0'}{a_0})^2\right)^{-1}$ and
for St_{A_4}, the basis of solutions $H_1 = {}_2F_1\left(\frac{-1}{12}, \frac{1}{4}, \frac{2}{3}; x\right)$ and $H_2 = \sqrt[3]{x}\ {}_2F_1\left(\frac{1}{4}, \frac{7}{12}, \frac{4}{3}; x\right)$;
for St_{S_4}, the basis of solutions $H_1 = {}_2F_1\left(\frac{-1}{24}, \frac{5}{24}], \frac{2}{3}; x\right)$ and $H_2 = \sqrt[3]{x}\ {}_2F_1\left(\frac{7}{24}, \frac{13}{24}, \frac{4}{3}; x\right)$;
for St_{A_5}, the basis of solutions $H_1 = {}_2F_1\left(\frac{11}{60}, -\frac{1}{60}, \frac{2}{3}; x\right)$ and $H_2 = \sqrt[3]{x}\ {}_2F_1\left(\frac{31}{60}, \frac{19}{60}, \frac{4}{3}; x\right)$.

Here ${}_2F_1$ denotes the hypergeometric function. In the case of St_{A_4} the solutions can also be given in terms of radicals or roots of an algebraic equation of degree 24.

Dihedral case: $PG = \mathbf{D}_n$ for $n \in \mathbb{N}$. The procedure is similar; however, one has to determine here the value of n.

For the sake of more symmetry, the standard equation in reference is chosen here with exponent differences $1/2$ at $+1$ and -1 and $1/n$ at infinity. It has a semi-invariant $\mathcal{S}_2 = Y_1 Y_2$ of degree 2 and two semi-invariants $\mathcal{S}_{n,a} = Y_1^n + Y_2^n$ and $\mathcal{S}_{n,b} = Y_1^n - Y_2^n$ of degree n. The new standard equation

$$S_{D_n} = \partial^2 - \frac{z}{z^2-1}\partial - \frac{1}{4n^2}\frac{1}{z^2-1}$$

has exponents $(0, 1/2)$ at $+1$ and -1, and $(-1/2n, 1/2n)$ at infinity; it has a semi-invariant of degree 2 and value 1. The operator $L = \partial^2 + a_1\partial + a_0$ is a pullback of S_{D_n} if

$$a_0 = -\frac{1}{4n^2}\frac{f'^2}{f^2-1} \qquad \text{and} \qquad a_1 = -\frac{1}{2}\frac{a_0'}{a_0}.$$

The equation $Ly = 0$ admits the solutions $\exp\int \pm\sqrt{-a_0} = \exp\int \frac{1}{2n}\frac{f'}{\sqrt{f^2-1}}\,dx$. The number n can thus be determined with the algorithm of integration on algebraic curves [3, 5, 6]; in fact, the authors give refinements of this part of the algorithm to compute a multiple of n.

Algorithm. Input: $\tilde{L} = \partial^2 + A_1(x)\partial + A_2(x)$ with finite differential Galois group.

1. Check for a semi-invariant of degree 2 and call v its logarithmic derivative.
2. If successful, let $L = \tilde{L} \otimes \left(\partial + \frac{1}{m}v\right)$ be a proper pullback of S_{D_n} with invariant value 1. Denote $L = \partial^2 + a_1\partial + a_0$.
3. Determine a candidate for a multiple of n.
4. For an adequate n, the equation $L_n y \equiv \partial^2 y + a_1\partial y + n^2 a_0 y = 0$ has solutions f and $\sqrt{f^2-1}$, hence f can be determined.
5. Let c be such that $c^2 = \frac{4n^2 a_0}{f'^2 + 4n^2 f^2 a_0}$.

Output: the pullback function $\pm cf$ and the solutions $\left(cf \pm \sqrt{c^2 f^2 - 1}\right)^{1/n}$.

The procedure appears to be more efficient than the Kovacic algorithm. In addition, it provides the pullback and the solutions in simple form.

Bibliography

[1] Baldassarri (F.) and Dwork (B.). – On second order linear differential equations with algebraic solutions. *American Journal of Mathematics*, vol. 101, n° 1, 1979, pp. 42–76.

[2] Beukers (Frits) and van der Waall (Alexa). – Lamé equations with algebraic solutions. – Available online at `http://www.math.uu.nl/people/beukers/`, 2002. 19 pages. Submitted to *Journal of Differential Equations*.

[3] Bronstein (Manuel). – Integration of elementary functions. *Journal of Symbolic Computation*, vol. 9, n° 2, 1990, pp. 117–173.

[4] Kovacic (Jerald J.). – An algorithm for solving second order linear homogeneous differential equations. *Journal of Symbolic Computation*, vol. 2, n° 1, 1986, pp. 3–43.

[5] Risch (Robert H.). – The solution of the problem of integration in finite terms. *Bulletin of the American Mathematical Society*, vol. 76, 1970, pp. 605–608.

[6] Trager (Barry M.). – *On the integration of algebraic functions*. – PhD thesis, MIT, 1984.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 47–48.

Available online at the URL
http://algo.inria.fr/seminars/.

The Structure of Multivariate Hypergeometric Terms

Marko Petkovšek
University of Ljubljana (Slovenia)

December 3, 2001

Summary by Bruno Salvy

Abstract

The structure of multivariate hypergeometric terms is studied. This leads to a proof of (a special case of) a conjecture formulated by Wilf and Zeilberger in 1992.

A function $u(n_1, \ldots, n_d)$ with values in a field $\mathbb{K}$ is called a *hypergeometric term* if there exist rational functions $R_i \in \mathbb{K}(n_1, \ldots, n_d)$, $i = 1, \ldots, d$, such that u is solution of a system of d first-order recurrences $S_i \cdot u = R_i(n_1, \ldots, n_d)u$, $i = 1, \ldots, d$, where S_i denotes the shift operator with respect to n_i (e.g., $S_1 \cdot u(n_1, \ldots, n_d) = u(n_1 + 1, n_2, \ldots, n_d)$).

In the univariate case, the numerator and denominator of R_1 factor into linear factors over the algebraic closure $\overline{\mathbb{K}}$ of $\mathbb{K}$. This factorization induces an explicit form for univariate hypergeometric terms as $C\rho^n \prod_{i=1}^{I} (n + \alpha_i)^{k_i}$, where C is a constant, $\rho \in \mathbb{K}$, $\alpha_i \in \overline{\mathbb{K}}$, $k_i \in \{1, -1\}$, and I is the sum of the degrees of the numerator and denominator of R_1. These terms thus express the Taylor coefficients of generalized hypergeometric series, whence their name.

In the multivariate case, no such simple factorization exists, but the rational functions are related through the identities $R_j(S_j R_i) = R_i(S_i R_j)$, $1 \le i, j \le d$. A non-obvious consequence of these relations is the following theorem from an entirely elementary Appendix of [5] (see also [2]). The bivariate case was proved by Ore in [4].

Theorem 1 (Ore–Sato)**.** *Hypergeometric terms can be written*

$$R(n_1, \ldots, n_d) \prod_{i=1}^{d} \rho_i^{n_i} \cdot \prod_{i=1}^{p} \prod_{k=0}^{e_i(n_1,\ldots,n_d)-1} \psi_i\big(e_i(n_1, \ldots, n_d) - k\big), \tag{1}$$

where R is a rational function, and for $i = 1, \ldots, d$, $\rho_i \in \mathbb{K}$, the e_i's are linear forms with integer coefficients and the ψ_i are univariate rational functions.

Definition 1. An expression of the form (1) where $R = 1$ is called a *proper hypergeometric term.*

Proper hypergeometric terms have the property of forming *holonomic* sequences. These are defined as follows.

Definition 2. A (multivariate) sequence is *holonomic* when the set of partial derivatives of its generating series spans a finite-dimensional vector space over the rational functions.

These series are sometimes called *D-finite*. An elementary proof of Kashiwara's equivalence between D-finiteness and holonomy in the sense of D-module theory is derived in [6, Appendix]. A characterization of holonomic sequences is provided by the following [3].

Theorem 2 (Lipshitz)**.** *A sequence $u_{n_1,\ldots,n_d}$ is holonomic if and only if there exists $s \in \mathbb{N}$ such that*

1. *for each* $i = 1, \ldots, d$, u *satisfies a linear recurrence of the form*

$$\sum_{\mathbf{h} \in \{0,\ldots,s\}^d} p_{\mathbf{h},i}(n_i) u_{\mathbf{n}-\mathbf{h}} = 0, \qquad n_i \geq s, i = 1, \ldots, d, \tag{2}$$

where bold letters indicate multi-indices and the coefficients are univariate polynomials;

2. *if* $d \geq 2$, *each of the specialized sequences* $u_{n_1,\ldots,n_{i-1},k,n_{i+1},\ldots,n_d}$ *is holonomic, for* $i = 1, \ldots, d$, $k = 0, \ldots, s-1$.

The importance of holonomy in computer algebra comes from its use by Zeilberger [8] for an algorithmic proof of many identities. In this context, holonomy provides with a sufficient condition for several definite summation or integration algorithms to terminate. An algorithm specifically designed for the definite summation or integration of *hypergeometric* terms was given by Wilf and Zeilberger [7]. There, they give a conjecture which has the following as a special case.

Theorem 3. *Hypergeometric terms form holonomic sequences if and only if they are proper.*

This result is due to Abramov and Petkovšek [2]. The sketch of the proof is as follows.

First, it was shown by Lipshitz [3] that D-finite series are closed under Hadamard (i.e., termwise) product. In other words, holonomic sequences are closed under product. Lipshitz's proof relies on combinatorial considerations on the dimensions of the vector spaces that are involved.

Second, proper hypergeometric terms are holonomic. In view of the closure property above, it is sufficient to prove this for factorials of linear forms with integer coefficients and their reciprocals. In these cases, a linear recurrence with *constant* coefficients is found by shifting the argument along a vector with non-zero integer coordinates living in the kernel of the linear form. Then Theorem 2 can be applied. (One uses the same recurrence for each i.)

The holonomy of a hypergeometric term u in the form given by the Ore–Sato theorem is equivalent to that of the leading rational function R: u can be multiplied by the inverse of its proper part, itself proper and therefore holonomic. The problem is thus reduced to the study of which *rational* sequences are holonomic. The conclusion follows from considering the *univariate* constraints (2).

It should be mentioned that holonomy is only a sufficient condition for Zeilberger's algorithm to terminate. In the bivariate case, a necessary and sufficient condition was given recently in [1].

Bibliography

[1] Abramov (S. A.). – Applicability of Zeilberger's algorithm to hypergeometric terms. In Mora (Teo) (editor), *ISSAC 2002 (July 7-10, 2002. Université de Lille, Lille, France)*. pp. 1–7. – ACM Press, New York, 2002. Conference proceedings.

[2] Abramov (S. A.) and Petkovšek (M.). – On the structure of multivariate hypergeometric terms. *Advances in Applied Mathematics*, vol. 29, n° 3, 2002, pp. 386–411.

[3] Lipshitz (L.). – *D*-finite power series. *Journal of Algebra*, vol. 122, n° 2, 1989, pp. 353–373.

[4] Ore (Oystein). – Sur la forme des fonctions hypergéométriques de plusieurs variables. *Journal de Mathématiques Pures et Appliquées*, vol. 9, n° 4, 1930, pp. 311–326.

[5] Sato (Mikio). – Theory of prehomogeneous vector spaces (algebraic part)—the English translation of Sato's lecture from Shintani's note. *Nagoya Mathematical Journal*, vol. 120, 1990, pp. 1–34. – Notes by Takuro Shintani, Translated from the Japanese by Masakazu Muro.

[6] Takayama (Nobuki). – An approach to the zero recognition problem by Buchberger algorithm. *Journal of Symbolic Computation*, vol. 14, n° 2-3, 1992, pp. 265–282.

[7] Wilf (Herbert S.) and Zeilberger (Doron). – An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Inventiones Mathematicae*, vol. 108, n° 3, 1992, pp. 575–633.

[8] Zeilberger (Doron). – A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, vol. 32, n° 3, 1990, pp. 321–368.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 49–54.

Available online at the URL
http://algo.inria.fr/seminars/.

Numerical Elimination, Newton Method and Multiple Roots

Jean-Claude Yakoubsohn
MIP, Université Paul Sabatier, Toulouse (France)

November 19, 2001

Summary by Bruno Salvy

Abstract

Newton's iteration has quadratic convergence for *simple* roots. We present a Newton-based iteration scheme with quadratic convergence for *multiple* roots of systems of analytic functions. This is a report on work in progress.

1. Newton Iteration, Approximate Roots and γ-Theorems

1.1. **Newton Iteration.** Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be an analytic function. Newton's method for solving $f = 0$ consists in approximating f by its linearization at a given point z, whence the equation

$$f(z) + f'(z)(y - z) = 0 \tag{1}$$

from where solving for y yields the following iteration

$$z_{k+1} = N_f(z_k) := z_k - f'(z_k)^{-1} f(z_k). \tag{2}$$

For this method to converge to a root ζ, it is necessary that $f'(\zeta)$ be invertible. The exact domain from where the iteration converges to a solution can have a very complicated fractal structure. However, convergence is usually very fast provided the initial point z_0 be chosen sufficiently close to ζ. This is made more precise in the following [1, Ch. 8].

Theorem 1 (Smale)**.** *Let f be analytic and*

$$\gamma(f, z) := \sup_{k>1} \left(\frac{\left\| f'(z)^{-1} f^{(k)}(z) \right\|}{k!} \right)^{\frac{1}{k-1}}.$$

If $f(\zeta) = 0$ and $f'(\zeta)^{-1}$ exists then for any z such that $\|z - \zeta\| \leq (3 - \sqrt{7})/\big(2\gamma(f, \zeta)\big)$, the sequence defined by $z_0 = z$ and (2) is well defined and satisfies

$$\|z_k - \zeta\| \leq \frac{\|z - \zeta\|}{2^{2^k - 1}}, \qquad k \geq 0. \tag{3}$$

Thus, there is a ball around the root such that starting with any point inside this ball, each step of Newton's iteration decreases the distance to the root quadratically.

An important property of γ in Theorem 1 is that it is actually invariant under unitary changes of coordinates, thus it is related to geometry rather than computation.

Proof. This is the only proof we give in detail. It gives a good idea of the principle of most of the other proofs in this summary.

All the necessary quantities are first expressed in terms of the Taylor coefficients of f at ζ by means of Taylor expansion of f and f' inside (2):

$$N_f(z)-\zeta = f'(z)^{-1}\big(f'(z)(z-\zeta)-f(z)\big) = f'(z)^{-1}f'(\zeta)\sum_{k\geq 1}(k-1)f'(\zeta)^{-1}\frac{f^{(k)}(\zeta)}{k!}(z-\zeta)^k.$$

Set $u=\gamma(f,\zeta)\|z-\zeta\|$, then provided $u<1$, the norm of the sum above is bounded by $\|z-\zeta\|\times\big((1-u)^{-2}-(1-u)^{-1}\big)$. The norm of the remaining product is bounded by considering first its inverse, again via a Taylor expansion: $f'(\zeta)^{-1}f'(z)=1+B$, with $\|B\|\leq(1-u)^{-2}-1$, provided again that $u<1$. Now if $\|B\|$ itself is smaller than 1 (i.e., $u<\big(5-\sqrt{17}\big)/4\approx .2192$), the norm of the inverse can be bounded by the geometric series and this leads to $\big\|N_f(z)-\zeta\big\|\leq\big\|z-\zeta\big\|u/\psi(u)$ where $\psi(u)=1-4u+2u^2$. Thus the Newton iteration converges as soon as $u<\psi(u)$, i.e., $u<\big(5-\sqrt{17}\big)/4$ and the conclusion of the theorem follows from considering $u<\psi(u)/2$. □

A simple corollary of the end of this proof is that two distinct roots ζ and ζ' are at distance at least $\big(5-\sqrt{17}\big)/\big(4\gamma(f,\zeta)\big)$. This last result has been strengthened by Dedieu [2], who obtained $1/2$ instead of .2192 (see below for a proof). Theorem 1 is the basis for the following.

Definition 1. An *approximate root* of $f(z)=0$ is a point such that the sequence defined by $z_0=z$ and (2) is well defined and satisfies (3).

Although the results in Theorem 1 are stated in terms of quantities evaluated at the root, similar manipulations of Taylor expansions lead to variants in terms of quantities evaluated at the current iterate, on which tests can be based [1, Section 8.2].

1.2. **Homotopy.** A remaining problem is to locate approximate solutions.

In the univariate case, this can be achieved by starting from sufficiently many initial points [5].

Another method, which works also in the multivariate case, starts from the map $f_t:[0,1]\times\mathbb{C}^n\to\mathbb{C}^n$ defined by $f_t(x)=f(x)-tf(x_0)$. For $t=1$ a root x_0 is known and then one "follows" the curve from $(1,x_0)$ to $(0,x)$ where x is a root of f. The idea is to partition $[0,1]$ into $t_0=1>t_1>\cdots>t_k=0$ and then apply only one Newton iteration at each t_i: $z_0=x_0$, $z_{i+1}=N_{f_{t_{i+1}}}(z_i)$, for $i=0,\ldots,k-1$. The complexity of this method is related to how small k can be made. This in turn is eventually related to the so-called *fiber distance* of the system along this curve to the discriminant variety (see [1]).

2. Multiple Roots and Clusters in the Univariate Case

Numerically, there is no difference between multiple roots and clusters of roots. Also, the scale of the problem is the only difference between clusters of roots and well-separated roots. Thus, it is important to find an algorithm converging efficiently to clusters of roots. After a cluster has been isolated, either the computation stops and outputs a ball containing the cluster and the number of its elements, or the scaling is refined and the method is applied recursively to converge to each of the roots or subclusters. It turns out that this is possible by exploiting the fact that Newton's iteration does *not* converge *quadratically* to multiple roots. We now review the properties of Newton's method leading to an algorithm [7].

2.1. **Multiplicity and Speed of Convergence.** If ζ is a root of multiplicity m of f, then in the neighbourhood of ζ, one has

$$f(z)\sim f^{(m)}(\zeta)\frac{(z-\zeta)^m}{m!},\qquad f'(z)\sim f^{(m)}(\zeta)\frac{(z-\zeta)^{m-1}}{(m-1)!}.$$

From there it follows that if $m > 1$, the speed of convergence gives information on the value of m:

$$N_f(z) - \zeta = (z - \zeta) - \frac{f(z)}{f'(z)} \sim \left(1 - \frac{1}{m}\right)(z - \zeta). \tag{4}$$

Another consequence of this estimate is that in the neighbourhood of ζ, *the Newton sequence is close to a straight line.*

2.2. **Algorithm.** The idea is to use these properties in three steps: *(i)* compute three iterates x_0, x_1, x_2 and use them to estimate m in view of $(x_2 - x_1)/(x_1 - x_0) \sim 1 - 1/m$; *(ii)* use this value of m to "jump" directly to a better approximation of ζ using (4); *(iii)* control whether this approximation lies in the center of a cluster of m roots using *a posteriori* bounds; if so return the approximation and a radius of a ball containing the cluster, otherwise compute a new iterate x_3 and start again.

2.3. **A Posteriori Bounds.** Recall that Rouché's theorem states that if $\big|f(x) - g(x)\big| < \big|g(x)\big|$ on a circle $|x - z| = r$, then f and g have the same number of zeroes (counted with multiplicity) in the disk with centre z and radius r. Bounds are obtained by considering g defined by $g(x) = \sum_{k \geq m} f^{(k)}(z)(z - x)^k/k!$. If $f^{(m)}(z) \neq 0$, this polynomial has a zero of multiplicity m at z and does not have any other root in a disc of radius at least $R = 1/2\gamma_m(f, z)$, where

$$\gamma_m(f, z) := \sup_{k > m} \left| \frac{m! f^{(k)}(z)}{k! f^{(m)}(z)} \right|^{\frac{1}{k-m}}.$$

This is proved by evaluating f at another zero w using the Taylor expansion of f at z: from $f(w) = 0$ we get

$$\frac{f^{(m)}(z)}{m!}(w - z)^m = \sum_{k > m} \frac{f^{(k)}(z)}{k!}(w - z)^k.$$

Then using the triangular inequality and the definition of γ_m yields the result.

The next step is to find a radius $r \leq R$ such that $|f - g| < |g|$ on the corresponding circle. Rewriting this using the triangular inequality leads to the following.

Theorem 2 (Yakoubsohn)**.** *Let $z \in \mathbb{C}$ and $0 < r < 1/2\gamma_m(f, z)$ be such that*

$$\frac{|f^{(m)}(z)|}{m!} r^m - \sum_{k \neq m} \frac{|f^{(k)}(z)|}{k!} r^k > 0,$$

then f contains m roots counted with multiplicity in the disk of centre z and radius r.

2.4. **Homotopy.** In the same way as in the case of simple roots, it is possible to combine these ideas with a homotopy method to extend the domain of application of this algorithm. A discussion of a way to find an appropriate subdivision $1 = t_0 > \cdots > t_k = 0$ can be found in [7], together with a complexity analysis.

3. Moore–Penrose Inverse

When the number of variables is different from the number of equations, or more generally when $f'(z)$ in (1) is not an isomorphism, Newton's iteration (2) does not apply. It turns out that a simple modification of this iteration gives a process with interesting fixed points both in the underdetermined and overdetermined case. In the underdetermined case, the fixed points are the points of the variety defined by f, while in the overdetermined case, the fixed points are solutions

of the least-square problem associated with the equations. This is achieved by means of the Moore–Penrose inverse. Let A be a linear operator between two Euclidean or Hermitian spaces E and F. The *Moore–Penrose* inverse $A^\dagger$ of A is defined as $A^\dagger = \iota B^{-1}\pi_{\operatorname{Im} a}$, where $\pi_{\operatorname{Im} a}$ is the orthogonal projection on the image $\operatorname{Im} A$ of A, B is the restriction B of A to the orthogonal of its kernel, and ι is the injection of this orthogonal to A. It satisfies $A^\dagger A = \pi_{(\operatorname{Ker} A)^\perp}$, $AA^\dagger = \pi_{\operatorname{Im} A}$.

With this inverse, the Newton iteration can be generalized to the Newton–Gauss iteration:

$$z_{k+1} = N_f(z_k) := z_k - f'(z_k)^\dagger f(z_k). \tag{5}$$

The convergence properties of this iteration are expressed in terms of

$$\gamma^\dagger(f,z) := \sup_{k>1} \left(\frac{\left\| f'(z)^\dagger f^{(k)}(z) \right\|}{k!} \right)^{\frac{1}{k-1}}.$$

The result is parallel to Theorem 1, d denotes the distance.

Theorem 3 (Shub–Smale, Dedieu–Shub)**.** *Let V be the zero-set of f and $\zeta \in V$. If $f'(\zeta)$ is surjective, then for any z such that $|z-\zeta| \le c/\gamma^\dagger(f,\zeta)$, the sequence defined by $z_0 = z$ and (5) is well defined and satisfies $d(z_k,V) \le d(z,V)/2^{2^k-1}$, for $k \ge 0$. Moreover, the sequence converges to a point $Z \in V$ such that $d(z,V) \le d(z,Z) \le 2d(z,V)$.*

In this theorem, c is a universal constant that does not depend on f.

Fixed points of the Newton–Gauss iteration in the overdetermined case are not necessarily attractive. A similar constant γ can be defined, but this time the convergence (to a solution of the least-square problem $F'(x) = 0$ where $F(x) = \left\| f(x) \right\|^2$) is not quadratic anymore, see [3].

4. **Fast Deflation**

The results above do not apply in cases with multiple roots, or clusters of roots. In a slightly different context, that of finding series solutions of polynomial systems, G. Lecerf has devised in [6] an algorithm based on a Newton iteration with low complexity and quadratic convergence even in presence of multiplicities. We now give an outline of this algorithm and then sketch how this algorithm is adapted to the Archimedean world. To simplify notations, all series expansions are performed at the origin.

4.1. **Deflated System.** Let f be a system of n polynomial equations in n variables $x_1, \ldots, x_n$ having a finite number of solutions. To simplify the description, we assume some genericity conditions to be satisfied. The algorithm proceeds recursively by constructing an auxiliary block-triangular non-linear system (the deflated system). In each block, a subset of the variables $x_i, x_{i+1}, \ldots, x_{i+r_i-1}$ can be solved for in terms of the remaining ones $x_{i+r_i}, \ldots, x_n$ by means of a Newton iteration with quadratic convergence. At the end, it is therefore sufficient to solve these systems starting from the last one to get a solution of the original polynomial system.

The recursive step of this construction starts by computing the valuation of x_i (the first indeterminate which does not belong to one of the blocks constructed so far) in all the given equations. Let m_i be the smallest of these valuations. When computing series at a multiple root, $m_i > 1$ for $i > 1$ and the product of these m_i's is bounded by the multiplicity.

The equations are then differentiated j times with respect to x_i, for $j = 1, \ldots, m_i - 1$. A rectangular system Φ_i is formed with the original equations and these new equations. A subsystem Ω_i is then extracted, whose Jacobian has maximal possible rank r_i. By construction, Ω_i makes it possible to compute $x_i, \ldots, x_{i+r_i-1}$ by a Newton iteration with quadratic convergence, given values for the next variables. The next step is then started with Φ_i and the remaining variables as input.

4.2. **Series Expansions and Complexity.** Differentiation is needed at several stages during the algorithm: first when computing the systems Φ_i and then when performing Newton iterations on the systems Ω_i. However, the equations to be differentiated involve quantities that are known only implicitly as solutions of previous systems.

A way of computing the necessary derivatives is to compute multivariate series expansions at each stage. This means that each of the systems Ω_i is solved by a Newton iteration that converges both to the values and to the series expansions out of which extracting coefficients yields the values of the desired derivatives.

The nontrivial proof that this method leads to a correct algorithm with good complexity properties can be found in [6].

4.3. **Towards a Numerical Fast Deflation.** The idea of this work [4] is to follow the same steps as in Lecerf's algorithm, using numerical tools at intermediate steps.

The computation of valuations in series is replaced by the computation of multiplicity by the algorithm of Section 2.2. Because some of the variables are expressed as solutions of previous Ω_i's, the equations are not polynomial anymore. The idea is to work as if they were polynomial, using the Weierstrass preparation theorem to compute the required bounds. The computation of the rank is performed, for instance, by an LU-decomposition with a proper threshold to erase smaller diagonal entries. The computation of series is performed as in the symbolic case, with floating point coefficients that are themselves found by the Newton iteration.

What remains then is a rigorous analysis of how the convergence of this method and its quadratic behaviour can be related to the geometry of the problem via analogues of the γ-functions used in the theorems presented here. This will be treated in [4].

4.4. **Example.** We treat in detail a system borrowed from [6]: $f = g = h = 0$, with

$$f = 2x+2x^2+2y+2y^2+z^2-1, \quad g = (x+y-z-1)^3-x^3, \quad h = (2x^3+5y^2+10z+5z^2+5)^3-1000x^5.$$

This system has a solution of multiplicity 18 at $(0, 0, -1)$.

The computation begins with initial point $(.2, .1, -.98)$. Setting $y = .1$, $z = -.98$ in the system and using the algorithm of Section 2.2 reveals that .2 is close to a *simple* root for x, in the first equation only. Thus no differentiation is needed at this stage and the first block of the final system is $f = 0$. This defines a function $X(y, z)$ that can be computed by Newton iteration, as well as its Taylor expansion. The next step considers the remaining equations at $z = -.98$, x being replaced by $X(y, -.98)$. Applying the same technique shows that .1 is close to a root of multiplicity 3 for y in g and 5 in h. Thus the second block of the final system is formed by $\partial^2 g/\partial y^2\big(X(y, z), y, z\big)$ which defines a function $Y(z)$ that can be computed by Newton iteration, together with its Taylor expansion. Finally, $H(z) = \partial^2 h/\partial y^2\big(X(Y(z), z), Y(z), z\big)$ is found to have a root with multiplicity 4 for z close to $-.98$. This yields the last block of the system: $H^{(3)}(z) = 0$. Note that the product of the multiplicities that have been found—$1 \times 3 \times 4 = 12$—is smaller than the actual multiplicity 18.

One iteration of the algorithm on the system consists, for each of the blocks, in performing several Newton iterations to compute sufficiently many terms of the series expansion. After each such Newton iteration, the previous coordinates are updated using the derivatives of their series.

Thus, we start with f which we evaluate at $x = .2 + d_x$, $y = .1 + d_y$, $z = -.98 + d_z$. This yields

$$S(d_x, d_y, d_z) = (.6604 + 2.4d_y - 1.96d_z + 2d_y^2 + d_z^2) + 2.8d_x.$$

From there, the derivative with respect to x is obtained as the coefficient of d_x and we get as a result of the Newton iteration

$$X = .2 - S(0, d_y, d_z)/(\partial S/\partial d_x) = -.0358571 + .7d_z - .3571428d_z^2 - .8571429d_y - .7142857d_y^2.$$

Iterating again twice with $f(X + d_x, .1 + d_y, -.98 + d_z)$ yields

$$X = -.10022540 + 1.2248159d_z - 2.4860889d_z^2 + \cdots + O(d_y^4 + d_z^5).$$

We now turn to the second block of the deflated system, $\partial^2 g/\partial y^2\big(X(y,z), y, z\big)$. To perform a Newton iteration on this block, g is first evaluated at $X, .1 + d_y, -.98 + d_z$ and then differentiated twice with respect to d_y. This yields

$$\begin{aligned} S(d_y, d_z) &= 1.5559281 - 31.251336d_z + 269.81555d_z^2 - 1962.5419d_z^3 + 13304.819d_z^4 \\ &\quad + (42.453768 - 708.11523d_z + 7291.0178d_z^2 - 63814.277d_z^3 + 506494.98d_z^4)d_y + O(d_y^2 + d_z^5). \end{aligned}$$

From there a Newton iteration yields

$$Y = .063350059 + .12481705d_z + 2.0206612d_z^2 + 3.4053518d_z^3 + 21.246801d_z^4 + O(d_z^5).$$

This value is then used to *update* the estimate X obtained at the previous stage, via

$$X_{\text{new}} := X_{\text{old}} + \partial X/\partial d_y (Y_{\text{new}} - Y_{\text{old}}).$$

The new estimate, $X = -.045258749 + .87056807d_z - 4.1178532d_z^2 + \cdots + O(d_y^4 + d_z^5)$, is then used together with Y to perform another Newton iteration on this block, and another update of X.

The treatment of the last block is similar. The only novelty is that the second derivative of h with respect to y has to be computed. This is achieved by evaluating h at $X(d_y, d_z), Y(d_z) + d_y, -.98 + d_z$, and extracting the coefficient of d_y^2. Then, differentiating three times with respect to d_z yields

$$S = 1673.8759 + 59921.515d_z + O(d_z^2),$$

from which one Newton iteration gives $Z = -1.0079345$ and updating the previous coordinates

$$\begin{aligned} X &= -.0020677183 + .069477760d_z + .44632326d_z^2 + \cdots + O(d_y^4 + d_z^5), \\ Y &= .0023005230 + .85445586d_z - 1.2958872d_z^2 + 36.553123d_z^3 - 250.55237d_z^4 + O(d_z^5). \end{aligned}$$

In brief, this iteration of the algorithm has led from $(.2, .1, -.98)$ to $(-.0021, .0023, -1.008)$, from distance .2 to distance $8\,10^{-2}$ to the root. Similarly, the next iterations are respectively at distance of order $8\,10^{-4}$, $6\,10^{-6}$, $3\,10^{-9}$, $6\,10^{-18}$ and $2\,10^{-36}$ from the root, thus exhibiting a clearly quadratic behaviour after the first few iterations.

Bibliography

[1] Blum (Lenore), Cucker (Felipe), Shub (Michael), and Smale (Steve). – *Complexity and real computation*. – Springer-Verlag, New York, 1998, xvi+453p.

[2] Dedieu (Jean-Pierre). – Condition number analysis for sparse polynomial systems. In *Foundations of computational mathematics*. pp. 75–101. – Springer, Berlin, 1997. Proceedings of a conference held at Rio de Janeiro, 1997.

[3] Dedieu (Jean-Pierre). – Newton's method and some complexity aspects of the zero-finding problem. In DeVore (Ronald A.), Iserles (Arieh), and Süli (Endre) (editors), *Foundations of computational mathematics (Oxford, 1999)*, pp. 45–67. – Cambridge University Press, Cambridge, 2001. Proceedings of FoCM'99.

[4] Giusti (Marc), Lecerf (Grégoire), Salvy (Bruno), and Yakoubsohn (Jean-Claude). – Numerical Newton iteration with quadratic convergence to multiple solutions. – In preparation.

[5] Hubbard (John), Schleicher (Dierk), and Sutherland (Scott). – How to find all roots of complex polynomials by Newton's method. *Inventiones Mathematicae*, vol. 146, n° 1, 2001, pp. 1–33.

[6] Lecerf (Grégoire). – *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques*. – Thèse de doctorat, École polytechnique, September 2001.

[7] Yakoubsohn (Jean-Claude). – Finding a cluster of zeros of univariate polynomials. *Journal of Complexity*, vol. 16, n° 3, 2000, pp. 603–638. – Complexity theory, real machines, and homotopy (Oxford, 1999).

Part III

Analysis of Algorithms, Data Structures, and Network Protocols

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 57–62.

Available online at the URL
http://algo.inria.fr/seminars/.

Everything You Always Wanted to Know about Quicksort, but Were Afraid to Ask

Marianne Durand
Projet Algorithmes, Inria Rocquencourt (France)

November 5, 2001

Summary by Michel Nguyễn-Thế

Abstract

The algorithm Quicksort was invented by Hoare in 1960. Numerous improvements have been suggested since then, like optimization of the choice of the pivot or simultaneous use of several pivots or also hybrid methods. Different parameters like the cost of comparisons, the size or the height of the associated binary search tree have been studied for Quicksort and its variants. We present here the principal methods used to get the mean, the variance, and the nature or at least a few properties of the limit laws of these parameters.

1. Description of the Algorithm and of a Few Variants

1.1. **Quicksort.** The procedure Quicksort takes as arguments an array A of n elements and two integers First and Last representing indices of elements of the array. The algorithm runs as follows: if First $<$ Last then:

1. Choose a pivot in the array (e.g., A[First]).
2. Partition the elements in the subarray A[First] ... A[Last] so that the pivot value is in place (let PivotIndex be its position then).
3. Apply Quicksort to the first subarray A[First] ... A[PivotIndex $-$ 1].
4. Apply Quicksort to the second subarray A[PivotIndex $+$ 1] ... A[Last].

1.2. **Variants.** In step 1, the pivot is chosen in a fixed manner. It is possible to use a strategy to choose the pivot to improve the efficiency of the algorithm. By choosing the pivot randomly, we can wipe out the possible bias of the data we want to sort. The pivot is all the more efficient if it cuts the array in two arrays of similar size. With this aim in view, the Quicksort with median of $2t+1$ consists in picking out $2t+1$ elements randomly in the array to sort, where t is a fixed integer, and to choose as pivot the $(t+1)$th element among the picked elements. Martínez and Roura [7] even analysed the situation with a sample size depending on n, and obtained that the optimal sample size to minimize the average total cost of Quicksort, including comparisons and exchanges, is $s = a\sqrt{n} + o(\sqrt{n})$, for some constant a. Quicksort with 3–3 median consists in picking 3 samples of the array, each of 3 elements. We take the median element of each sample, so that we are left with three elements, of which we take again the median element, that we finally choose as pivot. This strategy can be furthered in choosing $m-1$ pivots or medians, among $m(t+1)-1$ elements, instead of one only, that leaves us with sorting recursively m subarrays instead of two only.

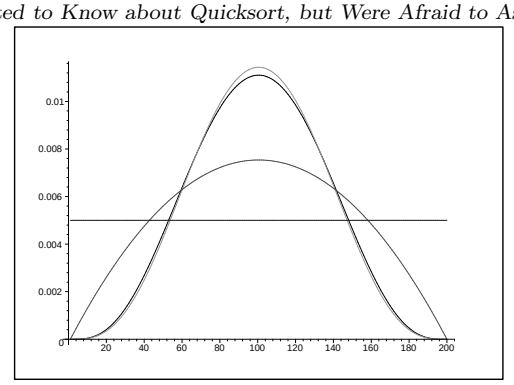


FIGURE 1. Probability of choice of pivot.

1.3. **Parameters.** The parameters of interest are: the cost in number of comparisons, that is the internal path length of the associated search tree; the size of the associated m-ary tree; the profile of the tree. Notice that the size of a binary tree is the number of internal nodes, and that the size of an m-ary tree for $m \geq 3$ is the number of both internal and external nodes. We can generally compute the first moments or cumulants of these parameters, especially the mean and the variance, by using generating functions. With the aid of various other tools, it is possible to show the existence of a limit law for the cost of Quicksort and to derive some properties of this law. For the other parameters, the knowledge of the moments sometimes turns out to be sufficient to establish a Gaussian limit.

2. Moments of Internal Path Length

2.1. **Expectation.** Recall that m is the arity of the tree. The cost expectation f_n of Quicksort and its variants is given by the recurrence relation

$$f_n = t_n + \sum_{k=1}^{n} \mathbf{P}(\text{PivotIndex} = k)\,(f_{k-1} + f_{n-k}), \tag{1}$$

which can be rewritten into the equation $\mathcal{L}\big(f(z)\big) = (1-z)^{-\beta}$, where $f(z) = \sum_n z^n$, and the operator $\mathcal{L}$ is of the form $\mathcal{L}(y) = a_m(1-z)^m y^{(m)} + \cdots + a_0 y$. There exists a polynomial $I(\alpha)$, called the index polynomial, such that $\mathcal{L}\big((1-z)^{-\alpha}\big) = I(\alpha)(1-z)^{-\alpha}$. The solutions of $\mathcal{L}(y) = 0$ are given by $(1-z)^{-\alpha}\log^k \frac{1}{1-z}$ with $I(\alpha) = 0$ and k smaller than the order of multiplicity of root α. Given the initial conditions and the particular solution $I^{(r)}(\beta)(1-z)^{-\beta}\big(\log\frac{1}{1-z}\big)^r$, where r is the order of β as root of I (r can be zero), it is then easy to get the right solution, that is for instance

$$f(z) = \sum_{\alpha} \frac{\lambda_\alpha}{(1-z)^\alpha} + \frac{10!}{2311776}\frac{1}{(1-z)^2}\log\frac{1}{1-z} - \frac{26}{3}\frac{1}{1-z} \tag{2}$$

for Quicksort with 3–3 median, and by singularity analysis to compute the following expectations

Method	Mean
Quicksort	$2n\log n$
Median of 3	$(12/7)n\log n$
Median of 3–3	$1.57n\log n$

2.2. **Cumulants.** Consider a random variable of probability generating function $g(z) = \sum_n g_n z^n$. Its cumulants are defined by

$$\kappa_p(n) = \frac{\partial^p}{\partial y^p} \ln g_n(e^y)\bigg|_{y=0}.$$

Notice that κ_1 and κ_2 respectively represent the mean and the variance of the considered distribution. Hennequin [5] showed that the cumulants of median of $2t+1$ Quicksort cost for s-ary trees are of the form

$$\kappa_p(n) = n^p K_{s,t}^p (L_{p,s,t} - (p-1)!\,\zeta(p)) + o(n^p),$$

where ζ is the zeta Riemann function and the constants $K_{s,t}$ and $L_{p,s,t}$ are rational numbers easily computed by induction.

3. Properties of the Limit Law for Internal Path Length (Binary Case)

Though the problem is still open whether there exists a close form expression of the limit law in terms of known functions, we know some properties of the limit law.

3.1. **Existence of a limit law.** Let X_n be the random variable counting the number of comparisons in an array of size n, and $Y_n = \frac{X_n - \mu_n}{n}$ the corresponding normalized random variable. Régnier showed the existence of a limit law for Y_n with almost sure convergence by using martingales.

3.2. **Method of contraction.** X_n follows the same distribution as $n-1+X_{Z_n-1}+X_{n-Z_n}$, where Z_n is uniformly drawn in the set $\{1,\cdots,n\}$: Z_n-1 and $n-Z_n$ represent the sizes of the left and right subarrays. In terms of Y_n, it rewrites into the recurrence relation

$$Y_n \overset{\mathcal{D}}{=} Y_{Z_n-1}\frac{Z_n-1}{n} + \overline{Y}_{n-Z_n}\frac{n-Z_n}{n} + C_n(Z_n),$$

for some computable $C_n(Z_n)$, and one can guess that the limit law of Quicksort cost is a fixed point of the equation

$$Y \overset{\mathcal{D}}{=} \overline{Y}\tau + \overline{\overline{Y}}(1-\tau) + C(\tau),$$

where $\overline{Y}$ and $\overline{\overline{Y}}$ are independent copies of Y, and $C(u) = 1+2u\ln u+2(1-u)\ln(1-u)$. Rösler [8, 9] established that it is true by using a method of contraction, working in a metric space of distribution, endowed with the Wasserstein metrics d_2 defined by $d_2(F,G) = \inf \|X-Y\|_2$.

Using the same method but with more precise majorizations, Fill and Janson found the following bounds on the speed of convergence: $d_2(Y_n,Y) < 2/\sqrt{n}$, and more generally $d_p(Y_n,Y) < c_p/\sqrt{n}$ for certain constants c_p. They also showed that $d_p(Y_n,Y) = O(\log n/n)$.

3.3. **Density of limit law.**

3.3.1. *Existence.* Tan and Hadjicostas [10] showed that the limit law Y of Quicksort cost admits a density, by considering the function $h_{y,z}(u) = uy + (1-u)z + C(u)$, which is clearly related to the fixed-point equation. By exchanging the axes, we get a curve with two branches r and l that are differentiable and hence admit a density.

Hence we can write

$$\mathbf{P}\big(h(U) \le t\big) = \int_{-\infty}^{t} \left(r'\,1_{[b,y+1]} - l'\,1_{[b,z+1]}\right) d\lambda = \int_{-\infty}^{t} g(y,z,t),$$

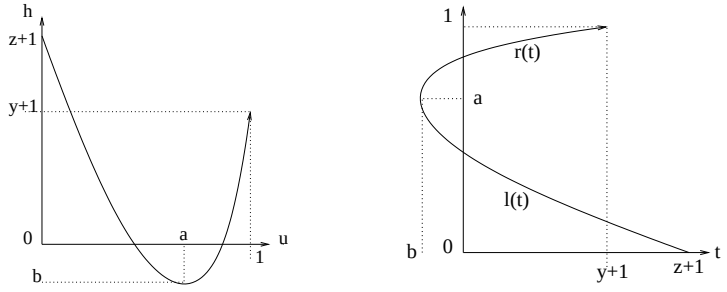


FIGURE 2. The function $h_{y,z}$ and its inverse.

(1_S is the characteristic function of the set S) and then, for all Borel set B,

$$\mu(B) = \mathbf{P}\big(UY + (1-U)\overline{Y} + C(U) \in B\big) = \int_{\mathbb{R}^2} \mathbf{P}\big(h_{Y,\overline{Y}}(U) \in N\big)\, d\mu \otimes d\nu$$
$$= \int_{\mathbb{R}^2} \int_B \left(g(y,z,s)\, d(\mu\otimes\mu)(y,z)\right) d\lambda(s) = \int_B \left(\int_{\mathbb{R}^2} g(y,z,s)\, d(\mu\otimes\mu)(y,z) \right) d\lambda(s),$$

which proves the result.

3.3.2. *Bounds on the density and its derivatives.* Fill and Janson showed that, for all integer p, there exists a constant b_p such that the characteristic function $\phi(t) = \mathbf{E}\, e^{itY}$ satisfies $|\phi(t)| \le c_p|t|^{-p}$ for all $t \in \mathbb{R}$. Using the equality

$$f^{(k)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (-it)^k e^{-itx} \phi(t)\, dt,$$

they deduce that the density f of Quicksort is C^∞ and the bounds, for all $k \in \mathbb{N}$ and $p \in \mathbb{R}^+$, $\big|f^{(k)}(x)\big| \le C_{p,k}|x|^{-p}$. In particular we have $\big|f(x)\big| \le 15.3$.

3.3.3. *Queues on limit distribution.* Knessl and Szpankowski [6] established that the left tail of the limiting distribution has a doubly exponential decay, while the right tail only has an exponential decay:

$$\begin{cases} \mathbf{P}\big(\mathcal{L}(Y_n) - \mathbf{E}\,\mathcal{L}(Y_n) \le nz\big) \sim \frac{2}{\pi} \frac{1}{\sqrt{2\log 2 - 1}} \exp\left(-\alpha \exp\left(\frac{\beta - z}{2 - \log^{-1} 2}\right)\right), \\ \mathbf{P}\big(\mathcal{L}(Y_n) - \mathbf{E}\,\mathcal{L}(Y_n) \ge ny\big) \sim a(y) \exp(-y b(y)), \end{cases}$$

where α and β are constants, and a and b are positive and polynomially bounded functions.

3.3.4. *Simulation of Quicksort Distribution.* Devroye, Fill, and Neininger devised a rejection algorithm that simulates Quicksort in a perfect way. They use a fully known function g majorizing f, and a sequence of error bounds based on the difference between the distribution function F_n of X_n and the limit distribution function F. Figure 3 shows the functions g, Quicksort density (bold curve), and successive error bounds. The algorithm stops when one goes outside an error bound, rejects if one is over the upper bound, and accepts if one is below the lesser bound.

4. m-ary Trees

The m-ary search trees are a generalization of binary search trees. We choose now up to $m-1$ medians among $m(t+1)-1$ elements, and put these medians in the same node.

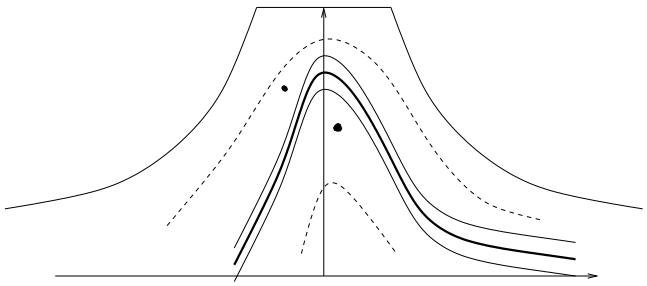


FIGURE 3. Quicksort simulation with rejection algorithm.

4.1. **Space requirement for m-ary trees.** As the number of keys occupying a node can be less than $m-1$ (it corresponds to a subarray with a size inferior to $m-1$), an issue at stake for $m>2$ is to know the number X_n of nodes (both internal and external) required to sort a given sequence of n keys.

4.1.1. *Moments.* It is possible to compute the moments of any order of the centered random variable [2]. The generating function $F(z,y)=\sum_n \mathbf{E}\left(y^{X_n}\right) z^n$ satisfies

$$D_z^{m-1}F(z,y)=(m-1)!\,F^m(z,y).$$

For the centered generating function defined by $G(z,y)=\sum_n \mathbf{E}\left(y^{X_n-\mu(n+1)}\right) z^n = y^{-\mu}F(zy^{-\mu},y)$, where μ satisfies $X_n\sim\mu n$, it translates into

$$D_z^{m-1}G(z,y)=(m-1)!\,yG^m(z,y).$$

The generating function of the kth factorial moment is $G_k(z)=D_y^kG(z,y)\big|_{y=1}$. It satisfies

$$\mathcal{L}[G_k]=k!\,(m-1)!\,(1-z)^{m-1}Q_k(z),$$

where the operator $\mathcal{L}$ is here defined by $\mathcal{L}[G]=(1-z)^{m-1}D_z^{m-1}G-m!\,G$, and Q_k is a linear combination of products of G_j's with $j<k$. The asymptotics of the variance depends on the position of the zeroes of the indicial polynom of $\mathcal{L}[G_2]$, and the limit behaviour varies with m.

4.1.2. *Gaussian limit law for $m\le 26$.* The variance is linear if $m\le 26$ because

$$G_2(z)\sim\frac{\sigma^2}{(1-z)^2}.$$

If $m\le 26$ then $\frac{X_n-\mu n}{\sigma\sqrt{n}}\to\mathcal{N}(0,1)$. Indeed, pumping moments provides an asymptotics for the G_k's

$$\begin{cases} G_{2k-1}(z)=o\left(|1-z|^{-k-1/2}\right),\\ G_{2k}(z)\sim(2k)!\,2^{-k}\sigma^{2k}(1-z)^{-k-1}, \end{cases}\qquad \text{which entails}\qquad \begin{cases} \mathbb{E}\left(\frac{X_n-\mu n}{\sigma\sqrt{n}}\right)^{2k-1}=o(1),\\ \mathbb{E}\left(\frac{X_n-\mu n}{\sigma\sqrt{n}}\right)^{2k}=\frac{(2k)!}{2^k k!}. \end{cases}$$

4.1.3. *Still an open case for $m>26$.* The variance is more than linear if $m>26$:

$$\mathbf{Var}(X_n)\sim a(n)n^{2\alpha-2}.$$

The limit law is conjectured not to be Gaussian any longer. If ever it was, then the normalization would be exotic, because for $m>26$, the limit distribution of the random variable $(X_n-\mu n)/n^{\alpha-1}$ does not exist.

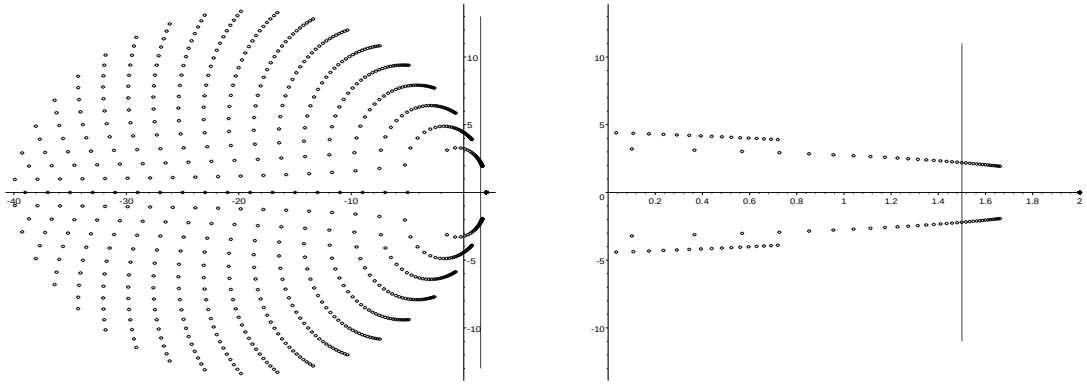


FIGURE 4. (Borrowed from [2].) Zeros of the indicial polynom $\Lambda(\vartheta)$ of $\mathcal{L}[G_1]$ for m from 5 to 40; zeros with positive real parts and the vertical line $\mathrm{Re}(\vartheta) = 3/2$ (which may be called the "phase-change line") are shown on the right.

4.2. **Profile of the tree.** Let $f_n(u)$ be the generating function of the height of leaves in m-ary trees of size n. The recurrence $f_n = um \sum_i \pi_i f_i$, where $\pi_i = \binom{n}{m-1}^{-1}\binom{n-i-1}{m-2}$ is the probability that an m-ary tree of size n has a first child of size i [3], translates into the differential equation $D_z^{m-1}F(z,u) = um!\,F(z,u)(1-z)^{1-m}$, where $F(z,u) = \sum f_n(u)z^n$. It solves to $F(z,u) \sim \lambda(u)(1-z)^{\alpha(u)}$ in the vicinity of $u = 1$. Hence $f_n(u) \sim \lambda(u)\Gamma\big(\alpha(u)\big)^{-1}\left(e^{\alpha(u)-1}\right)^{\log n}$, and according to the Quasi Powers theorem [4], the limit law of the level of the leaves in an m-ary tree is Gaussian.

It was already noticed in [4] that, heuristically, there seems to be a strong limit theorem for the profile of binary search trees. Almost sure convergence is now established for the limiting behaviour of nodes in level k of binary search trees of size n in the central region $1.2\log n \le k \le 2.8\log n$ [1], by use of martingale methods and complex analysis.

Bibliography

[1] Chauvin (Brigitte), Drmota (Michael), and Jabbour-Hattab (Jean). – The profile of binary search trees. *The Annals of Applied Probability*, vol. 11, n° 4, 2001, pp. 1042–1062.

[2] Chern (Hua-Huai) and Hwang (Hsien-Kuei). – Phase changes in random m-ary search trees and generalized quicksort. *Random Structures & Algorithms*, vol. 19, n° 3-4, 2001, pp. 316–358. – Analysis of algorithms (Krynica Morska, 2000).

[3] Durand (Marianne). – *Holonomie et applications en analyse d'algorithmes et combinatoire*. – Mémoire de DEA, Projet Algorithmes, INRIA Rocquencourt, 2000.

[4] Flajolet (Philippe) and Sedgewick (Robert). – *The average case analysis of algorithms: multivariate asymptotics and limit distributions*. – Research Report n° 3162, Institut National de Recherche en Informatique et en Automatique, 1997. 123 pages.

[5] Hennequin (Pascal). – *Analyse en moyenne d'algorithmes, tri rapide et arbres de recherche*. – Thèse de doctorat, École polytechnique, March 1991. 162 pages.

[6] Knessl (Charles) and Szpankowski (Wojciech). – Quicksort algorithm again revisited. *Discrete Mathematics & Theoretical Computer Science*, vol. 3, n° 2, 1999, pp. 43–64.

[7] Martínez (Conrado) and Roura (Salvador). – Optimal sampling strategies in quicksort and quickselect. *SIAM Journal on Computing*, vol. 31, n° 3, 2001, pp. 683–705.

[8] Rösler (U.). – On the analysis of stochastic divide and conquer algorithms. *Algorithmica*, vol. 29, n° 1-2, 2001, pp. 238–261. – Average-case analysis of algorithms (Princeton, NJ, 1998).

[9] Rösler (Uwe). – A fixed point theorem for distributions. *Stochastic Processes and their Applications*, vol. 42, n° 2, 1992, pp. 195–214.

[10] Tan (Kok Hooi) and Hadjicostas (Petros). – Some properties of a limiting distribution in Quicksort. *Statistics & Probability Letters*, vol. 25, n° 1, 1995, pp. 87–94.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 63–68.

Available online at the URL
http://algo.inria.fr/seminars/.

Traveling Waves and the Height of Binary Search Trees

Michael Drmota
Institut für Geometrie, Technische Universität Wien (Austria)

September 24, 2001

Summary by Brigitte Chauvin

1. Introduction

Binary search trees are widely used to store (totally ordered) data, and many parameters have been discussed in the literature (the monograph of Mahmoud [6] gives a very good overview of the state of the art). Starting from a permutation of $\{1, 2, \ldots, n\}$ we get a binary tree T_n with n internal nodes such that the keys of the left subtree of any given node x are smaller than the key of x and the keys of the right subtree are larger than the key of x. Usually it is assumed that every permutation of $\{1, 2, \ldots, n\}$ is equally likely and hence any parameter of binary search trees may be considered as a random variable.

Here we consider the height H_n which is the largest distance of an internal node from the root. In 1986, Devroye [2] proved that the expected value $\mathbf{E}\, H_n$ satisfies the asymptotic relation

$$\mathbf{E}\, H_n \sim c \log n, \tag{1}$$

and it is also proved [1] that

$$\frac{H_n}{c \log n} \to 1 \quad a.s., \tag{2}$$

(as $n \to \infty$), where $c = 4.31107\ldots$ is the (largest real) solution of the equation

$$\left(\frac{2e}{c}\right)^c = e. \tag{3}$$

Better bounds for the expected value were given by two completely different methods by Devroye and Reed [3] and by Drmota [4]. Finally Drmota [5] and Reed [8, 9] proved the so-called Robson conjecture

$$\mathbf{V} H_n = \mathcal{O}(1). \tag{4}$$

Reed [8, 9] was also able to obtain a very precise bound for the expected value:

$$\mathbf{E}\, H_n = c \log n - \frac{3c}{2(c-1)} \log \log n + \mathcal{O}(1). \tag{5}$$

Notice that properties analogous to (1) and (2) hold for the (dual) saturation level H'_n with constant c replaced by the other real solution of Equation (3) [1, 5, 6].

Here, the purpose is to obtain more precise information on the asymptotic behaviour of the distribution of the height H_n. This will also lead to a perspective of improving (4) and (5). To this end, we first need to understand the two main ideas. They are:

1. an analytic approach, due to Drmota, of the generating function
$$Y_k(z) = \sum_{n\geq 0} \mathbf{P}(H_n \leq k) z^n$$
2. Devroye's connection between Binary Search Trees (bst) and Branching Random Walks (brw), which allows to use the above analytic approach to a "close" model (brw), easier to deal with. Moreover, the analytic approach is applied to the Random Bisection Problem, considered as a brw with a continuous parameter.

This seminar is devoted to connect such methods to some facts and results. Very precise estimates are shown to be consequences of rather natural conjectures.

2. **Results and Conjectures**

Following the analytic approach, the generating function
$$Y_k(z) = \sum_{n\geq 0} \mathbf{P}(H_n \leq k) z^n$$
is a solution of the difference equation

$$\begin{cases} Y_0(z) = 1 \\ Y'_{k+1}(z) = Y_k(z)^2, \qquad Y_k(0) = 1. \end{cases} \tag{6}$$

For
$$x_k := Y_k(1) = \sum_{n\geq 0} \mathbf{P}(H_n \leq k),$$
it is shown in [4, 5] that x_k is related to $\mathbf{E}\, H_n$ by the following result.

Fact 1.
$$\mathbf{E}\, H_n = \max\{\, k \mid x_k \leq n \,\} + \mathcal{O}(1).$$

We also already noticed the following result by Reed [8, 9].

Fact 2.
$$\mathbf{E}\, H_n = c \log n - \frac{3c}{2(c-1)} \log\log n + \mathcal{O}(1).$$

Together, Facts 1 and 2 give the following bounds:
$$c_2 \alpha^k k^\beta \leq x_k \leq c_1 \alpha^k k^\beta$$
where $\alpha = e^{1/c}$ and $\beta = \frac{3}{2(c-1)}$.

It follows that the following conjectures are quite natural.

Conjecture 1.
$$x_k \sim \gamma \alpha^k k^\beta \qquad (k \to +\infty).$$

Conjecture 2.
$$\lim_{k\to+\infty} \frac{x_{k+1}}{x_k} \text{ exists.}$$

Assume for a while that Conjecture 2 is true,[1] then the following theorem holds.

[1]Recently Conjecture 2 could be verified so that Theorem 1 is now an unconditioned result.

Theorem 1. *There exists some distribution function $F(x)$ such that*

$$\mathbf{P}(H_n \leq k) = F(\log n - \log x_k) + o(1) \tag{7}$$

uniformly in k as $n \to +\infty$.

Let us point out here that, if Conjecture 1 is true, there exists some distribution function $F(x)$ such that

$$\mathbf{P}(H_n \leq k) = F\left(\log n - \frac{1}{c}k - \beta \log k\right) + o(1) \tag{8}$$

uniformly in k as $n \to +\infty$. The limit distribution F which appears in (7) and (8) can be understood as a traveling wave.

As another consequences of Conjecture 1, precise estimates of the first and second moment of the height are:

$$\mathbf{E}(H_n) = c \log n - \frac{3c}{2(c-1)} \log\log n + \Delta_1\left(c \log n - \frac{3c}{2(c-1)} \log\log n\right) + o(1)$$

and

$$V(H_n) = \Delta_2\left(c \log n - \frac{3c}{2(c-1)} \log\log n\right) + o(1)$$

where Δ_1 and Δ_2 are continuous, periodic functions with period 1.

There is an intimate relation of Random Binary Search Trees to Devroye's Tree Model, resp. a relation between a Binary Search Tree and a Branching Random Walk. Recall that in this connection, the considered Branching Random Walk is defined by an infinite binary tree with weights $\tilde{U}$, equal to U or $1-U$ on left and right edges respectively (U denotes a uniform random variable on $[\,0,1\,]$). In this model, each node v of the tree has a weight

$$l(v) = \prod_{e<v} \tilde{U}_e.$$

Let the tree $\bar{T}_n$ be defined by

$$\bar{T}_n := \left\{ v \;\middle|\; l(v) \geq \frac{1}{n} \right\},$$

and let $\bar{H}_n$ denotes the height of $\bar{T}_n$. Devroye has shown that the distribution of $\bar{H}_n$ is "very close" to that of H_n.

Let us see now why the the distribution of $\bar{H}_n$ is close to that of H_n. We work in terms of the Random Bisection Problem (which is a reformulation of $\bar{H}_n$): in that problem, an interval with length x is randomly cut into two intervals with length $x_1 := Ux$ and $x_2 := (1-U)x$, where U is uniformly distributed on $[\,0,1\,]$.

Let $P_k(x,l)$ be the probability that all segments are less than l after k steps, and let

$$\bar{P}_k\left(\frac{x}{l}\right) := P_k\left(\frac{x}{l}, 1\right) = P_k(x,l),$$

then $\bar{P}_k(x)$ looks like a wave, and is a solution of the following recursion:

$$\bar{P}_{k+1}(x) = \frac{1}{x}\int_0^x \bar{P}_k(y)\bar{P}_k(x-y)\,dy.$$

By definition of P_k, $\bar{H}_n$, $\bar{T}_n$,

$$\bar{P}_k(n) = P_k\left(1, \frac{1}{n}\right) = \mathbf{P}\left(\bar{H}_n \leq k\right)$$

so that the Random Bisection Problem appears as a generalized tree model with continuous parameter x:

$$\bar{T}_x = \left\{ v \;\middle|\; l(v) \geq \frac{1}{x} \right\}, \qquad \bar{H}_x = \text{height of } \bar{T}_x.$$

For this generalized tree model, the analytic approach is close to that for Binary Search Trees and it provides an analogy between H_n and $\bar{H}_n$: let

$$\bar{Y}_k(z) := \int_0^\infty \bar{P}_k(x) e^{(z-1)x}\, dx = \int_0^\infty P\left(\bar{H}_x \leq k\right) e^{(z-1)x}\, dx$$

then

$$\bar{Y}_0(z) = \frac{1}{z-1}(e^{z-1} - 1)$$

and

$$\bar{Y}'_{k+1}(z) = \bar{Y}_k(z)^2. \tag{9}$$

For

$$\bar{x}_k := \bar{Y}_k(1) = \int_0^\infty \bar{P}_k(x)\, dx = \int_0^\infty P\left(\bar{H}_x \leq k\right)\, dx$$

we have the following results.

Fact 1'.

$$\mathbf{E}\, \bar{H}_n = \max\{\, k \mid x_k \leq n \,\} + \mathcal{O}(1) \qquad (n \to \infty).$$

Fact 2'.

$$\begin{aligned} \mathbf{E}\, \bar{H}_n &= \mathbf{E}\, H_n + \mathcal{O}(1) \\ &= c \log n - \frac{3c}{2(c-1)} \log\log n + \mathcal{O}(1) \qquad (n \to \infty). \end{aligned}$$

Both results imply

$$\bar{c}_2 \alpha^k k^\beta \leq \bar{x}_k \leq \bar{c}_1 \alpha^k k^\beta$$

for the same constants α and β. Analogous conjectures are

Conjecture 1'.

$$\bar{x}_k \sim \bar{\gamma} \alpha^k k^\beta \qquad (k \to +\infty).$$

Conjecture 2'.

$$\lim_{k \to +\infty} \frac{\bar{x}_{k+1}}{\bar{x}_k} \text{ exists.}$$

Note that Conjectures 1 and 1' on the one hand, and Conjectures 2 and 2' on the other hand, are equivalent. Admitting these conjectures, the following theorem can be deduced as well:

Theorem 2. *If Conjecture 2' is true,*[2] *there exists some distribution function $\bar{F}(x)$ such that*

$$\mathbf{P}\left(\bar{H}_n \leq k\right) = \bar{P}_k(n) = \bar{F}(\log n - \log \bar{x}_k) + o(1) \tag{10}$$

uniformly in k as $n \to +\infty$.

If Conjecture 1' is true, there exists some distribution function $\bar{F}(x)$ such that

$$\mathbf{P}\left(\bar{H}_n \leq k\right) = \bar{P}_k(n) = \bar{F}\left(\log n - \frac{k}{c} - \beta \log k\right) + o(1) \tag{11}$$

uniformly in k as $n \to +\infty$. The limit distribution $\bar{F}$ which appears in (10) and (11) can be understood as a traveling wave.

[2] ... which has been verified

Note that $F(x)$ of Theorem 1 and $\bar{F}(x)$ of Theorem 2 in fact coincide.

3. **Sketch of Proof**

To prove Theorem 1 (and similarly Theorem 2) it is necessary to get information on $\bar{Y}_k(x)$, the solution of Equation (6) (resp. of (9)). The method consists in considering an auxiliary function $\tilde{Y}_k(x)$, related to a solution of the Retarded Differential Equation with a parameter α:

$$\Phi'(u) = -\frac{1}{\alpha^2}\Phi\left(\frac{u}{\alpha}\right)^2, \qquad \Phi(0) = 1,$$

by

$$\tilde{Y}_k(x) := \alpha^k \Phi\left(\alpha^k(1-x)\right) \qquad (k \in \mathbb{R}).$$

The Retarded Differential Equation can be solved, because Φ is the Laplace transform of some function Ψ

$$\Phi(u) := \int_0^\infty \Psi(y) e^{-uy}\, dy$$

solution of the integral equation

$$y\Psi\left(\frac{y}{\alpha}\right) = \int_0^y \Psi(z)\Psi(y-z)\, dz.$$

The existence and unicity of solutions of this integral equation, considered as a fixed-point equation, come from a contraction method which applies only for values of parameter α between 1 and a critical value $\alpha_0 = e^{1/c} = 1.26\ldots$

The relation between the auxiliary function $\tilde{Y}_k(x)$ and the true function $Y_k(x)$ relies on a scaling: define e_k by

$$\alpha^{e_k} = x_k,$$

then, locally around $x = 1$,

$$Y_k(z) \sim \tilde{Y}_{e_k}(x),$$

at least if Conjecture 2 is right!, i.e.,

$$\lim_{k\to\infty} \frac{x_{k+1}}{x_k} = \alpha.$$

Then, it remains to extract the coefficient with degree n in $Y_k(x)$

$$\mathbf{P}(H_n \le k) = [x_n]\, Y_k(x) = \Psi(n/x_k) + o(1)$$

to get by comparison with $\tilde{Y}_{e_k}(x)$, the asymptotics of Theorem 1:

$$\mathbf{P}(H_n \le k) \sim F(\log n - \log x_k)$$

with $F(x) = \Psi(\log x)$.

As a last remark, it is worth to connect the above objects, especially $\bar{x}_k$, to some heuristics in statistical physics literature (see for instance [7]), where quite similar traveling waves appear. There, $\bar{x}_k$ is the front position, it increases as $\alpha^k k^\beta$ (Conjecture 1') and parameter α of the Retarded Differential Equation is nothing but the velocity of the front wave.

Bibliography

[1] Biggins (J. D.). – How fast does a general branching random walk spread? In *Classical and modern branching processes (Minneapolis, MN, 1994)*, pp. 19–39. – Springer, New York, 1997.

[2] Devroye (Luc). – A note on the height of binary search trees. *Journal of the Association for Computing Machinery*, vol. 33, n° 3, 1986, pp. 489–498.

[3] Devroye (Luc) and Reed (Bruce). – On the variance of the height of random binary search trees. *SIAM Journal on Computing*, vol. 24, n° 6, 1995, pp. 1157–1162.

[4] Drmota (M.). – An analytic approach to the height of binary search trees. *Algorithmica*, vol. 29, n° 1-2, 2001, pp. 89–119. – Average-case analysis of algorithms (Princeton, NJ, 1998).

[5] Drmota (Michael). – The variance of the height of binary search trees. *Theoretical Computer Science*, vol. 270, n° 1-2, 2002, pp. 913–919.

[6] Mahmoud (Hosam M.). – *Evolution of random search trees*. – John Wiley & Sons, New York, 1992, *Wiley-Interscience Series in Discrete Mathematics and Optimization*, xii+324p.

[7] Majumdar (Satya N.) and Krapivsky (P. L.). – Traveling waves, front selection, and exact nontrivial exponents in a random fragmentation problem. *Physical Review Letters*, vol. 85, n° 26, 2000, pp. 5492–5495.

[8] Reed (Bruce). – How tall is a tree? In *Proceedings of the thirty-second annual ACM symposium on Theory of computing (Portland, Oregon, United States)*, pp. 479–483. – 1999. Proc. of STOC'00.

[9] Reed (Bruce). – The height of a random binary search tree. *Journal of the Association for Computing Machinery*, vol. 50, n° 3, 2003, pp. 306–332.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 69–72.

Available online at the URL
http://algo.inria.fr/seminars/.

Microscopic Behavior of TCP

Philippe Robert
INRIA Rocquencourt

February 11, 2002

Summary by Christine Fricker

Abstract

TCP (Transmission Control Protocol) is the ubiquitous data transfer protocol in communication networks. We focus on the control of the congestion of TCP. One long TCP connexion is studied when the loss rate of a packet tends to zero. It is shown that the Markov processes renormalized in a suitable way converge to a limit related to an auto-regressive Markov process. From a probalistic point of view, exponential functionals associated to compound Poisson processes play a key role. Analytically, the natural framework of this study turns out to be q-calculus. The talk presents a joint work with Vincent Dumas, Fabrice Guillemin and Bert Zwart (see [2] and [3]).

1. Introduction

In communication networks, sources send data to destinations via routers with limited capacity and TCP is a protocol which allows to transmit data with reliability in a loss network. The basic principles of TCP are due to Cerf and Kahn in 1973 and are based on acknowledgment: a source transmits at most W packets without response from destination. The control of congestion is due to Jacobson in 1987. Roughly speaking, if W packets are successfully transmitted, then the so-called congestion window size W is incremented by one; if a packet is lost, then W is divided by 2 (more generally multiplied by a factor δ). This is of course a simplification of the real algorithms involved, but the basic mechanism of reducing the congestion (called congestion avoidance) is captured by this model. Other algorithms (Slow Start, Fast Retransmit, and Fast Recovery) are also discussed (see [3]).

Consider the exchange between the source and the destination: each packet has some probability of being lost. The influence of the network is described in our model only through this loss process. We assume first that the packets are lost independently (a more general model where packets are lost by bursts will be considered in Section 4). Thus the sequence of the congestion window sizes is a Markov chain (W_n^α) on $\mathbb{N}$ with probability transitions

$$p\big(x, \min(x+1, w_{\max})\big) = e^{-\alpha x},$$
$$p\big(x, \lfloor \delta x \rfloor\big) = 1 - e^{-\alpha x},$$

where $w_{\max}$ is the maximum congestion size, $\delta \in (0,1)$ and $\alpha > 0$. The problems of special interest are estimations of the throughput (defined here as the mean congestion window size) and the stationary behavior. Asymptotic estimates will be presented when the loss rate α tends to zero.

Among other works, simulations are due to Floyd and Madhavi. Approximated models are investigated by Ott et al. [4] and Padhye et al., and analytical results are due to Adjih et al., Altman et al. [1], and Baccelli et al.

2. Convergence Results When the Loss Rate Tends to Zero

The main result of this section is that the congestion window size is of the order of $1/\sqrt{\alpha}$ when the loss rate or equivalently α tends to zero. For the sake of simplicity, assume that the maximum window size $w_{\max}$ is infinite.

Theorem 1. *If* $\lim_{\alpha\to 0}\sqrt{\alpha}\,W_0^\alpha = \overline{w}$ *and* $W^\alpha(t) = \sqrt{\alpha}\,W^\alpha_{\lfloor t/\sqrt{\alpha}\rfloor}$ *then* $\big(W^\alpha(t)\big)$ *converges in distribution to the Markov process* $\big(\overline{W}(t)\big)$ *given by* $\overline{W}(0) = \overline{w}$ *whose infinitesimal generator is*

$$\Omega(f)(x) = f'(x) + x\big(f(\delta x) - f(x)\big). \tag{1}$$

where f *is* $\mathcal{C}^1$ *on* $\mathbb{R}^+$.

A similar result is also valid for the embedded process (V_n^α) on $\mathbb{N}$ where V_n^α is the state of the Markov chain (W_n^α) just after the nth loss. It is clearly a Markov chain whose transitions are such that if $V_0^\alpha = x \geq 1$ then

$$V_1^\alpha = \big\lfloor \delta(x + G_x^\alpha)\big\rfloor$$

where $\mathbf{P}(G_x^\alpha \geq m) = \exp\big(-\alpha(mx + m(m-1)/2)\big)$. Indeed, $\sqrt{\alpha}$ is the right scaling for (G_x^α) and (V_n^α).

Proposition 1. *For* $x \in \mathbb{R}^+$, *as* α *tends to zero, the sequence* $\big(\sqrt{\alpha}\,G^\alpha_{\lfloor x/\sqrt{\alpha}\rfloor}\big)$ *converges in distribution to a random variable* $\overline{G}_x$ *with the property that for* $y \geq 0$,

$$\mathbf{P}\big(\overline{G}_x \geq y\big) = \exp^{-(xy+y^2/2)}. \tag{2}$$

Theorem 2. *If* $\lim_{\alpha\to 0}\sqrt{\alpha}\,V_0^\alpha = \overline{v}$ *then* $(\sqrt{\alpha}\,V_n^\alpha)$ *converges in distribution to the Markov chain* $\big(\overline{V}_n\big)$ *with* $\overline{V}_0 = \overline{v}$ *and transitions*

$$\overline{V}_{n+1} = \delta\big(\overline{V}_n + \overline{G}_{\overline{V}_n}\big).$$

3. The Equilibrium

Up to now, a closed form expression for the invariant probabilities of the Markov chains (W_n^α) and (V_n^α) is not known, but only bounds in some special cases. Nevertheless these invariant probability measures converge in distribution when α tends to 0 respectively to the distribution of $\overline{W}_\infty$, a random variable with distribution the invariant distribution of $\big(\overline{W}(t)\big)$ and of $\overline{V}_\infty$, a random variable with distribution the invariant distribution of $\big(\overline{V}_n\big)$. These limiting probabilities have rather simple closed form expressions. The key argument is the following result.

Lemma 1. *For* $x > 0$, *if* $\overline{G}_x$ *is defined by (2) then*

$$\big(x + \overline{G}_x\big)^2 \overset{dist.}{=} 2E_1 + x^2$$

where E_1 *is an exponentially distributed random variable wih parameter 1.*

It implies the important fact that the square of the limiting embedded Markov chain $\big(\overline{V}_n^2\big)$ is an AR (auto-regressive) process. By definition a process (X_n) is AR if and only if $X_{n+1} = A_nX_n + B_n$ where (A_n) and (B_n) are i.i.d. In Altman [1] the AR property is assumed, a priori, for the Markov chain $\big(\overline{V}_n\big)$ itself. The following result presents this property which leads to a close form expression for the distribution of $\overline{V}_\infty$ and its density function.

Proposition 2. *The sequence* $\big(\overline{V}_n^2\big)$ *is AR. More precisely for* $n \in \mathbb{N}$,

$$\overline{V}_{n+1}^2 = \delta^2\big(\overline{V}_n^2 + 2E_n\big)$$

where (E_n) *is an i.i.d. sequence of exponentially random variables with parameter 1. The distribution of* $\overline{V}_\infty$ *is thus given by*

$$\overline{V}_\infty \overset{dist.}{=} \sqrt{2\sum_{n=1}^{+\infty}\delta^{2n}E_n} \overset{dist.}{=} \sqrt{2\int_0^{+\infty}\delta^{2N(s)}\,ds}$$

where N *is a Poisson process with parameter 1. The density function* h_δ *of* $\overline{V}_\infty$ *is given by*

$$h_\delta(x) = \frac{1}{\prod_{n=1}^{+\infty}(1-\delta^{2n})}\sum_{n=1}^{+\infty}\frac{1}{\prod_{k=1}^{n-1}(1-\delta^{-2k})}\delta^{-2n}xe^{-\delta^{-2n}x^2/2} \qquad (x \geq 0).$$

The throughput of the TCP model is defined in the literature by $\rho^\alpha(\delta) = \lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^n W_k^\alpha$. The ergodic theorem for the Markov chain (W_n^α) gives that $\rho^\alpha(\delta) = \mathbf{E}(W_\infty^\alpha)$. Using the embedded chain $\big(\overline{V}_n\big)$ and defining the asymptotic throughput as $\overline{\rho}(\delta) = \lim_{\alpha\to 0}\sqrt{\alpha}\,\rho^\alpha(\delta)$, the following result can be deduced from Proposition 2.

Corollary 1. *The asymptotic throughput of the TCP model when* α *tends to 0 is given by*

$$\overline{\rho}(\delta) = \frac{\delta}{(1-\delta)\,\mathbf{E}(\overline{V}_\infty)} = \sqrt{\frac{2}{\pi}}\prod_{n=1}^{+\infty}\frac{1-\delta^{2n}}{1-\delta^{2n-1}}.$$

Remark. For the case of TCP, δ is set to 2 and the throughput $\overline{\rho}(1/2)$ is approximately 1.3098, which is the value observed in earlier simulations and experiments by Floyd, Padhye, and Madhavi.

4. A More General Model

A model with correlated losses generalizes the previous one. The evolution of W_n^α, the congestion window size over the nth RTT (Round Trip Time) interval, i.e., the maximum number of packets that can be transmitted without receiving acknowledgement from destination, is given as previously by the AIMD (additive increase, multiplicative decrease) algorithm: $W_{n+1}^\alpha = W_n^\alpha + 1$ when none of the W_n^α packets is lost and $W_{n+1}^\alpha = \max\big(\lfloor W_n^\alpha\rfloor, 1\big)$ otherwise. Nevertheless packet losses occur by clumps: if a packet is lost then several packets are lost during the following RTT intervals. These "clumps" are i.i.d. (see [3] for a complete definition). In particular if X_n is the number of losses in the nth clump then (X_n) is i.i.d. Though (W_n^α) is not a Markov chain, the embedded chain at the end of the consecutive clumps (V_n^α) is still Markov. Thus convergence results of Section 2 when the loss rate α tends to zero are valid with the infinitesimal generator in Theorem 1

$$\Omega(f)(x) = f'(x) + x\int_{\mathbb{R}^+}\big(f(\delta^u x) - f(x)\big)\,\mathbf{P}_{X_1}(du)$$

where the distribution of X_1 is denoted by $\mathbf{P}_{X_1}$ and δ replaced by δ^{X_1} in Theorem 2. As to Section 3, Proposition 2 is replaced by the following.

Proposition 3.

$$\overline{V}_\infty^2 = \delta^{2X_1}\big(\overline{V}_\infty^2 + 2E_1\big)$$

where X_1, E_1, *and* $\overline{V}_\infty$ *are independent random variables,* E_1 *being a random variable with an exponential distribution with parameter 1.*

Let I be a solution to

$$I \overset{dist.}{=} \beta^{X_1} I + E_1$$

where $\beta = \delta^2$ and E_1, I, and X_1 are independent. By definition, it turns out that I is the exponential functional associated to the Lévy process $Y(t) = \log \frac{1}{\beta} \sum_{k=1}^{N(t)} X_k$, N being a Poisson process with parameter 1. This functional occurs in mathematical finance (Asian options) where the Lévy process is generally a brownian motion with drift. In this setting, Bertoin, Carmona, Monthus, Petit, Yor, and many others (see for example Yor [5] for a survey) proved that the density of I is the solution of an integro-differential equation and that the moments of I are known. We present here an expression of the distribution of I for some special cases ($X_1 = 1$, X_1 with exponential distribution, and X_1 having a rational generating function). The Laplace transform of I can be expressed as a q-hypergeometric function (see [3] for details). The following proposition gives its fractional moments, in particular $E(\sqrt{I})$.

Proposition 4. *For each real s, if $-s$ is not in $\mathbb{N}^*$, $\mathbf{E}\big(\beta^{(s+1)X_1}\big) < \infty$ and $\mathbf{E}\big(\frac{1}{1-\beta^{X_1}}\big) < \infty$ then*

$$\mathbf{E}(I^s) = \Gamma(s+1) \prod_{k=1}^{+\infty} \frac{1 - \mathbf{E}\big(\beta^{(s+k)X_1}\big)}{1 - \mathbf{E}(\beta^{kX_1})}.$$

As a sketch of the proof, to obtain the fractional moments, if $\psi(\lambda) = \mathbf{E}(e^{-\lambda} I)$ then, from the definition of I, we derive

$$\psi(\lambda) = \frac{1}{1+\lambda}\, \mathbf{E}\big(\psi(\lambda \beta^{X_1})\big),$$

which gives a simple recurrence relation on the Mellin transform $\psi^*(s) = \int_0^{+\infty} \psi(\lambda) \lambda^{s-1}\, d\lambda$ of ψ. Then, using the fact that $\psi^*(s) = \mathbf{E}(I^{-s})\Gamma(s)$ for $\Re(s) > 0$, one proves the result.

As in the independent losses model, asymptotic throughput when α tends to zero can be derived.

Theorem 3. *The asymptotic throughput for the correlated model when α tends to zero is given by*

$$\overline{p}_{X_1}(\delta) = \lim_{\alpha \to 0} \sqrt{\alpha\, \mathbf{E}(X_1)}\, \rho^{\alpha} = \sqrt{\frac{2\, \mathbf{E}(X_1)}{\pi}} \prod_{n=1}^{+\infty} \frac{1 - \mathbf{E}\big(\delta^{2nX_1}\big)}{1 - \mathbf{E}\big(\delta^{(2n-1)X_1}\big)}.$$

To conclude it is possible to compare throughputs for different distributions of X_1. In particular, the throughput for the independent losses model is a lower bound for the throughput of a correlated losses model (see [3] for details).

Bibliography

[1] Altman (E.), Avrachenkov (K.), and Barakat (C.). – A stochastic model of TCP/IP with stationary random losses. In *Proceedings ACM-SIGCOMM'00*, vol. 4, pp. 231–242. – Stockholm, 2000.

[2] Dumas (Vincent), Guillemin (Fabrice), and Robert (Philippe). – A Markovian analysis of additive-increase multiplicative-decrease algorithms. *Advances in Applied Probability*, vol. 34, n° 1, 2002, pp. 85–111.

[3] Guillemin (Fabrice), Robert (Philippe), and Zwart (Bert). – AIMD algorithms and exponential functionals. – April 2002. Preprint.

[4] Ott (Teunis J.), Kemperman (J. H. B.), and Mathis (Matt). – Window size behavior in TCP/IP with constant loss probability. – Available online at `http://www.psc.edu/networking/papers/papers.html#teunis`, November 1996.

[5] Yor (Marc). – *Exponential functionals of Brownian motion and related processes.* – Springer-Verlag, Berlin, 2001, x+205p.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), p. 73.

Available online at the URL
http://algo.inria.fr/seminars/.

Interaction Between Sources Controlled by TCP

François Baccelli
Projet Trec, Inria Rocquencourt and ENS (France)

February 11, 2002

Abstract

The interaction between TCP sources can be modeled by products of random matrices. We show how one can use this representation for the analysis and the simulation of large IP networks. Joint work with Dohy Hong.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 75–76.

Available online at the URL
http://algo.inria.fr/seminars/.

Asymptotic Analysis of TCP Performances Under Mean-field Approximation

Philippe Jacquet
INRIA Rocquencourt

September 24, 2001

Summary by Christine Fricker

Abstract

The talk deals with the performances of TCP when N connections share the same router with high capacity NT. Using mean field approximation, some asymptotic results on the throughput and the distribution of the size of the congestion window when N is large are established. The talk is based on a joint paper with Cédric Adjih and Nikita Vvedenskaya [1].

1. The Real Protocol and the Models

TCP (Transmission Control Protocol) is the protocol which controls 99.9% of the traffic on the Internet network. It is a end-to-end protocol where the destination sends acknowledgments to the source, which controls its congestion window according to them and retransmits the lost packets.

We present the study of the multi-connection case where N connections share the same router with finite capacity. Every user is loading a file with infinite size via a unique connection and adapts its congestion window according to TCP, which can be roughly described as follows: the size of the congestion window is increased by one each time a number of acknowledgments equal to the window size has been received; each time there is a packet loss, the user halves the size of its congestion window. Losses are due to the finite capacity of the buffer which receives packets from all the users. The model under study does not take into account refinements of the protocol like the *slotted time*, the *slow start*, and the *self-clocking* (see [1] for details).

The time between sending a packet and receiving the acknowledgment is called round-trip time (RTT). In the buffer with capacity TN, the service time of a packet is one and the RTT has an exponential distribution with mean N/λ. This model is highly unrealistic but analytically tractable. Nevertheless the analysis can be generalized to a RTT that is the sum of a fixed term NT and a delay with an exponential distribution with mean N/λ, much smaller than NT. This second model is much more realistic.

2. The Asymptotic Case

Let $R^N(t)$ be the free capacity in the buffer at time t and $W_i^N(t)$ the size of the congestion window of user i at time t. Let $R^N(x,t) = \mathbf{P}(R^N(t) > x)$ and $w^N(y,t) = -\frac{\partial}{\partial y} W^N(y,t)$ the density of the window size distribution. According to the dynamic of the system, the following equations

hold:

$$\frac{\partial}{\partial t}R^N(x,t) = -\frac{\partial}{\partial x}R^N(x,t) + \frac{\lambda}{N}\sum_{i=1}^{N}\Big(R^N(x+W_i^N(t),t) - R^N(x,t)\Big), \tag{1}$$

$$\frac{\partial}{\partial t}w^N(y,t) = \frac{\lambda}{N}\Big(R^N(y-1,t)w^N(y-1,t) + (1-R^N(2y,t))w^N(2y,t) - w^N(y,t)\Big). \tag{2}$$

These equations show a separation of time scales: $R^N(t)$ varies at rate of order λ and $w^N(y,t)$ obviously varies at rate λ/N. Therefore when N is large, $w^N(y,t)$ tends to be slowly varying and $R^N(t)$ reaches its steady state distribution $\tilde{R}$ where $w(y,t)$ is independent of t. When N is large, $R^N(t)$ converges to $R(t)$ satisfying, using Equation (1),

$$\frac{\partial}{\partial t}R(x,t) = -\frac{\partial}{\partial x}R(x,t) + \left(\int_0^{+\infty} R(x+y,t)\,dW(y) - R(x,t)\right)\lambda.$$

Thus $R(t)$ has a stationary limit $\tilde{R}$ which has an exponential distribution with parameter $a > 0$ such that $\lambda\Big(1 - \mathbf{E}\big(\exp(-aW)\big)\Big) = a$. Heuristically, when t tends to infinity, if $a(t)$ tends to a limit a then the limit solution $w(y)$ is the solution of

$$w(y) = e^{-a(y-1)}w(y-1) + (1-e^{-2ay})w(2y) \tag{3}$$

where

$$\left(1 - \int_0^{+\infty} e^{-ay}w(y)dy\right)/a = 1/\lambda.$$

In the case $a \ll 1$ (i.e., the loss rate tends to 0) and with the approximation that $w(y) = \sqrt{a}g(y\sqrt{a}) + O(a)$, Equation (3) becomes at first order

$$yg(y) + g'(y) = 2yg(2y). \tag{4}$$

It can be solved introducing its Mellin transform $g^*(s) = \int_0^{+\infty} g(y)y^{s-1}\,dy$ and it comes that

$$g(y) = \frac{2}{\pi}\prod_{k\geq 1}(1-4^{-k})^{-1}\sum_{n\geq 0}a_n 2^n \exp(-4^n y^2/2)$$

where $\sum_{n\geq 0} a_n x^n = \prod_{k\geq 1}(1-4^{-k}x)$. This result can be compared to the result by Dumas et al. [2]. Is is proved in their paper that if $\tilde{W}$ is the congestion window size just before a loss then $\tilde{W}^2$ has the same distribution as $2\sum_{k\geq 1} 2^{-2^k} I_k$ where I_k are i.i.d. variables with exponential distribution of parameter 1.

These analytical results, typically the distribution of the size of the congestion window, have been compared with simulations of two types: the previous *simplified* model of TCP and the *real* TCP, using the simulator ns2. In both cases of simulations, an oscillation of the size of the buffer occupancy from the limit capacity has been observed. The analytical mean value of the free buffer size agrees with the simulation of *simplified* TCP.

Bibliography

[1] Adjih (Cédric), Jacquet (Philippe), and Vvedenskaya (Nikita). – *Performance evaluation of a single queue under multi-user TCP/IP connections.* – Research Report n° 4141, Institut National de Recherche en Informatique et en Automatique, March 2001. 40 pages.

[2] Dumas (Vincent), Guillemin (Fabrice), and Robert (Philippe). – A Markovian analysis of additive-increase multiplicative-decrease algorithms. *Advances in Applied Probability*, vol. 34, n° 1, 2002, pp. 85–111.

Part IV

Asymptotics and Analysis

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 79–82.

Available online at the URL
http://algo.inria.fr/seminars/.

A Hyperasymptotic Approach of the Multi-Dimensional Saddle-Point Method

Éric Delabaere
Université d'Angers (France)

December 3, 2001

Summary by Marianne Durand

Abstract

This summary is a presentation of the saddle-point method and of one application. It concludes with an attempt to give some intuition towards problems that can arise (for example the Stokes phenomenon), and gives references for further exploration.

1. Introduction

The content of this summary is only an introduction to the presentation of Éric Delabaere on multi-dimensional saddle-point method. We will only consider the one-dimensional problem on an example: the Airy function. For a general study of the problem, see [3]. The first section briefly presents the saddle-point method for oscillating integrals, the second section presents an application where the Airy function appears, and the last section shows the consequences of the Stokes phenomenon on the saddle-point method. Finally we give references to the resurgence theory which is a general (and difficult) approach for this type of problems.

2. The Saddle-Point Method

The saddle-point method is a way to get an approximate value of an integral by a good deformation of the integration contour. Consider an integral of the type

$$I(x) = \int g(z)e^{ixf(z)}\,dz. \tag{1}$$

We suppose that the integrand is analytic in some domain of the complex plane, so that a deformation of the integration contour is allowed. If we view the complex plane with a parameter height equal to $|e^{ixf(z)}|$ then it is convenient to talk about valleys and hills to describe the integrand. In order to get a good approximation for large x, it is interesting to keep the contour in the zones where the integrand is as small as possible. This is realized by staying in the valleys, except when going from one valley to another, where the contour should cross saddle points by using steepest-descent paths. These saddle points are characterized by $f'(z) = 0$, and are the points that contribute the most to the integral I. Suppose that f' has only one zero located at z_0 (the case with multiple zeros is very similar), and expand g up to order 0 and f up to order 2, then I is rewritten as (neglecting error terms):

$$I(x) = \int g(z_0)e^{ixf(z_0)}e^{ix\frac{(z-z_0)^2}{2}f''(z_0)}\,dz.$$

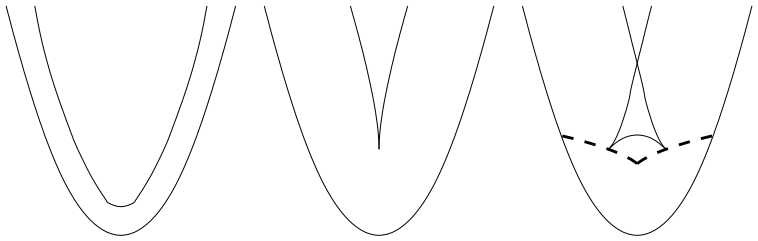


FIGURE 1. A parabolic light source, with the evolution of Σ_t.

The change of variable $u = (z - z_0)\left(\frac{xf''(z_0)}{2i}\right)^{1/2}$ gives the following approximation for I:

$$I(x) \sim g(z_0)e^{ixf(z_0)}\left(\frac{2\pi i}{xf''(z_0)}\right)^{1/2}.$$

When there are several saddle points (not too close), the result is obtained by adding the contributions of all of the saddle points.

3. One Application in Optics

In this section, we study an example from optics where an integral of type (1) occurs. This application is treated in [7, 8]. Consider a monochromatic source of light produced by a curve Σ. Following geometrical optic rules, the space is split into two distinct zones, one luminous and the other totally dark, but this model does not fit totally well with reality.

The Huygens principle describes light propagation in the space (filled with ether according to Huygens) by an analogy with the propagation of sound in the air, or waves on water. It says that each point of the light source is a punctual source that emits a spherical wave. The wave surface Σ_t at a time t is thus the envelope of the spherical wave surfaces of all the punctual sources. We easily deduce from the Huygens principle that Σ_t is the location of the points at distance ct of the source (in the proper direction), where c is the speed of light. An example is given for a parabolic luminous source in Figure 1. As t grows, the curve Σ_t starts to have cusps. The location of all these cusps is called the *caustic* (represented with a dotted line on Figure 1). Another way of observing the phenomenon is to trace all the normal lines to the luminous source, the caustic appears naturally as the accumulation of lines, or for a real luminous source, by an accumulation of light, see Figure 2.

On the caustic itself, there appears some interference fringes that cannot be explained by the sole Huygens principle. At the beginning of the 19th century, Fresnel completed the Huygens principle into the Huygens–Fresnel principle by adding an amplitude and and a phase to the wave, both depending on the position and on the time. So up to a multiplicative constant, the electromagnetic amplitude $\Psi(p,t)$ is

$$\Psi(p,t) \propto \int_\Sigma \frac{e^{ikd(p,q)}}{d(p,q)}\,dq, \tag{2}$$

where k is the wave number, and $d(x,y)$ the distance between x and y. The luminous intensity is then proportional to the square of the electro-magnetic amplitude.

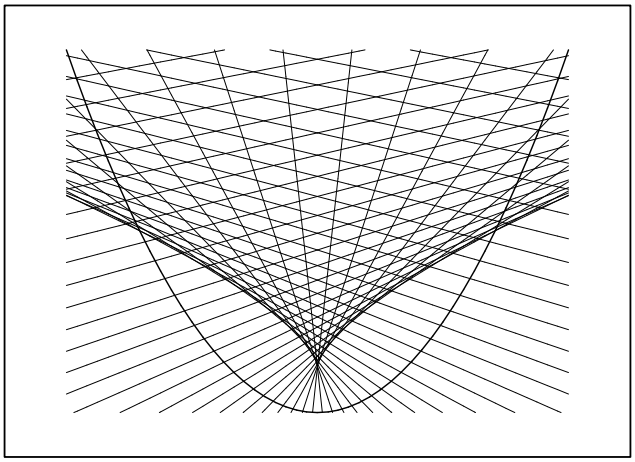


FIGURE 2. A parabolic source of light, and some of its normal rays.

For a point p located near the caustic and far enough from the source Σ, Equation (2) can be approximated by

$$\mathrm{Ai}\left(\left(\frac{2k^2}{\rho}\right)^{1/3} y\right),$$

with ρ the curving ray of the caustic, y the distance between p and the caustic, and Ai the Airy function defined by $\mathrm{Ai}(w) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i(z^3/3+wz)}\,dz$.

4. **The Stokes Phenomenon**

This section shows how the Stokes phenomenon appears on the example of the Airy function introduced in the previous section. A more detailed study can be found in [1].

The Airy function is approximated using the saddle-point method, that consists in choosing the integration path along the steepest descent lines of $\Re\big(i(z^3/3+wz)\big)$, passing by the saddle points. This path depends on the value of w, and in fact only on the argument of w, so we assume that $|w| = 1$.

Figure 3 shows various integration paths represented by a thick line and oriented from left to right, depending on the argument of w. The saddle points and the lines of steepest descent are also drawn. The Stokes phenomenon can be seen on this figure. First when $\mathrm{Arg}\, w = \pi/3$, only one

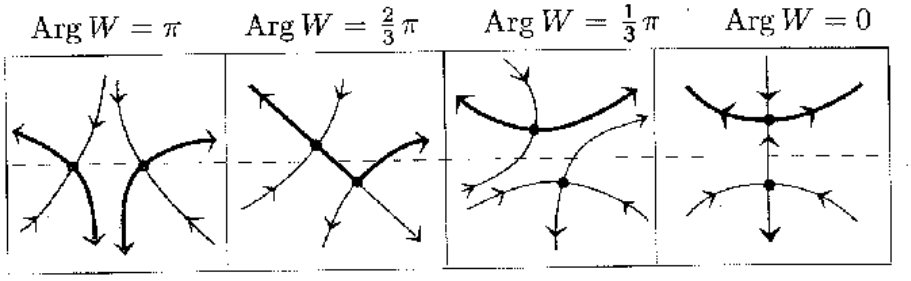


FIGURE 3. The integration path for various w

saddle point contributes to the asymptotic (as the integration path goes only through one saddle point). When the integration path goes through the second saddle point, for $\operatorname{Arg} w = 2\pi/3$, both saddle points compete for the asymptotic (this occurs when a line of steepest descent descending from a saddle point goes through another saddle point). The added contribution is negligible, so the asymptotic remains the same, but the Borel summability is lost. This is the Stokes phenomenon. And the line defined by $\operatorname{Arg} w = 2\pi/3$ is a Stokes line. Then when $\operatorname{Arg} w$ grows up to π, both saddle points contribute. The change of Borel summabillity is handled very well by the theory of resurgent functions due to Écalle [4, 5, 6] (see [2] for an introduction).

The resurgent point of view can be generalized to oscillating integrals of higher dimension, and has the interesting property of giving an exact coding of the integral by resurgent symbols, and not only an asymptotic expansion.

Bibliography

[1] Candelpergher (B.), Nosmas (J.-C.), and Pham (F.). – Premiers pas en calcul étranger. *Annales de l'Institut Fourier*, vol. 43, n° 1, 1993, pp. 201–224.

[2] Candelpergher (Bernard). – Une introduction à la résurgence. *Gazette des Mathématiciens*, n° 42, 1989, pp. 36–64.

[3] Delabaere (E.) and Howls (C. J.). – Global asymptotics for multiple integrals with boundaries. *Duke Mathematical Journal*, vol. 112, n° 2, 2002, pp. 199–264.

[4] Écalle (Jean). – *Les fonctions résurgentes. Tome I.* – Université de Paris-Sud, Département de Mathématique, Orsay, 1981, *Publications Mathématiques d'Orsay 81 [Mathematical Publications of Orsay 81]*, vol. 5. Les algèbres de fonctions résurgentes. [The algebras of resurgent functions]. 247 pages.

[5] Écalle (Jean). – *Les fonctions résurgentes. Tome II.* – Université de Paris-Sud, Département de Mathématique, Orsay, 1981, *Publications Mathématiques d'Orsay 81 [Mathematical Publications of Orsay 81]*, vol. 6, 248–531p. Les fonctions résurgentes appliquées à l'itération. [Resurgent functions applied to iteration].

[6] Écalle (Jean). – *Les fonctions résurgentes. Tome III.* – Université de Paris-Sud, Département de Mathématiques, Orsay, 1985, *Publications Mathématiques d'Orsay [Mathematical Publications of Orsay]*, vol. 85. L'équation du pont et la classification analytique des objects locaux. [The bridge equation and analytic classification of local objects]. 587 pages.

[7] Pham (Frédéric). – Caustiques : aspects géométriques et ondulatoires. In *Leçons de mathématiques d'aujourd'hui. Volume 1.* pp. 277–306. – Cassini, 2003.

[8] Soares (Manuel) and Vallée (Olivier). – *Les fonctions d'Airy pour la physique.* – Diderot, 1998, xi+174p.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 83–88.

Available online at the URL
http://algo.inria.fr/seminars/.

Ramanujan's Summation

Éric Delabaere
Université d'Angers (France)

December 3, 2001

Summary by Vincent Puyhaubert

Abstract

Ramanujan has brought a number of impressive results to analysis. Some of them have been obtained by a very free use of divergent series, which tends to show that he possessed an intuitive summation process for such divergent series, a process that could even depend of the context. The first step of our analysis is based on some considerations of Ramanujan from Chapter VIII of his Notebooks.

1. Introduction

The famous self-taught Indian mathematician Ramanujan (1887–1920) was accustomed to using convergent as well as divergent series freely in his derivation of identities. Most of the results reported on his notebooks were proven to be true, even if the ways he used to find them were not always rigorous. Actually, behind his way of thinking, a few summation schemes have been detected like Borel summation and what is called here the Ramanujan summation.

At the beginning of the 8th chapter of his Notebooks, as it is reported in Berndt's account [2], Ramanujan starts with the Euler–Maclaurin formula

$$a(1)+a(2)+\cdots+a(x-1)=C+\int_1^x a(t)dt+\sum_{k\geq 1}\frac{b_k}{k!}\partial^{k-1}a(x), \tag{1}$$

and remarks that the constant C entertains a mysterious relationship with the series—it is like its "center of gravity"—so that Ramanujan proposes to consider it as the sum of the serie. As an exemple, this process assigns the value γ to the sum $\sum_{n=1}^{+\infty}\frac{1}{n}$. The work of Delabaere attempts to make this idea rigorous, in a suitable space of analytic functions.

Let $a(x)$ be a function analytic in the right half-plane $P=\{\,x \mid \Re(x)>0\,\}$. First of all, the divergent serie $\sum_{n\geq 1}a(n)$ is considered as a formal expression. Let us introduce the tail of the series $R(x):=\sum_{n\geq 0}a(n+x)$. Then, R is a formal solution of the difference equation $R(x)-R(x+1)=a(x)$ and $\sum_{n\geq 1}a(n)=R(1)$. The problem of summation is then reduced to solving a difference equation.

2. Formal Solutions

As a first approach, using the Taylor formula, we write:

$$R(x+1)=\sum_{k\geq 0}\frac{1}{k!}\partial^k f(x)=e^{\partial}(f)(x).$$

If I denotes the identity operator, it follows that $(I - e^{\partial})R = a$. We may now use the formal series expansion:

$$\frac{I}{I - e^{\partial}} = -\partial^{-1}\frac{\partial}{e^{\partial} - I} = -\partial^{-1}\left(I + \sum_{k\geq 1}\frac{b_k}{k!}\partial^k\right),$$

where b_k are the Bernoulli numbers. Finally, we obtain what will be called the formal expansion of R:

$$R(x) = -\partial^{-1}a(x) - \sum_{k\geq 1}\frac{b_k}{k!}\partial^{k-1}a(x). \tag{2}$$

Our choice for the bounds of the definite integral in (1) then forces R to satisfy the condition $\int_1^2 R(t)\,dt = 0$. The formal expression in (2) gives us a solution to the difference equation. Observe that in full generality, there can be no uniqueness of solutions since we may add to our solution any periodic non-constant function with mean value 0 over the interval $[\,1, 2\,]$. In order to determine a "principal solution," we need to impose suitable conditions on the function a.

The second approach uses the Laplace transform. It is classically given by the formula

$$\mathcal{L}(g)(x) = \int_0^{+\infty} e^{-x\xi}g(\xi)\,d\xi.$$

The Laplace transform has the following propertie: if $f(x) = \mathcal{L}(g)(x)$, then

$$f(x+1) = \mathcal{L}\big(\xi \mapsto e^{-\xi}g(\xi)\big)(x).$$

Therefore, if the solution R to the difference equation is a Laplace tranform of a function f and a is a Laplace transform of a function b, an expression of R may be obtained, using the inverse transform, by

$$R = \mathcal{L}\left(\xi \mapsto \frac{1}{1 - e^{-\xi}}b(\xi)\right) = \mathcal{L}\left(\xi \mapsto \frac{1}{1 - e^{-\xi}}\mathcal{L}^{-1}(a)(\xi)\right). \tag{3}$$

However, here, this formula cannot be applied in general and needs to be adapted, because of the possible singularity induced by $(1 - e^{-\xi})^{-1}$ at $\xi = 0$ in (3). In the following part, another definition of Laplace transform is thus given, together with its inverse transform (called Borel transform) in a suitable space of function. Such transforms are then used to solve the difference equation, yielding a unique principal solution.

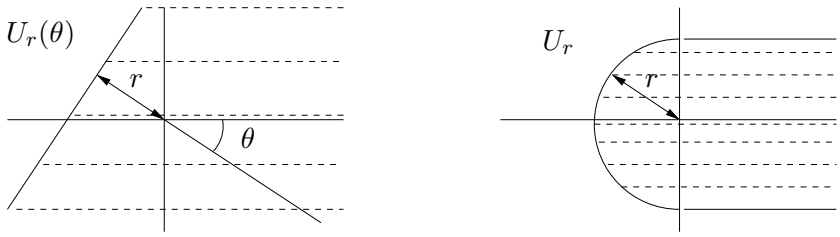
3. **Ramanujan Summation and Borel–Laplace Transform**

Let a be an analytic function over the set P as defined earlier. We will say that a is of exponential type r if for every $\epsilon > 0$, there exists $C > 0$ such that for all x in P, there holds $\big|a(x)\big| \leq Ce^{(r+\epsilon)|x|}$. The Borel transform of a is then defined by

$$\mathcal{B}_d(a)(\xi) = -\frac{1}{2i\pi}\int_d e^{x\xi}a(x)\,dx$$

where d is a half-line in P. It is easy to see that if θ is the angle of d relatively to the real axis, and if a is exponential of type r, then this integral converges for values of x such that $\Re(xe^{i\theta}) < -r$. The Borel transform of a may then be defined in the half-plane $U_r(\theta)$ as in Figure 1. Moreover, if the integral converges for different values of θ, then Cauchy's theorem implies that the integral

does not depend on θ. We may then define the Borel transform of f, which depends only on the origin α of d, in the whole set $U_r = \bigcup_{-\pi/2<\theta<\pi/2} U_r(\theta)$.

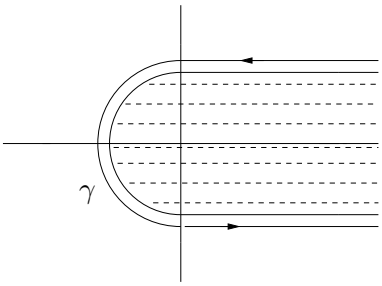


The Borel transform is then of exponential type $k = \Re(\alpha)$. As we can choose α anywhere in the set P, we may take k as small as we want. Furthermore, two Borel transforms of the same function differ by an entire function; by Cauchy's theorem, their difference is the integral of the function $\xi \longmapsto -\frac{1}{2i\pi} e^{x\xi} a(x)$ along any closed path joining the two origins of the contours and is thus analytic.

Let us suppose now that g is an analytic function over the set U_r as introduced before and is of exponential type k. The Laplace transform of g is given by

$$\mathcal{L}(g)(x) = \int_\gamma e^{-x\xi} g(\xi)\, d\xi$$

where γ is the following path:



This formula defines an analytic function over the set $P_k = \{\, x \mid \Re(x) > k \,\}$, which is of exponential type r. If g is entire, then by Cauchy's theorem it follows that $\mathcal{L}(g) = 0$. The Laplace transform of two Borel transforms of a function f is thus the same and, as the Borel transform may be choosen to be of any type $k > 0$, is defined over the set P. Moreover, we have the identity:

$$\forall x \in P \qquad \mathcal{L}\big(\mathcal{B}(a)\big)(x) = a(x).$$

From this we may now ensure the unicity of the function R that is solution of our difference equation. We have the following theorem:

Theorem 1. *Let a be an analytic function over the set P that is of exponential type $\alpha < 2\pi$. The difference equation $R(x) - R(x+1) = a(x)$ admits a unique analytic solution over P that is of exponential type α (denoted as R_a), satisfying $\int_1^2 R_a(t)\, dt = 0$.*

Getting back to the difference equation, we apply the Borel transform to the relation between R and a to get $\mathcal{B}_d(R)(\xi) - e^{-\xi}\mathcal{B}_{d'}(R)(\xi) = \mathcal{B}(a)(\xi)$ where d' is the half-line obtained from d by a translation $z \to z+1$. As mentioned before, we can write

$$\mathcal{B}_d(R)(\xi) - e^{-\xi}\mathcal{B}_{d'}(R)(\xi) = (1-e^{-\xi})\mathcal{B}_d(R)(\xi) + f(x) = a(x)$$

where f is entire. We then apply the Laplace transform to this equality into

$$R(x) = \int_\gamma e^{-x\xi}\frac{1}{1-e^{-\xi}}\mathcal{B}(a)(\xi)\,d\xi - \int_\gamma e^{-x\xi}\frac{1}{1-e^{-\xi}}f(\xi)\,d\xi.$$

As f is entire, the second term of the right member is equal to the residue of $e^{-x\xi}\frac{1}{1-e^{-\xi}}f(\xi)$ in 0 and therefore is a constant equal to $f(1)$. Hence, we have found a solution of the equation that is of the same exponential type as a:

$$R(x) = \int_\gamma e^{-x\xi}\frac{1}{1-e^{-\xi}}\mathcal{B}(a)(\xi)\,d\xi - f(1).$$

The only other exponential solutions of order less than 2π are obtained from this one by adding a constant value, since every entire periodic function of period 1 with an exponential growth less than 2π is a constant. It is easily checked, using Fourier's formulas that if a is entire of exponential type $\alpha < 2\pi$, each of its Fourier coefficients, except the constant one, are zeros.

From the Borel transform properties, it follows that the function $x \longmapsto -\int_\gamma e^{-x\xi}\frac{1}{\xi}\mathcal{B}(a)(\xi)\,d\xi$ is an antiderivative function of a. Hence, the following function is another solution of the difference equation:

$$R(x) = -\int_1^x a(t)\,dt + \int_\gamma e^{-x\xi}\left(\frac{1}{1-e^{-\xi}} - \frac{1}{\xi}\right)\mathcal{B}(a)(\xi)\,d\xi.$$

This last solution does not depend any more on the choice on the Borel transform of a. Furthermore, this function satisfies $\int_1^2 R(t)dt = 0$ and thus is the unique solution of our problem. We can then define the Ramanujan summation of a series of general term $a(n)$ as the following:

$$\sum_{n\geq 1}^{[\mathcal{R}]} a(n) = R(1) = \int_\gamma e^{-\xi}\left(\frac{1}{1-e^{-\xi}} - \frac{1}{\xi}\right)\mathcal{B}(a)(\xi)\,d\xi.$$

It is then easy to see that this sum is a linear functional of a.

4. **Examples and Properties**

For the following functions, we present solutions of the difference equation of exponential order less than 2π, the value of their integral from 1 to 2, and finally the Ramanujan sum of the serie $a(n)$. We will use the Riemann zeta function given for all $x > 0$ and $\Re(z) > 1$ by

$$\zeta(x,z) = \sum_{n=0}^{+\infty}\frac{1}{(n+x)^z}, \qquad \zeta(z) = \zeta(1,z) = \sum_{n\geq 1}\frac{1}{n^z}.$$

$a(x)$	$R(x)$	$\int_1^2 R(t)dt$	$\sum_{n>1}^{[\mathcal{R}]} a(n)$
$\frac{1}{x^z}$	$\zeta(x,z)$	$\frac{1}{z-1}$	$\zeta(z) - \frac{1}{z-1}$
x^k	$-\frac{B_k(x)}{k+1}$	$-\frac{1}{k+1}$	$\frac{1-B_{k+1}}{k+1}$
$\ln x$	$-\ln\Gamma(x)$	$1 - \frac{1}{2}\ln(2\pi)$	$-1 + \frac{1}{2}\ln(2\pi)$
$e^{\alpha x}$	$\frac{e^{\alpha x}}{1-e^{\alpha}}$	$-\frac{e^{\alpha}}{\alpha}$	$e^{\alpha}\left(\frac{1}{1-e^{\alpha}} + \frac{1}{\alpha}\right)$

The first example shows that even if the series is convergent, then we do no have its sum in the usual sense equal to its Ramanujan sum. In fact, we have the following proposition:

Proposition 1. *If $R(x)$ tends to a finite limit when $x \longrightarrow +\infty$, then the series $\sum_{n\geq 1} a(n)$ is convergent, and we have:*

$$\sum_{n\geq 1}^{[\mathcal{R}]} a(n) = \sum_{n=1}^{+\infty} a(n) - \lim_{n\to+\infty} \int_1^n a(t)\,dt.$$

It is thus possible to regard the Ramanujan summation scheme as a convenient renormalisation of the usual summation scheme, where enough terms have been subtracted from the usual sum in order to ensure that the result converges at points where it usually diverges (see the example of the function ζ to the point $z = 1$).

The last example shows another important point. From this last Ramanujan sum, one can compute that

$$\sum_{n\geq 1}^{[\mathcal{R}]} \sin(nt) = \frac{1}{2i}\left(\sum_{n\geq 1}^{[\mathcal{R}]} e^{it} - \sum_{n\geq 1}^{[\mathcal{R}]} e^{-it}\right) = \frac{1}{2}\cot\frac{t}{2} - \frac{\cos t}{t}.$$

Then, if we take $t = \pi$, we get $\sum_{n\geq 1}^{[\mathcal{R}]} \sin(nt) = \frac{1}{\pi}$ whereas $\sum_{n\geq 1}^{[\mathcal{R}]} 0 = 0$. This example shows that the Ramanujan summation depends on the function chosen to represent the sequence we want to sum. In fact, if a and b are two functions that are of exponential type $\alpha < \pi$, if $a(n) = b(n)$ for all $n \geq 1$, then $a = b$, due to a theorem by Carlson [4].

We now have the following properties of Ramanujan summation, considering some classical operations:

Translation. The following holds to compute the sum of the series from the Nth element:

$$\sum_{n\geq 1}^{[\mathcal{R}]} a(n) = a(1) + \cdots + a(N-1) + \sum_{n\geq 0}^{[\mathcal{R}]} a(n+N) - \int_1^N a(t)\,dt.$$

Derivability. Considering the solution R as a function of a, we get

$$R_{\partial^n a} = \partial^n(R_a) + \partial^{n-1}a(1).$$

As an application of this formula, we have the following (the functions R here are defined up to one constant):

$$\begin{aligned} a(x) = \ln x \quad &\Longrightarrow \quad R(x) = \ln\Gamma(x), \\ a(x) = \tfrac{1}{x} \quad &\Longrightarrow \quad R(x) = \tfrac{\Gamma'(x)}{\Gamma(x)}. \end{aligned}$$

As $\int_1^2 \frac{\Gamma'(t)}{\Gamma(t)}\,dt = \ln\Gamma(2) - \ln\Gamma(1) = \ln(1) - \ln(1) = 0$, we have proved that

$$\sum_{n\geq 1}^{[\mathcal{R}]} \frac{1}{n} = \frac{\Gamma'(1)}{\Gamma(1)} = \gamma.$$

Summation by parts. If a and b are both exponential of type less than π:

$$\sum_{n\geq 1}^{[\mathcal{R}]} a(n)\sum_{k=1}^{n} b(k)+\sum_{n\geq 1}^{[\mathcal{R}]} b(n)\sum_{k=1}^{n} a(k) = \sum_{n\geq 1}^{[\mathcal{R}]} a(n)b(n)+\sum_{n\geq 1}^{[\mathcal{R}]} a(n)\sum_{n\geq 1}^{[\mathcal{R}]} b(n)+\int_1^2 R_a(t)R_b(t)\,dt.$$

This formula is in particular interesting when we take $a(x) = 1$ for all x. The formula then gives:

$$\sum_{n\geq 1}^{[\mathcal{R}]}\sum_{k=1}^{n} b(k) = \frac{3}{2}\sum_{n\geq 1}^{[\mathcal{R}]} b(n) - \sum_{n\geq 1}^{[\mathcal{R}]} nb(n) - \sum_{n\geq 1}^{[\mathcal{R}]}\partial^{-1}b(n).$$

with $\partial^{-1}b(n) = \int_1^n b(t)dt$. From this last formula, we compute the following harmonic Ramanujan sum where $H_n = 1+\cdots+\frac{1}{n}$:

$$\sum_{n\geq 1}^{[\mathcal{R}]} H_n = \frac{3}{2}\gamma + \frac{1}{2} - \frac{1}{2}\ln(2\pi)$$

Analytic dependence on a variable.

Proposition 2. *Let D be an open set in $\mathbb{C}$ and let the function $a(z,x)$ be analytic in $D\times P$. Suppose for each compact set K in D, there exists C_K and τ_K such that for all x with $|x|>1$, and all z in K, we have $\big|a(z,x)\big| \leq C_K e^{\tau|x|}$. Then $z\longmapsto \sum_{n\geq 1}^{[\mathcal{R}]} a(z,n)$ is analytic in D and we have:*

$$\partial_z\left(\sum_{n\geq 1}^{[\mathcal{R}]} a(z,n)\right) = \sum_{n\geq 1}^{[\mathcal{R}]}\partial_z a(z,n).$$

It follows from this theorem that the function $z\longmapsto\sum_{n\geq 1}^{[\mathcal{R}]}\frac{1}{n^z}$ is entire. For all z in $\mathbb{C}$, we have:

$$\sum_{n\geq 1}^{[\mathcal{R}]}\frac{1}{n^z} = \zeta(z) - \frac{1}{z-1} \qquad \sum_{n\geq 1}^{[\mathcal{R}]}\frac{(\ln n)^k}{n^z} = (-1)^k\zeta^{(k)}(z) - \frac{(k-1)!}{(z-1)^k}.$$

These formulas remain true when z assumes the limit value 1.

Bibliography

[1] Apostol (Tom M.) and Vu (Thiennu H.). – Dirichlet series related to the Riemann zeta function. *Journal of Number Theory*, vol. 19, n° 1, 1984, pp. 85–102.

[2] Berndt (Bruce C.). – *Ramanujan's notebooks. Part I.* – Springer-Verlag, New York, 1985, x+357p.

[3] Berndt (Bruce C.). – *Ramanujan's notebooks. Part II.* – Springer-Verlag, New York, 1989, xii+359p.

[4] Boas, Jr. (Ralph Philip). – *Entire functions.* – Academic Press, New York, 1954, x+276p.

[5] Borwein (David), Borwein (Jonathan M.), and Girgensohn (Roland). – Explicit evaluation of Euler sums. *Proceedings of the Edinburgh Mathematical Society. Series II*, vol. 38, n° 2, 1995, pp. 277–294.

[6] Cartier (Pierre). – An introduction to zeta functions. In *From number theory to physics (Les Houches, 1989)*, pp. 1–63. – Springer, Berlin, 1992.

[7] Guelfond (A. O.). – *Calcul des différences finies.* – Dunod, Paris, 1963, x+378p.

[8] Hardy (G. H.). – *Divergent series.* – Éditions Jacques Gabay, Sceaux, 1992, xvi+396p. With a preface by J. E. Littlewood and a note by L. S. Bosanquet, Reprint of the revised (1963) edition.

[9] Lewin (Leonard). – *Polylogarithms and associated functions.* – North-Holland Publishing Co., New York, 1981, xvii+359p.

[10] Malgrange (B.). – *Équations différentielles à coefficients polynomiaux.* – Birkhäuser Boston, Boston, MA, 1991, vi+232p.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 89–92.

Available online at the URL
http://algo.inria.fr/seminars/.

Multi-Variable sinc Integrals and the Volumes of Polyhedra

Jonathan M. Borwein
CECM, Simon Fraser University (Canada)

October 22, 2001

Summary by Ludovic Meunier

Abstract

This talk investigates integrals of the form

$$\tau_n := \int_0^\infty \prod_{k=0}^{n} \operatorname{sinc}(a_k x)\, dx$$

and their multi-dimensional analogues. These integrals are related to volumes of polyhedra, which allows to derive various monotony results of such integrals.

1. Introduction and Motivation

A conjecture stated that

$$\mu := \int_0^\infty \prod_{k=1}^{\infty} \cos\left(\frac{x}{k}\right)\, dx < \frac{\pi}{4}. \tag{1}$$

Indeed, $\mu \approx 0.785\underline{38}0$, while $\frac{\pi}{4} \approx 0.785\underline{39}8$ differs in the fifth place. The highly oscillatory integral of an infinite product of cosines (1) is connected to the integrals

$$\tau_n := \int_0^\infty \prod_{k=0}^{n} \operatorname{sinc}(a_k x)\, dx,$$

where $\operatorname{sinc}(\cdot)$ is the *sine cardinal* function,[1] defined by

$$\operatorname{sinc}(x) := \begin{cases} \frac{\sin(x)}{x} & \text{if } x \neq 0, \\ 1 & \text{if } x = 0. \end{cases}$$

Section 2 investigates the behavior of the integrals τ_n as functions of n and exhibits a duality between the τ_n and volume of polyhedra. This duality allows to derive various monotony results for the τ_n and to extend the one-dimensional analysis to the multi-dimensional case, which is sketched in Section 3. Section 4 returns to the integral μ and proves Conjecture (1). Some material contained in this summary is taken from [2].

[1] See, e.g., http://mathworld.wolfram.com/SincFunction.html.

2. **Fourier Transform and** sinc **Integrals**

2.1. **Fourier cosine transform.** This section recalls some standard results about the Fourier cosine transform (FCT) [3, §13].

Definition 1. The FCT of a function $f \in \mathcal{L}^2(-\infty,\infty)$ is defined to be the $\mathcal{L}^2$-limit $\hat{f}$, if it exists, as $y \to \infty$ of the functions

$$c_y(x) := \frac{1}{\sqrt{2\pi}} \int_{-y}^{y} f(t)\cos(xt)\,dt.$$

Property 1. *The function $\hat{f}$ exists, belongs to $\mathcal{L}^2$ and is unique, apart from sets of zero Lebesgue measure.*

Property 2. *If f is continuous over $(-\alpha,\alpha)$ for some $\alpha > 0$ and if $\hat{f} \in \mathcal{L}^1(-\infty,\infty)$ then, conversely, for $t \in (-\alpha,\alpha)$*

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(x)\cos(xt)\,dt = f(t).$$

Property 3 (Convolution)**.** *If $\hat{f}_1$ and $\hat{f}_2$ are the FCTs of* even *functions f_1 and f_2 in $\mathcal{L}^2(-\infty,\infty)$, then $\hat{f}_1\hat{f}_2$ is the FCT of $\frac{1}{\sqrt{2\pi}} f_1 * f_2$, where $*$ denotes the convolution product over $(-\infty,\infty)$.*

Property 4 (Parseval)**.** *With the same notations as in Property 3 and provided that at least one of the functions f_1 or f_2 is real, then*

$$\int_0^{\infty} f_1(t) f_2(t)\,dt = \int_0^{\infty} \hat{f}_1(x)\hat{f}_2(x)\,dx.$$

The function χ_a, for $a > 0$, is defined by

$$\chi_a(x) := \begin{cases} 1 & \text{if } |x| < a \\ \frac{1}{2} & \text{if } |x| = a \\ 0 & \text{if } |x| > a. \end{cases}$$

The FCT of χ_a is $a\sqrt{\frac{2}{\pi}}\,\mathrm{sinc}(ax)$ and, conversely, the FCT of $a\sqrt{\frac{2}{\pi}}\,\mathrm{sinc}(ax)$ is *equivalent* to χ_a. Note that the functions χ_a and sinc are both even and real functions and they both belong to $\mathcal{L}^1(0,\infty)\bigcap\mathcal{L}^2(0,\infty)$, which fulfills the hypotheses of the above properties.

2.2. **Duality.** One first introduces the following notations

$$\sigma_n := \prod_{k=1}^{n} \mathrm{sinc}(a_k x), \qquad s_n := \sum_{k=1}^{n} a_k,$$

$$f_n := \frac{1}{a_n}\sqrt{\frac{\pi}{2}}\,\chi_{a_n}, \qquad F_0 := f_0, \qquad F_n := \left(\sqrt{2\pi}\right)^{1-n} f_1 * f_2 * \cdots * f_n, \text{ for } n \geq 1.$$

By Property 3, one gets that F_n is the FCT of σ_n, and that σ_n is the FCT of F_n. Now, applying Property 4 leads to

$$\tau_n = \int_0^{\infty} F_0(x)F_n(x)\,dx \underset{\text{convolution}}{=} \frac{1}{a_0}\sqrt{\frac{\pi}{2}} \int_0^{\min(s_n,a_0)} F_n(x)\,dx, \tag{2}$$

provided that $\tau_0 = \pi(2a_0)^{-1}$, which is a standard result [1, p. 314].

Consider the hyper-cube H_n and the polyhedron P_n defined by

$$H_n := \big\{\, (x_1,\dots,x_n) \;\big|\; |x_k| \le 1,\ k \in [\,1,n\,] \,\big\},$$
$$P_n := \left\{\, (x_1,\dots,x_n) \;\middle|\; \left|\sum_{k=1}^{n} a_k x_k\right| \le a_0,\ |x_k| \le 1, k \in [\,1,n\,] \,\right\},$$

then (2) reads

$$\tau_n = \frac{\pi}{a_0}\frac{1}{2^n a_1 \dots a_n}\int_0^{\min(s_n,a_0)} \chi_{a_1}(x) * \cdots * \chi_{a_n}(x)\,dx = \frac{\pi}{2a_0}\frac{\mathrm{Vol}(P_n)}{\mathrm{Vol}(H_n)}, \tag{3}$$

where $\mathrm{Vol}(\cdot)$ denotes the volume. Equation (3) expresses a *duality* between the integrals τ_n and the volumes of polyhedra. This duality is used to prove the following theorem.

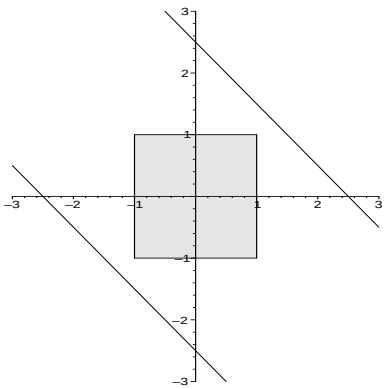
Theorem 1 (Monotony)**.** *For $a_k \ge 0$, then*

$$0 < \tau_n \le \frac{1}{a_0}\frac{\pi}{2} \qquad \textit{with equality if } a_0 \ge s_n,$$
$$0 < \tau_{n+1} \le \tau_n < \frac{1}{a_0}\frac{\pi}{2} \qquad \textit{provided that } a_{n+1} \le a_0 < s_n.$$

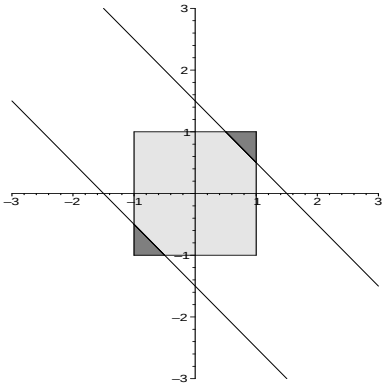
2.3. **Some puzzling integrals.** Consider the family τ_n, where $a_k = \frac{1}{2k+1}$. For $k \in [\,0,6\,]$, $\tau_k = \frac{\pi}{2}$. However,

$$\tau_7 = \frac{467807924713440738696537864469}{935615849440640907310521750000}\pi \approx 0.499999999992646\pi.$$

According to Theorem 1, this result is explained by the fact that the value of the integrals τ_n drops when the constraint $\sum_{k=1}^{n} a_k x_k \le a_0$ *bites* into the hyper-cube H_n. Indeed, $\sum_{k=1}^{6} a_k < 1$, but on the addition of the seventh term, the sum exceeds 1 and the identity $\tau_k = \frac{\pi}{2}$ no longer holds. This behavior is illustrated in the case of dimension 2 by the following diagrams.



Volume = □



Volume = □ − 2△

3. **Multi-Dimensional** sinc **Integrals**

Let $a := (a_1,\dots,a_m)$ and $y := (y_1,\dots,y_m)$ in $\mathbb{R}^m$. Define $ay := \sum_{k=1}^{m} a_k y_k$ and δ_a the Lebesgue measure restricted to $\{\, x \in \mathbb{R}^m \mid x = ta,\ -1 \le t \le 1 \,\}$. For any integrable function f over $\mathbb{R}^m$, $\int_{\mathbb{R}^m} f(x)\delta_a(dx) = \int_{-1}^{1} f(ta)\,dt$ and thus

$$\int_{\mathbb{R}^m} e^{ixy}\delta_a(dx) = 2\,\mathrm{sinc}(ay). \tag{4}$$

More generally, with $s_1, \ldots, s_n \in \mathbb{R}^m$ and the convolution measure $\lambda = \delta_{s_1} * \cdots * \delta_{s_n}$, Equation (4) becomes

$$F(y) := \int_{\mathbb{R}^m} e^{ixy} \lambda(dx) = 2^n \prod_{k=1}^{n} \operatorname{sinc}(s_k y).$$

Another version of Parseval's theorem yields the following theorem.

Theorem 2. *With the same notations as above and with $n \geq m$ and the constraint that the $m \times m$ matrix $(s_1, \ldots, s_m)$ is non-singular, then*

$$\int_{\mathbb{R}^m} F(y) \prod_{k=1}^{m} \operatorname{sinc}(y_k)\, dy = \frac{\pi^m}{2^n} \int_{[-1,1]^m} \lambda(dy).$$

This theorem relates the volume of a polyhedra of dimension n with a m-dimensional sinc integral.

4. The Cosine Integrals Revisited

Invoking the factor theorem of Weierstrass [4, p. 137], one gets

$$\operatorname{sinc}(x) = \prod_{k=1}^{\infty} \left(1 - \frac{x^2}{\pi^2 k^2}\right) \text{ and } \cos(x) = \prod_{l=0}^{\infty} \left(1 - \frac{4x^2}{\pi^2 (2l+1)^2}\right).$$

If one lets $C(x) = \prod_{k=1}^{\infty} \cos\left(\frac{x}{n}\right)$, it follows that $C(x) = \prod_{k=0}^{\infty} \operatorname{sinc}\left(\frac{2x}{2k+1}\right)$. By Theorem 1, where $a_k = \frac{2}{2k-1}$, one obtains

$$0 < \mu = \int_0^{\infty} C(x)\, dx = \lim_{n \to \infty} \int_0^{\infty} \prod_{k=1}^{n} \operatorname{sinc}(a_k x)\, dx < \frac{\pi}{4},$$

which proves the conjecture stated in Equation (1).

Bibliography

[1] Apostol (Tom M.). – *Mathematical analysis*. – Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1974, second edition, xvii+492p.

[2] Borwein (David) and Borwein (Jonathan M.). – Some remarkable properties of sinc and related integrals. *The Ramanujan Journal*, vol. 5, n° 1, 2001, pp. 73–89.

[3] Titchmarsh (A.C.). – *The theory of functions*. – Oxford University Press, 1939, second edition.

[4] Whittaker (E. T.) and Whatson (G. N.). – *A course of modern analysis*. – Cambridge University Press, 1927, fourth edition.

Part V

Number Theory

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), p. 95.

Available online at the URL
http://algo.inria.fr/seminars/.

L-Series of Squares of Squares

Jonathan Borwein
Department of Mathematics and Statistics, SFU (Canada)

October 22, 2001

The book by Hardy and Wright records elegant closed forms for the Dirichlet generating functions of the divisor functions $\sigma_k(n) = \sum_{d|n} d^k$ and $\sigma_k^2(n)$ in terms of the Riemann zeta function $\zeta(s)$. This talk extends such evaluations by providing a general identity for Dirichlet convolutions of completely multiplicative sequences. If f_1, f_2, g_1, and g_2 are completely multiplicative, then the formula

$$\sum_{n=1}^{\infty}(f_1 * g_1)(n) \times (f_2 * g_2)(n) \times n^{-s} = L_{f_1 f_2 g_1 g_2}(2s)^{-1} L_{f_1 f_2}(s) L_{g_1 g_2}(s) L_{f_1 g_2}(s) L_{g_1 f_2}(s)$$

holds, where for a sequence f,

$$L_f(s) = \sum_{n=1}^{\infty} f(n) n^{-s}.$$

Applications are given to the number of representations of integers as sums of squares. Let $r_N(n)$ be the number of integer solutions of $x_1^2 + \cdots + x_N^2 = n$ and $r_{2,P}(n)$ be the number of integer solutions of $x^2 + Py^2 = n$. Closed forms in terms of $\zeta(s)$ and Dirichlet L-functions are obtained for the generating functions of $r_N(n)$, $r_N(n)^2$, $r_{2,P}(n)$, and $r_{2,P}(n)^2$ and certain N and P.

The talk is based on joint work with Stephen Choi. See CECM Preprint 01:167, which can be obtained at http://www.cecm.sfu.ca/preprints.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 97–100.

Available online at the URL
http://algo.inria.fr/seminars/.

Irrationality of the ζ Function on Odd Integers

Tanguy Rivoal
Institut de Mathématiques de Jussieu (France)

February 5, 2001

Summary by Marianne Durand

Abstract

The ζ function is defined by $\zeta(s) = \sum_n 1/n^s$. This talk is a study of the irrationality of the zeta function on odd integer values > 2.

1. Introduction

The sum $\sum_n 1/n^2$ was first studied by Bernoulli, who proved around 1680 that it converged to a finite limit less than 2. Euler proved in 1735 that it is equal to $\pi^2/6$, and studied the more general function $\zeta(s) = \sum_n 1/n^s$. He also showed that on even integers the ζ function has a closed form, namely $\zeta(2n) = C_n \pi^{2n}$ where the coefficients C_n are rational numbers that he wrote in terms of Bernoulli numbers. A century later Riemann studied this function on the whole complex plane, and he stated a conjecture on the location of the zeroes of the zeta function, that is known as the Riemann hypothesis, and is still unproved.

The first result on the irrationality of the ζ function on odd integers is due to Apéry, who proved in 1978 that $\zeta(3)$ is irrational [1]. Recently Tanguy Rivoal showed that the ζ function takes infinitely many irrational values on the odd integers [4, 5], and that there exists an odd integer j with $5 \leq j \leq 21$ such that $\zeta(j)$ is irrational [5]. Zudilin [6] refined this result and proved it for $5 \leq j \leq 11$.

2. Irrationality of $\zeta(3)$

Theorem 1 (Apéry(1978))**.** *The number $\zeta(3)$ is irrational.*

The following proof is due to Nesterenko [3], after ideas by Beukers. The theorem is proved using the following generating function

$$S_n(z) = \sum_{k=1}^{\infty} \frac{\partial}{\partial k} \left(\frac{(k-1)^2(k-2)^2 \dots (k-n)^2}{k^2(k+1)^2 \dots (k+n)^2} \right) z^{-k}$$

The decomposition of the coefficient of z^{-k} in partial fractions gives the equality

$$S_n(z) = P_{0,n}(z) + P_{1,n}(z)\,\mathrm{Li}_2(1/z) + P_{2,n}(z)\,\mathrm{Li}_3(1/z) \tag{1}$$

where $\mathrm{Li}_s(z) = \sum_{n \geq 1} \frac{z^n}{n^s}$ is a polylogarithm function, and $P_{k,n}$ are polynomials of degree n such that $P_{1,n}(1) = 0$. When Equation (1) is specialized at $z = 1$, it becomes

$$S_n(1) = P_{0,n}(1) + P_{2,n}(1)\zeta(3),$$

with the additionnal properties that $P_{2,n}(1) \in \mathbb{Z}$ and $d_n^3 P_{0,n}(1) \in \mathbb{Z}$ where $d_n = \mathrm{ppcm}(1, 2, \dots, n)$.

The value $S_n(1)$ is bounded by using an integral representation.

$$S_n(1) = \frac{1}{2i\pi}\int_L \left(\frac{\Gamma(n+1-s)\Gamma(s)^2}{\Gamma(n+1+s)}\right)^2 ds, \tag{2}$$

where L is the vertical line $\Re(z) = c,\ \ 0 < c < n+1$, oriented from top to bottom. From this integral, the bounds $0 < S_n(1) \leq c(\sqrt{2}-1)^{4n}$ are obtained.

The inequalities $0 < d_n^3 P_{0,n}(1) + d_n^3 P_{2,n}(1)\zeta(3) < cr^n$, where c is a constant, and $r < 1$ prove that $\zeta(3)$ is irrational; because if $\zeta(3)$ is rational and equal to p/q, then $qd_n^3 P_{0,n}(1) + qd_n^3 P_{2,n}(1)\zeta(3)$ is an integer greater than 0 and bounded by qcr^n that converges to 0.

3. The ζ Function Has Infinitely Many Irrational Values on Odd Integers

Tanguy Rivoal in fact proved a stronger result, that is:

Theorem 2. *Let a be an odd integer greater than 3 and $\delta(a)$ be the dimension of the $\mathbb{Q}$-vector space spanned by 1, $\zeta(3)$, ..., $\zeta(a)$, then*

$$\delta(a) \geq \frac{1}{3}\log a.$$

This implies directly that infinitely many $\zeta(2n+1)$ are irrational.

To prove Theorem 2, we introduce the series

$$S_{n,a,r}(z) = n!^{a-2r}\sum_{k=1}^{\infty}\frac{(k-rn)_{rn}(k+n+1)_{rn}}{(k)_{n+1}^a}z^{-k},$$

where $(k)_n = k(k+1)\dots(k+n-1)$ is the Pochhammer symbol, and n, r, and a are integers satisfying $n \geq 0,\ \ 1 \leq r < a/2$, so that $S_{n,a,r}(z)$ exists when $|z| \geq 1$. As for the proof of the irrationality of $\zeta(3)$, an equality between the series studied and values of ζ is found, namely

$$S_{n,a,r}(1) = P_{0,n}(1) + \sum_{l=2}^{a} P_{l,n}(1)\zeta(l),$$

moreover, if $(n+1)a + l$ is odd then $P_{l,n}(1) = 0$. For n odd and a odd greater than 3, $P_{l,n}(1) = 0$ if l is even, so that $S_{n,a,r}(1)$ is a linear combination of values of ζ on odd integers.

The dimension of the vector space spanned by 1, $\zeta(3)$, ..., $\zeta(a)$ is based on the following theorem:

Theorem 3 (Nesterenko's criterion)**.** *Let θ_1, θ_2, ..., θ_N be N real numbers, and suppose that there exist N sequences $(p_{l,n})_{n\geq 0}$ such that*

1. $\forall\ i = 1, \dots, N,\ \ p_{l,n} \in \mathbb{Z}$;
2. $\alpha_1^{n+o(n)} \leq \left|\sum_{l=1}^{N} p_{l,n}\theta_l\right| \leq \alpha_2^{n+o(n)}$, *with* $0 < \alpha_1 \leq \alpha_2 < 1$;
3. $\forall l = 1, \dots, N,\ \ |p_{l,n}| \leq \beta^{n+o(n)}$ *with* $\beta > 1$.

Then

$$\dim_{\mathbb{Q}}(\mathbb{Q}\,\theta_1 + \mathbb{Q}\,\theta_2 + \dots + \mathbb{Q}\,\theta_N) \geq \frac{\log(\beta) - \log(\alpha_1)}{\log(\beta) - \log(\alpha_1) + \log(\alpha_2)}.$$

This criterion, applied to the real numbers $\theta_i = \zeta(2i+1),\ \ i \leq (a-1)/2$, with the sequences $p_{l,n}$ defined by $p_{0,n} = d_{2n}^a P_{0,2n}(1)$ and $p_{l,n} = d_{2n}^a P_{2l+1,2n}(1)$ if $1 \leq l \leq (a-1)/2$ yields the inequality

$$\delta(a) \geq \frac{\log(r) + \frac{a-r}{a+1}\log(2)}{1 + \log(2) + \frac{2r+1}{a+1}\log(r+1)}, \tag{3}$$

for all $1 \leq r < a/2$.

For each odd integer $a > 1$, there exists an r (that can be made explicit) such that the inequality (3) reduces to $\delta(a) \geq \log(a)/3$.

The proof of this property can be adapted to show that $\delta(169) > 2$, which means that there exists an integer j, $5 \leq j \leq 169$, such that 1, $\zeta(3)$, and $\zeta(j)$ are linearly independent over $\mathbb{Q}$.

4. At Least One Number Amongst $\zeta(5)$, $\zeta(7)$, ..., $\zeta(21)$ Is Irrational

The linear independence of $1, \zeta(3), \zeta(j)$ for some $j \leq 169$ implies the irrationality of $\zeta(j)$, but is stronger. The bound 169 is improved in this section by only seeking the irrationality.

Theorem 4. *There exists an integer j, $5 \leq j \leq 21$, such that $\zeta(j)$ is irrational.*

The proof of this theorem follows the same directions as the two previous ones. First an adequate generating function $S_n(z)$ is considered, that gives a linear equation implying the zeta function on odd integers when specialized. The coefficients of this equation are studied, and their denominator bounded; a saddle-point method gives asymptotic results on $S_n(1)$. These lemmas, combined with the Nesterenko criterion finally give the result.

The generating function $S_n(z)$ is

$$S_n(z) = n!^{a-6} \sum_{k=1}^{\infty} \frac{1}{2} \frac{d^2}{dk^2} \left(\left(k + \frac{n}{2}\right) \frac{(k-n)_n^3 (k+n+1)_n^3}{(k)_{n+1}^a} \right) z^{-k},$$

where a is an integer ≥ 6. This sum is convergent when $|z| \geq 1$. This sum is expanded in simple elements, and then specialized at $z = 1$ to give a relation between values of ζ on odd integers, $\zeta(3)$ excluded, namely

$$S_n(1) = P_{0,n}(1) + \sum_{j=2}^{a/2} j(2j-1) P_{2j-1,n}(1) \zeta(2j+1). \tag{4}$$

The coefficients $P_{l,n}$ satisfy $2d_n^{a+2} P_{0,n}(1) \in \mathbb{Z}$ and $2d_n^{a-l} P_{l,n}(1) \in \mathbb{Z}$ for $1 \leq l \leq a$.

The next step of the proof is to get an asymptotic result on $S_n(1)$, using a saddle-point method. We do not know of any integral representation similar to (2) for $S_n(1)$, but we can express $S_n(1)$ as the real part of a complex integral. First we introduce $R_n(k)$,

$$R_n(k) = n!^{a-6} \left(k + \frac{n}{2}\right) \frac{(k-n)_n^3 (k+n+1)_n^3}{(k)_{n+1}^a}.$$

So that $S_n(z) = \sum_{k=1}^{\infty} \frac{1}{2} \frac{d^2}{dk^2} R_n(k) z^{-k}$. We also define

$$J_n(u) = \frac{n}{2i\pi} \int_L R_n(nz) \left(\frac{\pi}{\sin(n\pi z)} \right)^3 e^{nuz} \, dz,$$

where L is a vertical line from $i\infty$ to $-i\infty$ with a real part between 0 and 1. With those notations, the property $S_n(1) = \Re\big(J_n(i\pi)\big)$ holds.

The quantity $J_n(i\pi)$ is rewritten in terms of the Γ function, using the complement formula $\Gamma(t)\Gamma(1-t) = \pi/\sin(\pi t)$, and is then approximated using the Stirling formula. This gives

$$J_n(i\pi) = \left(i(-1)^{n+1} (2\pi)^{a/2-1} n^{2-a/2} \int_L g(z) e^{nw(z)} \, dz \right) \big(1 + O(1/n)\big),$$

where $g(z) = (z+1/2) \frac{\sqrt{1-z}^3 \sqrt{2+z}^3}{\sqrt{z}^{a+3} \sqrt{z+1}^{a+3}}$ and $w(z) = (a+3)z\log(z) - (a+3)(z+1)\log(z+1) + 3(1-z)\log(1-z) + 3(z+2)\log(z+2) + i\pi z$. The variable a is now specialized to 20 in order to have a relation between $\zeta(5)$, ..., $\zeta(21)$. The saddle-point method, see [2, pp. 279–285], now

applies to the point z_0, the only root of $w'(z) = 0$ such that $0 < \Re(z) < 1$. The numerical value of z_0 is $0.992 - 0.012i$. The estimation of $J_n(i\pi)$ obtained is

$$J_n(i\pi) = u_n r(-1)^{n+1} n^{-8} e^{nw(z_0)+i\beta},$$

with r and β real constants and u_n a sequence of complex numbers converging to 1. We define $v_0 = \Im\big(w(z_0)\big)$. The real part of this expression is

$$r(-1)^{n+1} n^{-8} e^{\Re\left(nw(z_0)\right)} \big(\Re(u_n)\cos(nv_0+\beta) - \Im(u_n)\sin(nv_0+\beta)\big).$$

Since $v_0 \sim 3.104$ is not a multiple of π, there exists an increasing sequence $\phi(n)$ such that $\cos\big(\phi(n)v_0+\beta\big)$ tends to a limit $l \neq 0$. As a direct consequence

$$\lim_{n\to\infty} \Re J_{\phi(n)}(i\pi) = K(-1)^{\phi(n)+1}\phi(n)^{-8} e^{\Re\left(\phi(n)w(z_0)\right)},$$

where K is a constant. So $\lim_{n\to\infty}\big|S_{\phi(n)}(1)\big|^{1/\phi(n)} = e^{\Re\left(w(z_0)\right)}$.

This result, combined with Equation (4) proves Theorem 4 as follows. Equation (4) tells that $l_n = 2d_n^{22} S_n(1)$ is a linear combination of $\zeta(5), \ldots, \zeta(21)$ with integer coefficients. The paragraph above shows that l_n satisfies $\lim_{n\to\infty}\big|l_{\phi(n)}\big|^{1/\phi(n)} \in (0,1)$. So one of the values $\zeta(5), \ldots, \zeta(21)$ is irrational.

This result has been refined by Zudilin [6], who proved that at least one of the four numbers $\zeta(5)$, $\zeta(7)$, $\zeta(9)$, and $\zeta(11)$ is irrational, by using a general hypergeometric construction of linear forms in odd zeta values.

Bibliography

[1] Apéry (Roger). – Irrationalité de $\zeta(2)$ et $\zeta(3)$. *Astérisque*, vol. 61, 1979, pp. 11–13.

[2] Dieudonné (Jean). – *Calcul infinitésimal*. – Hermann, Paris, 1980. 479 pages.

[3] Nesterenko (Yu. V.). – Some remarks on $\zeta(3)$. *Mathematical Notes*, vol. 59, n° 5-6, 1996, pp. 625–636.

[4] Rivoal (Tanguy). – La fonction zêta de Riemann prend une infinité de valeurs irrationnelles aux entiers impairs. *Comptes Rendus de l'Académie des Sciences. Série I*, vol. 331, n° 4, 2000, pp. 267–270.

[5] Rivoal (Tanguy). – *Structures discrètes et analyse diophantienne*. – Thèse, Université de Caen, June 2001.

[6] Zudilin (V. V.). – One of the numbers $\zeta(5)$, $\zeta(7)$, $\zeta(9)$, $\zeta(11)$ is irrational. *Russian Mathematical Surveys*, vol. 56, n° 4, 2001, pp. 774–776.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 101–104.

Available online at the URL
http://algo.inria.fr/seminars/.

Irrationality Measures of $\log 2$ and $\pi/\sqrt{3}$

Nicolas Brisebarre

Université de St-Étienne

January 14, 2002

Summary by Bruno Salvy

Abstract

1. Irrationality Measures

An *irrationality measure* of $x \in \mathbb{R} \setminus \mathbb{Q}$ is a number μ such that

$$\forall \epsilon > 0, \exists C > 0, \forall (p,q) \in \mathbb{Z}^2, \left| x - \frac{p}{q} \right| \geq \frac{C}{q^{\mu+\epsilon}}.$$

This is a way to measure how well the number x can be approximated by rational numbers. The measure is *effective* when $C(\epsilon)$ is known. We denote $\inf \{ \mu \mid \mu \text{ is an irrationality measure of } x \}$ by $\mu(x)$, and we call it *the* irrationality measure of x.

By definition, rational numbers do not have an irrationality measure. Given two irrationality measures for a number, the smaller one is more precise, since it shows the number to be further "away" from rational numbers. For all $x \in \mathbb{R} \setminus \mathbb{Q}$, the inequality $\mu(x) \geq 2$ holds and gives the minimal possible value. This inequality follows from a pigeon-hole principle: for any integer $n > 1$, the fractional parts $\{qx\}$, $0 \leq q < n$ together with the number 1, are $n+1$ real numbers in the interval $[0,1]$; therefore two of them must be at distance less than or equal to $1/n$; their difference is of the form $qx - p$, so that $|x - p/q| < 1/nq < 1/q^2$. A more explicit construction of these rational approximations is given by continued fraction expansions. The periodicity of continued fraction expansions of irrational quadratic numbers implies that they have an (effective) measure equal to 2. This result was generalized by Liouville in 1844, when he obtained the first practical criterion for constructing transcendental numbers.

Theorem 1 (Liouville)**.** *An algebraic number α of degree n has effective irrationality measure n.*

Proof. Let P be the minimal polynomial of α. This is a polynomial of degree n with integer coefficients. By the mean value theorem,

$$P(\alpha) - P(p/q) = -P(p/q) = (\alpha - p/q)P'(\xi),$$

for some ξ between α and p/q. Since P is irreducible, $P(p/q) \neq 0$ and $\left|q^n P(p/q)\right|$ is an integer which is therefore at least 1. It is sufficient to restrict attention to p/q at distance less than 1 from α. Then $P'(\xi)$ has a lower bound on this interval and this proves the measure. The bound is made effective in terms of the *height* H of P (the largest absolute value of its coefficients), as $\left|P'(\xi)\right| < n^2 H \big(1 + |\alpha|\big)^{n-1}$. □

Number	$\log 2$	π	$\pi/\sqrt{3}$	$\zeta(2)$	$\zeta(3)$
measure	3.8913997	8.016045	4.601579	5.441243	5.513891
author	Rukhadze (1987)	Hata (1993)		Rhin & Viola (1996)	

TABLE 1. Irrationality measures and their authors.

Using this result, Liouville constructed so-called *Liouville numbers* whose smallest measure is infinite. These numbers are therefore transcendental, since their measure cannot be bounded by any integer as demanded by the above theorem. A family of such numbers is

$$\sum_{n\geq 0} a^{-n!}, \qquad a \in \mathbb{N} \setminus \{0,1\}.$$

Indeed, truncating after the kth term gives a rational approximation p_k/q_k with $q_k = a^{k!}$ and a simple computation on the tail of the series shows that it is less than q_k^{-k}.

In the twentieth century, a sequence of results improved on Liouville's theorem, this was ended by Roth, who showed in 1955 that all algebraic numbers have irrationality measure exactly 2 (this result is not effective). In a different direction, Khintchine showed that almost all real numbers (in the sense of Lebesgue) have irrationality measure 2. However, not all reals have measure 2: apart from Liouville numbers, for every $\mu \in [2, \infty)$ the following gives a family of numbers with measure exactly μ:

$$[a] + \cfrac{1}{[a^b] + \cfrac{1}{[a^{b^2}] + \cfrac{1}{[a^{b^3}] + \dots}}}, \quad a > 1,\ b = \mu - 1,$$

where $[a]$ denotes the integer part of a.

2. Padé–Hermite Approximants

Very few actual values of the irrationality measure are known. Techniques have been developed to derive upper bounds for given numbers. A summary of the current best known upper bounds for a few constants is given in Table 1. Note that in each case, the mere existence of a bound is a proof of irrationality.

The basis for several of these bounds lies in sequences of approximants of the form

$$q_n x - p_n = \epsilon_n, \tag{1}$$

where p_n and q_n are integers. Then if q_n does not grow too fast with n, while ϵ_n tends to 0 fast enough, an effective irrationality measure can be found. More precisely, several lemmas of the following type are available.

Lemma 1 (G. V. Chudnovsky)**.** *If there exist positive real numbers* σ *and* τ *such that*

$$\limsup_{n\to\infty} \frac{\log q_n}{n} \leq \sigma, \qquad \lim_{n\to\infty} \frac{\log|\epsilon_n|}{n} = -\tau,$$

then $\mu = 1 + \sigma/\tau$ *is an effective measure of irrationality for* x*.*

An important tool to obtain approximants of type (1) is the use of more general *Padé–Hermite approximants*. (See the summary of Rivoal's talk in these proceedings for a similar use in transcendence theory.) In the case of $\log 2$ and $\pi/\sqrt{3}$, the approximants that will be considered are of the

form

$$Q_n(z)\log(1-z) - P_n(z) = E_n(z),$$

where Q_n and P_n are polynomials while E_n is an analytic function. Setting $z=-1$ in this equation gives a relation involving $\log 2$, while setting $z = \exp(i\pi/3) = 1-\exp(-i\pi/3) = 1+i\sqrt{3}/2$ gives a relation involving π and $\sqrt{3}$.

If Q is a polynomial in $\mathbb{Z}_n[t]$ (polynomials with integer coefficients and degree at most n), then

$$I(z) = \int_0^1 \frac{Q(t)}{1-zt}\,dt = -\frac{Q(1/z)}{z}\log(1-z) + P(1/z)/d_n,$$

where $P(t) \in \mathbb{Z}_n[t]$ and $d_n = \operatorname{lcm}(1,2,\ldots,n)$. Now, the idea is to look for "good" families of polynomials Q_n in order to reach both a small σ and a large τ in the lemma.

In 1980, Alladi and Robinson [1] used $Q_n(z) = \left(z^n(1-z)^n\right)^{(n)}/n!$ (these are related to the Legendre polynomials). It is easily seen that $Q_n(z) \in \mathbb{Z}_n[z]$ with coefficients

$$\frac{(n+k)!}{k!^2(n-k)!}, \qquad k=0,\ldots,n$$

whose absolute value is asymptotically of order $\left(3+2\sqrt{2}\right)^n$ (the maximal coefficient is reached for $k \sim n/\sqrt{2}$). By repeated integration by parts one gets

$$I_n(z) = (-z)^n \int_0^1 \frac{t^n(1-t)^n}{(1-zt)^{n+1}}\,dt.$$

Now, for $z=-1$, the integral is easily bounded by considering the maximum of $t(1-t)/(1+t)$ in the interval $[\,0,1\,]$, which gives $\left(3-2\sqrt{2}\right)^n$. Finally, it is a classical result from number theory that $d_n \simeq e^n$. Putting all this together gives

$$\mu(\log 2) \le 1 - \frac{1+\log\left(3+2\sqrt{2}\right)}{1+\log\left(3-2\sqrt{2}\right)} \approx 4.622.$$

Similarly, they get $\mu\left(\pi/\sqrt{3}\right) \le 8.310$.

3. **Better Polynomials**

In 1987, G. Rhin [4] replaced the polynomials $t^n(1-t)^n$ in the integral I_n by polynomials with integer coefficients but giving the integrand a lower upper bound. Using the factors

$$X, 1-6X+X^2, 1-6X, 1-5X, 2-11X, 1-7X+2X^2$$

with linear exponents that are computed by an optimization process, he obtains $\mu(\log 2) \le 4.0765$ and $\mu(\pi/\sqrt{3}) \le 4.97$.

The following family of polynomials was considered by N. Brisebarre [3]. It generalizes the polynomials of Alladin & Robinson, but also more general families that had been used by M. Hata, E. A. Rukhadze and A. Dubitskas as well as D. V. and G. V. Chudnovsky to obtain the bounds for $\log 2$ and $\pi/\sqrt{3}$ in Table 1.

$$Q_{n,m,m'} = \frac{\left(z^{n+m'}(1-z)^{n+m}\right)^{(n+m+m')}}{(n+m+m')!} = \sum_{j=0}^{n} (-1)^{m+j}\binom{n+m}{m+j}\binom{n+m+m'+j}{n+m+m'}z^j.$$

One-parameter families are obtained by considering $Q_{an,bn,cn}$, with a,b,c integers, a being restricted to be positive. As shown by the formula above, these polynomials have integer coefficients. Moreover, it turns out that the *content* of these polynomials (the gcd of their coefficients) is quite large and can be exploited to some extent.

Proposition 1. *Let* c_n *be the content of* $Q_{an,bn,cn}$ *when* $a > -\min(b, c, b+c, 0)$*, then*

$$e(b/a, c/a) := \lim_{n\to\infty} \frac{\log c_n}{n} = \int_{E_{a,b,c}} \frac{dx}{x^2},$$

where

$$E = \left\{ x \;\middle|\; x > 0,\ 0 < \{x\} + \left\{\frac{c}{a}x\right\} - 1 < 1 - \left\{\frac{b}{a}x\right\} < \{x\} \right\}.$$

The proof of this lemma consists in exhibiting sufficiently many intervals containing prime divisors of each of the coefficients of the polynomial, see [3]. The computation of the integral starts by slicing the interval $[\,0, 1\,]$ in a finite number of subintervals, bounded by the rationals j/a, $j/|b|$, $j/|c|$, for $j \in \mathbb{N}$. On each subinterval, the value of the fractional parts in the definition of E are then studied in more detail, which leads to a more or less explicit formula for $e(b/a, c/a)$. For specific values of b and c, the formula becomes completely explicit, and for instance one recovers a few special cases due to Hata, like

$$e(a^{-1}, a^{-1}) = \log\left(\frac{a+1}{(a+2)^{(a+2)/(2a+2)}\, a^{a/(2a+2)}}\right) + \frac{\pi}{2a+2}\big(\chi(a+2) - \chi(a)\big), \quad \chi(a) := \sum_{r=1}^{[a/2]} \cot(r\pi/a).$$

As before, the next steps consist in bounding the coefficients of $Q_{an,bn,cn}$ and the maximum of $Q_{a,b,c}(t)/(1+t)$ in the interval $[\,0, 1\,]$ so as to get an irrationality measure. These are achieved without too much difficulty. The final result is in terms of $e(b/a, c/a)$ and one is left with an optimization problem in $\mathbb{R}^2$. Experiments show that the optimal result is reached at several values of (a, b, c), namely $(8, -1, -1)$, $(7, 1, -1)$, $(6, 1, 1)$, and $(7, -1, 1)$. The corresponding polynomials have been considered by Hata and Rukhadze, they lead to the bound from Table 1. Similar considerations apply for $\pi/\sqrt{3}$, see [3].

Bibliography

[1] Alladi (K.) and Robinson (M. L.). – Legendre polynomials and irrationality. *Journal für die Reine und Angewandte Mathematik*, vol. 318, 1980, pp. 137–155.

[2] Baker (Alan). – *Transcendental number theory*. – Cambridge University Press, Cambridge, 1990, second edition, *Cambridge Mathematical Library*, x+165p.

[3] Brisebarre (Nicolas). – Irrationality measures of $\log 2$ and $\pi/\sqrt{3}$. *Experimental Mathematics*, vol. 10, n° 1, 2001, pp. 35–52.

[4] Rhin (Georges). – Approximants de Padé et mesures effectives d'irrationalité. In *Séminaire de Théorie des Nombres, Paris 1985–86*, pp. 155–164. – Birkhäuser Boston, Boston, MA, 1987.

Part VI

Miscellany

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 107–112.

Available online at the URL
http://algo.inria.fr/seminars/.

Approximate Matching of Secondary Structures

Matthieu Raffinot

Génopôle, Université d'Évry (France)

February 25, 2002

Summary by Pierre Nicodème

Abstract

This talk presents an algorithm to search for all approximate matches of a helix in a genome, where a helix is a combination of sequence and folding constraints. It is a joint work with Nadia El-Mabrouk of University of Montréal and was presented at the RECOMB 2002 congress [1]. The method applies for more general secondary RNA structures including several helices.

1. Introduction

We give in this section an intuitive description of the problem considered and of the method used. We refer to the next section for more precise definitions. We consider the alphabet $\Sigma = \{A, C, G, T\}$ of DNA. RNA molecules are subject to Watson–Cricks base-pairings constraints, where the pairs are $A \leftrightarrow T$ and $C \leftrightarrow G$. A *network expression* over $\Sigma^\star$ is a regular expression built with the union and concatenation operators. The *complement* $\overline{w}$ of a word w is obtained by reversing the order of the letters of a word and by taking the pairing letter for each letter. For instance,

$$\text{complement}(AAGT) = \overline{AAGT} = ACTT.$$

The complement $\overline{E}$ of a network expression E is the set of complements of the words of the language defined by E. A secondary expression S is of the form

$$S = N_1 E_1 N_2 E_2 N_3 \dots N_3' \overline{E_2} N_2' \overline{E_1} N_1',$$

where the N_i, N_i', and E_i are network expressions. The $E_1, E_2, \dots$ (resp. $\overline{E_1}, \overline{E_2}, \dots$) are marked *sl* (resp. *sr*) for left (resp. right) strands. Figure 1 represents an example of secondary structure,

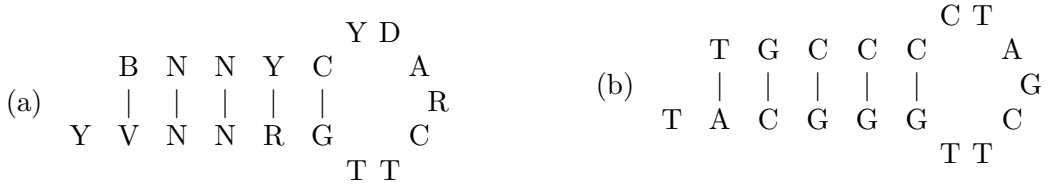


FIGURE 1. (a) A secondary expression S representing a signature for the $T\Psi C$ region of tRNAs; (b) An occurrence of the secondary expression S.

where $B = C|G|T$, $N = A|C|G|T$, $Y = C|T$, $D = A|G|T$, $R = A|G$ and $V = A|C|G$. With the same definition for the letters B, N, Y, D, R, and V, and the network expression E defined by $E = BNNYC$, this secondary structure may be written $EYDARCTT\overline{E}Y$. The problem is to

find all occurrences of such a structure in a DNA text. The more general approximate matching problem searches for matches with errors.

The algorithm goes along the following steps for matching with a secondary expression S.

1. Build a deterministic finite automaton $\mathcal{A}$ recognizing the language $\mathcal{S}$ defined by S when pairing constraints are erased.
2. Build over $\mathcal{A}$ a pushdown automaton $\mathcal{P}$. This automaton is designed to memorize which choices are made each time a union symbol | is met during the reading of the left strands of S (stacking phase), and to constraint the path followed during the reading of the right strands (unstacking phase).
3. When matching with errors is considered with a sequence of size n, an alignment graph is built with $n+1$ copies of the pushdown automaton $\mathcal{P}$ and a dynamical programming method is used to find the best alignment. Different valid (in the sense of the unstacking constraints) paths may lead to the same state, and it is therefore necessary to maintain during the dynamical programming step *sets of stacks*.

Note that Myers and Miller [2] give an algorithm to find approximate matching of regular expressions with complexity $O(np)$, where n is the size of the sequence and p is the size of the regular expression; this method applies to primary structures, but not to secondary structures.

2. Definitions

Definition 1 (network expression)**.** For $\alpha \in \Sigma \cup \{\epsilon\}$, the symbol α is a network expression. If E_1 and E_2 are network expressions, $E_1|E_2$ and E_1E_2 are network expressions.

Definition 2. The set *NetSet* is the set of network expressions.

Definition 3 (complement)**.** The complement $\overline{E}$ of a regular expression is defined by: *(i)* $\overline{\epsilon} = \epsilon$, *(ii)* $\overline{A} = T$, $\overline{T} = A$, $\overline{C} = G$, $\overline{G} = C$, *(iii)* $\overline{E_1E_2} = \overline{E_2}\,\overline{E_1}$ and $\overline{E_1|E_2} = \overline{E_1}|\overline{E_2}$.

Definition 4 (secondary expression)**.** A secondary expression is a sequence of elements of $\mathit{NetSet} \times \{p, sl, sr\}$, where p, sl, and sr respectively label unpaired, left strand, and right strand network expressions. The set of secondary expressions is recursively defined by: *(i)* if E is a network expression, then $S = (E, p)$ is a secondary structure; *(ii)* if E_1, E_2, E_3 are network expressions, and S' is a secondary expression, then the sequence $S = (E_1, p)(E_2, sl)S'(\overline{E_2}, sr)(E_3, p)$ is a secondary expression.

Definition 5. The language $\mathcal{L}(S)$ specified by a secondary expression S is recursively defined by:

- if $S = (E, p)$, then $\mathcal{L}(S) = \mathcal{L}(E)$;
- if $S = (E_1, p)(E_2, sl)S'(\overline{E_2}, sr)(E_3, p)$ such that E_1, E_2, E_3 are network expressions and S' is a secondary expression, then

$$\mathcal{L}(S) = \left\{ u \in \Sigma^\star \;\middle|\; u = vwx\overline{w}z \text{ for } v \in \mathcal{L}(E_1), w \in \mathcal{L}(E_2), z \in \mathcal{L}(E_3), \text{ and } x \in \mathcal{L}(S') \right\}.$$

Definition 6. For a secondary expression S, the *NetSet* expression $\mathit{NetSet}(S)$ is obtained by erasing the labels in the secondary expression.

As an example, if $S = (E_1, sl)(E_2, p)(\overline{E_1}, sr)$, then $\mathit{NetSet}(S) = E_1E_2\overline{E_1}$.

Definition 7 (approximate match)**.** Given a scoring function δ between two sequences (hamming distance, edit distance, measure of similarity), the set of sequences approximately matching a secondary expression S within k under scoring function δ is $\mathcal{L}_\delta(S, k) = \left\{ A \;\middle|\; \exists B \in \mathcal{L}(S), \delta(A, B) \leq k \right\}$. Note that this defines approximate matching of *primary* structures (sequences).

3. A Pushdown Automaton Recognizing a Secondary Expression

The language generated by a secondary expression S is a regular language recognized by a finite automaton. However, the size of the automaton is exponential in the number of symbols | in S. Using a pushdown automaton gives a more efficient algorithm.

El-Mabrouk anf Raffinot use a state labelled[1] finite pushdown automaton referred to later as ϵ-NFPA. Formally, an ϵ-NFPA $\mathcal{P} = \langle \Sigma, \Gamma, V, E, \lambda, \gamma, \theta, \phi, I \rangle$ consists of:

- an input alphabet Σ;
- a stack alphabet Γ;
- a set V of vertices called states;
- a set E of directed edges between vertices;
- a mapping λ of V on $\Sigma \cup \{\epsilon\}$;
- a mapping γ of $V \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ on a finite subset of $V \times \Gamma^\star$;
- an initial state θ;
- a final state ϕ;
- a particular stack symbol $I \in \Gamma$ called the start symbol.

If s and t are states, l is a letter of $\Sigma \cup \{\epsilon\}$, and the value of the top of the stack is Z, the interpretation of $\gamma(t, l, Z) = (s, \alpha)$, with $\alpha \in \Gamma^\star$ is that the automaton moves from state s to state t while reading letter l, popping Z from the top of the before pushing α into the stack. From there follows a partial mapping μ of $(V, \Sigma^\star, \Gamma^\star)$ onto itself defined by

$$(t, lw, Z\beta) \stackrel{\mu}{\longmapsto} (s, w, \alpha\beta) \quad \text{if} \quad \gamma(t, l, Z) = (s, \alpha).$$

Let $\mu^\star$ be the transitive closure of μ. The language accepted by the pushdown automaton $\mathcal{P}$ is

$$\mathcal{L}(\mathcal{P}) = \left\{ w \;\middle|\; (\theta, w, I) \stackrel{\mu^\star}{\longmapsto} (\phi, \epsilon, \alpha),\ \alpha \in \Gamma^\star \right\}.$$

(Note that by construction, for secondary structures, we have $\alpha = \epsilon$ in the last equation.) The letter μ will be omitted in what follows.

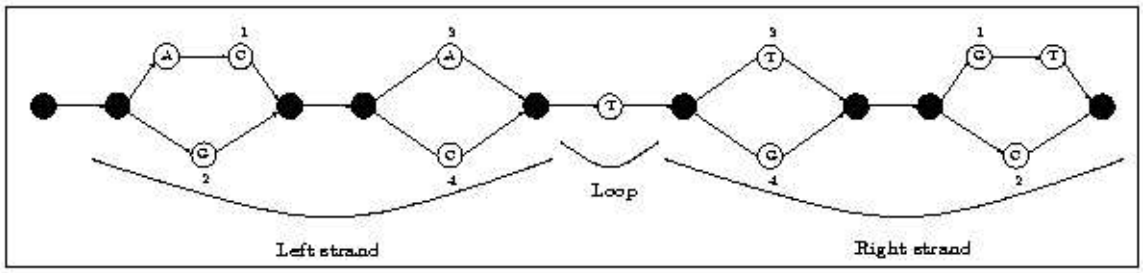


FIGURE 2. The state labelled ε-NFA recognizing $NetExp(S)$, for $S = (E_1, sl)(E_2, p)(\overline{E_1}, sr)$, with $E_1 = ((AC)|G)(A|C)$ and $E_2 = T$. Black states are labelled by ε. The numbers $1, 2, 3, 4$ mark the marked states. The loop is an unpaired region.

The construction of the automaton $\mathcal{P}$ recognizing S goes along the following steps:

1. build a state-labelled ϵ-NFA $\mathcal{A}$ recognizing Netset(S), with labelling function λ;
2. mark the possible choices for each union symbol | of the left strands of S;
3. define the rules for stacking the marks during reading the left strands of S;
4. define the unstacking and transitions rules while reading the right strands of S.

[1] A corresponding classical transition labelled automaton would be such that all the transitions entering a state are labelled with the same letter of $\Sigma \cup \{\epsilon\}$, whatever this state is.

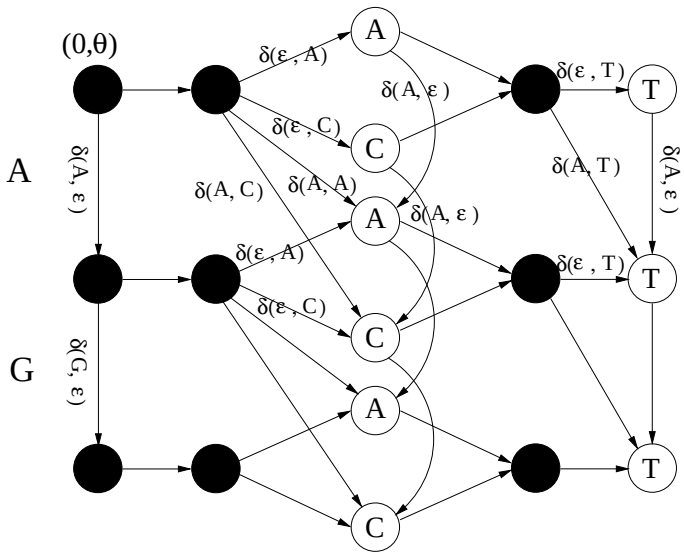


FIGURE 3. The alignment graph for a sequence $Q = AG$ versus the network expression $(A|C)T$.

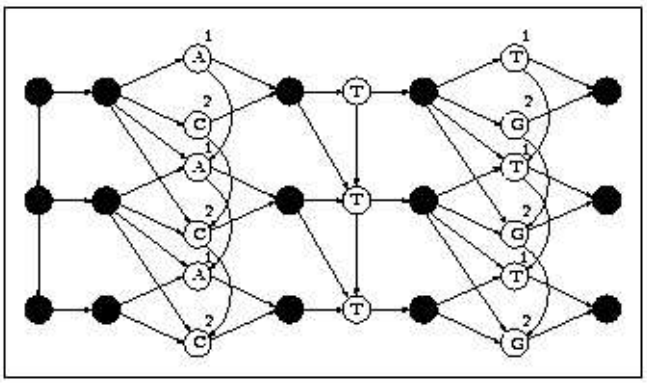


FIGURE 4. The alignment graph for $Q = \mathrm{AT}$ versus $NetExp(S) = (A|C)T(T|G)$.

Marking the states. For each $(E_{i_1}|E_{i_2})$ expression of S, where neither E_{i_1} nor E_{i_2} contains a union symbol $|$, and E_{i_1} and E_{i_2} are left strands expressions, let s_{i_j} be the state of $\mathcal{A}$ corresponding to the last atomic expression of E_{i_j} $(j = 1, 2)$. Each such state is marked unambiguously with a letter γ of Γ (two different states are marked by different letters). The other states of the left strands and the states of unpaired regions remain unmarked. Mark the states of the right strands by mirroring the corresponding left strands. An example of marking is given in Figure 2 where Γ is a subset of $\mathbb{N}^+$. Let ν be the mapping associating to a marked state s its mark $\nu(s)$.

Defining the mapping γ. The mapping γ of the pushdown automaton $\mathcal{P}$ is defined as follows:

Let Z be the top symbol of the stack, l be any character of $\Sigma \cup \epsilon$, s be any state, and $t \rightarrow s$ be any edge leading to s in the automaton $\mathcal{A}$. The transition $\gamma(t, l, Z)$ is defined in the automaton $\mathcal{P}$ if and only if $l = \lambda(s)$. In that case:

- if s is an unmarked state, then $\gamma\big(t, \lambda(s), Z\big) = (s, Z)$;
- if s is a marked sl-state, then $\gamma\big(t, \lambda(s), Z\big) = \big(s, \nu(s)Z\big)$;
- if s is a sr-state such that $\nu(s) = Z$, then $\gamma\big(t, \lambda(s), Z\big) = (s, \epsilon)$.

This definition of γ constrains the traversal of the right strands to be the mirror of the traversal of the corresponding left strand.

Lemma 1. *The pushdown automaton $\mathcal{P}$ recognizes the language generated by the secondary expression S.*

See [1] for a proof.

4. Matching with Errors and Alignment Graph

For the problem of aligning a network expression E to a sequence Q of size n within a threshold k, Myers and Miller[2] showed in [2] that it is easier to reduce the problem to one of finding a shortest source-to-sink path in a weighted and directed *alignment graph* depending on E and Q. The graph is constructed from $n + 1$ copies of the ϵ-NFA recognizing E, arranged one on top of another. Figure 3 shows an alignment graph of the expression $E = (A|C)T$ and of the sequence $Q = AG$.

Formally, the vertices of the graph are the pairs (i, s) for $1 \leq i \leq n + 1$ and $s \in V$. Insertion, deletion and substitution edges are defined as follows:

- if $i > 0$, then there is a *deletion edge* from $(i - 1, s)$;
- if $s \neq \theta$, then for each state t such that $t \to s$, there is an *insertion edge* from (i, t);
- if $i > 0$ and $s \neq \theta$, then for each state t such that $t \to s$, there is a *substitution edge* from $(i - 1, t)$.

The construction of Myers and Miller is applied to the pushdown automaton $\mathcal{P}$. Figure 4 shows the alignment obtained when matching $Q = AT$ against $NetExp(S) = (A|C)T(T|G)$, with $S = (A|C)T(\overline{A|C})$. The problem is that several paths may lead to the same state; it is therefore necessary to maintain sets of stacks. For a state (i, s), let $\Pi(i, s)$ be the set of least cost paths from $(0, \theta)$ to (i, s). For a path $\pi \in \Pi(i, s)$ let $\sigma(\pi)$ be the sequence obtained by concatenating the labels λ of the states on this path. The set of stacks of a state (i, s) is defined by

$$\mathrm{Stack}(i, s) = \Big\{ \alpha \in \Gamma^\star \;\Big|\; \exists \pi \in \Pi(i, s) \text{ such that } (\theta, \sigma(\pi), I) \xrightarrow{\star} (s, \epsilon, \alpha) \Big\}.$$

A path aligning the first i letters $Q[1, i]$ of Q and a sequence $\sigma(\pi)$ for a state s is a *valid path* if it respects the constraints given by the secondary expression. Therefore $\sigma(\pi)$ must belong to the language recognized by $\mathcal{P}(s)$, where s is made the final state of $\mathcal{P}$.

An edge from (j, t) to (i, s) is valid (noted $(j, t) \xrightarrow{v} (i, s)$) if the two following conditions are met:

- $(j, t) \to (i, s)$ is an insertion, deletion or substitution edge;
- if $(j, t) \to (i, s)$ is a substitution or deletion edge and s is a marked sr-state, then there is a stack P in $\mathrm{Stack}(j, t)$ with top symbol $\lambda(s)$.

Thus the problem of approximately matching a prefix of size i of Q to a prefix $\sigma(\pi)$ of a word of $\mathcal{L}(S)$ is equivalent to finding a least cost valid path between source vertex $(0, \theta)$ and (i, s). Computing such a path may be done by dynamic programming (procedure *CentralRec* of Figure 5).

[2]There is an error in this section that follows the content of the talk: a suboptimal left strand alignment may lead to an optimal right strand alignment. El-Mabrouk and Raffinot are working at correcting this error, that is compatible with Myers and Miller's approach.

procedure CentralRec:

1. $C(0, \theta) = 0$
2. $C(i, s) = \min_{(i,t) \xrightarrow{v} (i,s)} \{C(i, t) + \delta(\varepsilon, \lambda(s))\}$
3. **if** $(i-1, t) \xrightarrow{v} (i, s)$ **then**
4. $\quad C(i, s) = \min\{C(i, s), C(i-1, t) + \delta(q_i, \lambda(s))\}$
5. **if** $(i-1, s) \xrightarrow{v} (i, s)$ **then**
6. $\quad C(i, s) = \min\{C(i, s), C(i-1, s) + \delta(q_i, \varepsilon)\}$

FIGURE 5. Central recurrence computing the value of a least cost valid path from the source vertex to each vertex (i, s) of the alignment graph. The letter q_i is the letter at position i in Q.

Maintaining the set of stacks. El-Mabrouk and Raffinot implement the set of stacks as binary trees. They define a set of operations over these trees:

- *Insert*: a new node is inserted at the top of a tree;
- *Remove*: remove the top element;
- *Combine*: a new root points to trees T_1 and T_2 that were previously constructed;
- *Merge*: "superposition" of two trees; there must be coherence between the nodes of the two trees.

During the reading of the *sl*-strands, trees are grown through Insert, Combine and Merge operations, while during the reading of the right strand, the Remove operation is used, and one of the left or right tree is substituted to the tree representing the stacks.

Approximate matching algorithm. When looking for approximate matches of a secondary expression against a sequence, one alignment graph is constructed for each position of the sequence. Note that practically only two copies of the automaton $\mathcal{P}$ are maintained.

5. Complexity

Let p be the size of the secondary expression S (the number of all characters of the network expression *NetExp*(S)), and r be the number of symbols | in S. Let n be the size of the genome being traversed.

There are $O(np)$ vertices in the alignment graphs, and the in-degree of the vertices is at most 3. Computing the value at each vertex by *CentralRec* takes $O(1)$ time. Thus, computing all the costs $C(i, s)$ can be done in $O(np)$ time. When considering the stacks, the procedure *Merge* is $O(r)$ in the worst case (other procedures have lower complexity).

This gives a final complexity of $O(rpn)$.

As an example, scanning the 4MB of *bacillus subtillis* with a 200 base long secondary structure takes 215 seconds.

Bibliography

[1] El-Mabrouk (N.) and Raffinot (M.). – Approximate matching of secondary structures. In *Sixth Annual International Conference on Computational Molecular Biology*. pp. 156–164. – ACM Press, 2002.

[2] Myers (Eugene W.) and Miller (Webb). – Approximate matching of regular expressions. *Bulletin of Mathematical Biology*, vol. 51, n° 1, 1989, pp. 5–37.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 113–118.

Available online at the URL
http://algo.inria.fr/seminars/.

Les algorithmes évolutionnaires : état de l'art et enjeux

Marc Schoenauer
Projet Fractales, Inria Rocquencourt (France)

October 15, 2001

Summary by Philippe Dumas

Abstract

Les algorithmes évolutionnaires sont des algorithmes d'optimisation stochastique fondés sur un parallèle grossier avec l'évolution darwinienne des populations biologiques. Ils fournissent une approche heuristique, à l'occasion performante et dans certains cas prouvée.

Un algorithme évolutionnaire a pour but d'optimiser une fonction réelle. Il repose sur une vision darwinienne relativement simpliste et une optimitisation stochastique résumées dans le diagramme de la Figure 1. La fonction f à optimiser, appelée aussi performance, est définie sur un espace de recherche Ω. L'algorithme fait évoluer une population, un sous-ensemble de l'espace de recherche. Cette évolution résulte d'une part d'un darwinisme artificiel, qui se manifeste par la sélection et le remplacement et ne dépend que de la performance f ; d'autre part de l'effet du hasard, qui s'exprime dans l'initialisation et les opérateurs de variation et ne dépend que de la représentation de l'espace de recherche. L'idée fondamentale est que la sélection favorise les individus qui optimisent la performance et que les variations font apparaître dans la population sélectionnée des individus que l'ont peut espérer meilleurs au regard de la performance. Dans cette évolution, les générations successives de la population restent à taille constante et l'aspect stochastique ne dépend que de la génération précédente.

La mise en place d'un algorithme évolutionnaire est complexe et le coût de calcul est important. De tels algorithmes sont donc destinés à traiter des problèmes qui n'ont pas de solutions classiques. Si l'on veut bien négliger un discours pseudo-scientifique et des querelles de chapelles qui ont encombré le domaine dans ses premières décennies[1] il faut reconnaître à l'approche évolutionnaire des réussites frappantes.

[1]Ces aspects désagréables ne figuraient absolument pas dans l'exposé de M. Schoenauer.

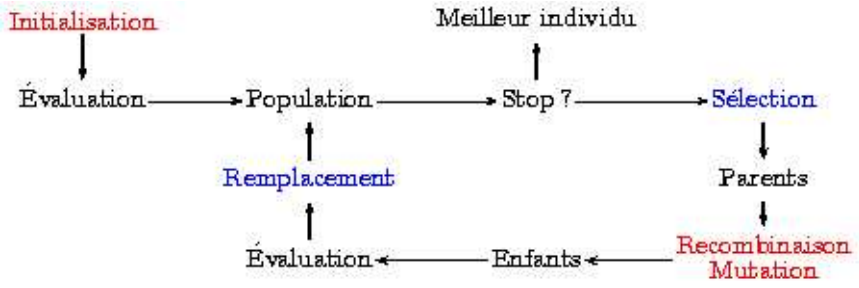


Figure 1. Variation (stochastique, en rouge) et darwinisme (déterministe ou stochastique, en bleu) sont les notions de base de l'algorithmique évolutionnaire.

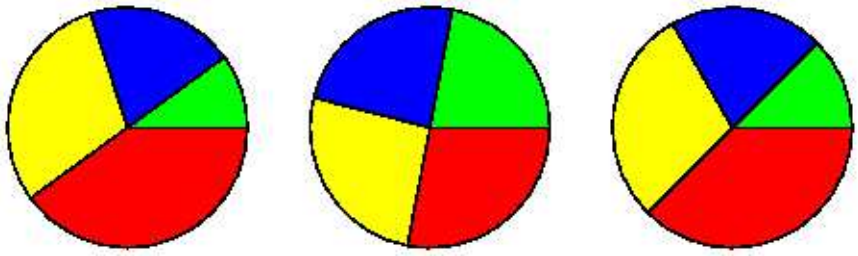


FIGURE 2. Une modification anodine de la fonction objectif peut avoir une effet marqué sur le processus de sélection par la roulette.

1. **Représentation binaire**

1.1. **Algorithme génétique classique.** Le modèle binaire [4] est le plus ancien et on parle à son sujet d'algorithme génétique classique. L'espace de recherche est l'ensemble des mots binaires de longueur donnée $\Omega = \{0,1\}^N$. Un élément de Ω est baptisé chromosome. La population est à chaque instant t entier constituée de μ chromosomes X_i^t, qui forment un vecteur X^t. L'initialisation se fait suivant la loi uniforme. L'évaluation est simplement l'évaluation des $f\left(X_i^t\right)$. La sélection suit la méthode de la roulette : à chacun des chromosomes X_i^t est associé sur un cercle un secteur d'angle proportionnel à la valeur $f\left(X_i^t\right)$; on effectue μ tirages suivant ce modèle probabiliste et à chaque fois que la roulette fournit le chromosome X_i^t, celui-ci est copié en un chromosome X_j'. On obtient ainsi une population intermédiaire X', où les chromosomes les meilleurs au regard de la performance sont présents en plusieurs exemplaires, alors que les pires sont éliminés. Ensuite sont effectués des recombinaisons entre ces chromosomes : $\mu/2$ couples $\left(X_j', X_k'\right)$ sont tirés au hasard et avec une probabilité p_c sont recombinés ; si la recombinaison a lieu, un point de croisement est tiré au hasard et les brins sont échangés. Après cela chaque chromosome subit une mutation : avec une probabilité p_m chacun de ses bits est changé en son complément. Au terme de ce processus, on dispose de la génération suivante X^{t+1}. Dans ce modèle, il n'y a pas de remplacement.

Le réglage des paramètres comme μ, p_c, p_m ou le test d'arrêt est délicat. Indiquons un simple problème : l'optimisation de f est équivalente à l'optimisation de $f' = af + b$ avec a une constante positive et b une constante. Cependant l'introduction de a et b peut avoir un effet drastique sur la sélection. On le voit sur la Figure 2, où l'on a supposé que la population comporte quatre individus. À gauche la performance prend les quatre valeurs 1, 2, 3, 4, d'où des secteurs de taille 0,1, 0,2, 0,3, 0,4. Au centre on a augmenté la performance de 10, ce qui donne les valeurs 11, 12, 13, 14 et des secteurs de taille 0,22, 0,24, 0,26, 0,28. Le tirage aléatoire est très fortement modifié et le meilleur individu est à peine avantagé. Ce problème peut être traité par une mise à l'échelle, qui est effectuée à chaque génération. Dans la mise à l'échelle linéaire, a et b sont choisis pour que la moyenne de la performance sur la population reste la même, $\overline{f}' = \overline{f}$, mais la meilleure valeur $f'_{\max}$ satisfait $f'_{\max} = \rho f_{\max}$ avec ρ choisi entre 1 et 2. C'est ce qu'on a appliqué, avec $\rho = 1{,}5$ pour obtenir la version de droite, qui correspond à $a = 5/6$, $b = 5/12$ et aux quatre valeurs 1,25, 2,08, 2,9, 3,75 pour la performance.

1.2. **Algorithme évolutionnaire.** La représentation binaire, longtemps dominante, a été vivement critiquée car peu naturelle. Pour résoudre un problème numérique dont les solutions sont, par exemple, cherchées dans $[\,0,1\,]$ avec une précision de 10^{-10}, elle amène à passer en représentation binaire avec des mots de trente-quatre bits (au moins), parce que 34 est le premier entier k satisfaisant à $2^k \geq 10^{10}$. De plus les opérations peuvent ne pas être naturelles ou ne pas faire sens. Par exemple la recombinaison peut produire des chromosomes qui n'ont pas d'interprétation dans

le modèle. Dans ce cas, on procède généralement à une pénalisation de ces chromosomes, qui vont ainsi être éliminés dans la sélection.

Ces questions ont modifié le point de vue des spécialistes, qui parlent maintenant d'algorithmes évolutionnaires. On cherche une représentation naturelle des données et des opérateurs génétiques qui font sens dans le problème. Un bon algorithme évolutionnaire utilise une représentation qui permet à un spécialiste du domaine d'application d'interpréter les caractéristiques de la population et de proposer des choix de paramètres qui font converger l'algorithme vers une solution.

2. **Représentation réelle**

Dans une représentation réelle, l'espace de recherche est une partie Ω d'un $\mathbb{R}^N$. La similitude avec le darwinisme se fait par des stratégies évolutionnaires [1] dans lesquelles les mutations sont au premier plan, alors que les algorithmes génétiques misent plutôt sur la recombinaison.

Les mutations reposent essentiellement sur l'ajout d'un bruit gaussien centré d'écart-type σ à chaque coordonnées de l'individu, tout l'art étant dans le choix du σ. On peut par exemple appliquer la règle du cinquième. Une mutation est réussie si elle fournit un individu meilleur que son parent[2] ; on note τ la proportion de mutations réussies sur les T dernières générations. Si τ est plus grand que 1/5, on change σ en $1{,}22\,\sigma$ (on élargit la recherche) ; sinon on change σ en $0{,}83\,\sigma$ (on restreint la recherche). S'il y a beaucoup de mutations réussies, les individus sont près d'un optimum local, mais l'optimum global risque d'être manqué ; s'il y a peu de mutations réussies, la région explorée est trop vaste et il convient de la restreindre.

On envisage aussi des mutations adaptatives. Chaque individu porte un paramètre σ, ce qui signifie que l'espace de recherche est maintenant une partie de $\mathbb{R}^N \times \mathbb{R}_+$. La mutation s'effectue en deux temps : on mute d'abord l'écart-type σ (en le multipliant par l'exponentielle d'une variable gaussienne), puis l'individu lui-même en utilisant le nouvel écart-type. Si la valeur de σ est aberrante, elle ne va pas fournir de mutation réussie et l'individu va être éliminé. Par contre les individus qui survivent ont une bonne valeur de σ. L'idée que l'individu porte les paramètres de la stratégie évolutionnaire a été développée en adjoignant à chaque individu une matrice de corrélation entre ses coordonnées, pour que les différentes coordonnées ne soient pas traitées indépendamment.

La recombinaison se fait par barycentre $z = (1-\alpha)x + \alpha y$, avec un poids α qui peut être une variable aléatoire.

À partir de la population de μ individus, les mutations et recombinaisons produisent λ nouveaux individus. La sélection est déterministe et fournit les μ individus de la génération suivante. Dans la stratégie (μ, λ), les μ meilleurs individus parmi les λ nouveaux individus ($\lambda > \mu$) sont conservés. Dans la stratégie $(\mu + \lambda)$, ce sont les μ meilleurs parmi les $\mu + \lambda$ individus disponibles ($\lambda > 1$) qui sont conservés. On ainsi appliqué le processus de remplacement de la Figure 1.

3. **Autres représentations**

3.1. **Représentation non structurée.** La recherche de structures matérielles optimales amène à une discrétisation de l'espace en cellules par découpages par des plans parallèles aux plans de coordonnées. Une cellule fait partie de la structure si elle est marquée 1 et sinon 0. On a ainsi un tableau de bits. L'inconvénient est qu'une représentation fine demande une énorme place en mémoire. Une représentation plus compacte repose sur la notion de diagramme de Voronoï. On fixe un domaine borné et un individu est une famille de N points marqués dans ce domaine. À cette famille est associé son diagramme de Voronoï, dans lequel la cellule attachée à un point de la famille est constituée des points de l'espace qui sont plus près de ce point que des autres points de

[2]On voit qu'une évaluation a lieu dans la sélection. Le diagramme de la Figure 1 illustre seulement une idée.

la famille. Si le point de référence est marqué 1, alors tous les points de sa cellule font partie de la structure ; si le point est marqué 0, alors les points de la cellule ne sont pas dans la structure.

On définit sur cette représentation des opérateurs d'évolutions. Par exemple la recombinaison de deux individus consiste à les couper tous les deux par un même plan de l'espace et à échanger les points marqués des deux individus qui sont dans l'un des demi-espaces limité par le plan.

3.2. **Représentation en arbre.** Un thème éculé du domaine est celui de l'ajustement. On dispose de données numériques et on cherche une fonction qui les modélise. Un espace de recherche possible est celui des arbres qui représentent des expressions de fonction. La recombinaison se fait par échange de sous-arbres. La mutation consiste en le remplacement d'un sous-arbre par un arbre aléatoire et en la mutation gaussienne des constantes. La performance tient compte à la fois de l'erreur quadratique dans l'ajustement et de la complexité de l'expression comptée à l'aide du nombre de nœuds de l'arbre.

3.3. **Programmation génétique.** Un pas supplémentaire dans la généralisation amène à accepter une population constituée de programmes informatiques. On parle alors de programmation génétique. Une éventuelle solution du problème posé est obtenu en appliquant un programme de la population à un embryon.

L'espace de recherche est constitué de programmes c'est-à-dire d'arbres étiquetés par des symboles de fonctions, qui représentent des fonctions mathématiques comme les fonctions et opérations usuelles (constantes, noms de variables, fonctions exp, ln, ..., opérations $+$, $\times$, ...) ou des fonctions informatiques (valeurs de vérité, conditionnement, boucle). La performance d'un programme se mesure sur la solution qu'il produit à partir de l'embryon. La population initiale peut être constituée de tous les arbres dont la profondeur est bornée par un certain entier. Les opérations génétiques sont celles que nous avons vues au sujet des arbres.

Cette idée a par exemple été utilisée dans la conception de circuits analogiques. L'embryon est un circuit simple et les fonctions qui apparaissent dans les programmes sont des modificateurs de circuits. La performance se mesure en testant le circuit pour un échantillon de fréquences. Il faut noter que pour la conception d'un filtre passe-bas 60 dB exposée dans [5], la population est de taille $\mu = 640\,000$, ce qui limite la programmation génétique à des exercices d'école.

4. **Domaines d'application**

L'approche évolutionnaire est valablement appliquée quand une technique classique n'est pas disponible ; quand le coût de calcul des méthodes standard est trop élevé ; quand la performance n'a pas les propriétés de régularité que requièrent les méthodes standard. Une démarche naïve sur des exemples d'école est donc sans intérêt. Citons trois exemples dans lesquels l'approche évolutionnaire montre son intérêt [3].

Le calcul du profil d'une aile d'avion est extrêmement coûteux. Une approche évolutionnaire, dans laquelle la performance est la différence entre la pression calculée et la pression désirée, a permis une amélioration de 14% de cette performance par rapport aux méthodes classiques.

Un algorithme génétique a permis de sélectionner quatre-cents chaînes peptidiques potentiellement actives comme bactéricide. Cinq d'entre elles ont été synthétisées, par exemple pour concevoir de nouveaux additifs alimentaires anti-bactériens.

La radiothérapie utilise un faisceau radioactif qui détruit les tumeurs mais aussi des tissus sains. On savait prévoir la forme de la zone lésée suivant celle du faisceau. Un algorithme génétique a permis de déterminer la forme du faisceau à produire pour atteindre une zone de forme donnée.

chromosome	état									
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
00	2	1	1	1	0	0	0	0	0	0
01	0	1	0	0	2	1	1	0	0	0
10	0	0	1	0	0	1	0	2	1	0
11	0	0	0	1	0	0	1	0	1	2

FIGURE 3. La matrice Z dans le cas où les populations sont constituées de deux chromosomes de longueur 2, les chromosomes et les états étant rangés dans un ordre lexicographique.

5. **Théorie**

5.1. **Schémas.** Un schéma H est un mot sur l'alphabet $\{0, 1, \#\}$. Un mot binaire est une réalisation du schéma H, s'il coïncide avec le schéma pour les lettres différentes du joker $\#$. L'ordre d'un schéma est le nombre $\omega(H)$ de caractères 0 ou 1 qu'il contient et sa longueur utile $\lambda(H)$ est la distance maximale entre deux lettres de H autres que le joker $\#$. Holland a prouvé l'énoncé suivant.

Théorème 1 (dit des schémas)**.** *La suite des $m(H,t)$, nombre de chromosomes qui réalisent le schéma H à la génération t, satisfait à l'inégalité (***E** *désigne l'espérance)*

$$\mathbf{E}\ m(H,t+1) \geq \mathbf{E}\ m(H,t)\frac{\overline{f}(H,t)}{\overline{f}(X^t)}\left(1-\frac{\lambda(H)}{N-1}p_c\right)(1-p_m)^{\omega(H)}\,.$$

Dans cet énoncé $\overline{f}(H,t)$ est la valeur moyenne de la performance sur les réalisations du schéma dans la population, alors que $\overline{f}\left(X^t\right)$ est la moyenne de la performance sur la population. L'inégalité vient du fait que le schéma peut apparaître par mutation. Ce résultat est interprété de la manière suivante : un schéma de faible longueur utile, de faible ordre, dont la performance est supérieure à la moyenne, a un nombre de chromosomes qui augmente exponentiellement dans la population. Il fournit une explication à la convergence vers un optimum.

5.2. **Chaînes de Markov.** La théorie des schémas ne permet pas d'expliquer la composition de la population au cours des générations. Une approche par chaînes de Markov a été développée dans [6, 7]. Le nombre de chromosomes possibles est 2^N. Chaque tirage avec remise de μ individus dans ces 2^N chromosomes fournit une population. Le nombre d'états de la chaîne de Markov est donc

$$\nu = \binom{\mu + 2^N - 1}{2^N - 1}.$$

Chaque état peut être vu comme une ligne d'une matrice $Z = (z_{c,s})$ (Figure 3), dans laquelle $z_{c,s}$ est le nombre d'occurrences du chromosome c dans la population s.

Théorème 2. *Si la probabilité de mutation p_m est non nulle, la chaîne de Markov associée à un algorithme génétique classique est ergodique. En particulier elle possède une distribution limite.*

Le théorème de Perron–Frobenius permet de préciser le comportement de la chaîne.

Cerf [2] a utilisé la notion de chaîne de Markov, mais avec une approche différente basée sur la théorie des perturbations stochastiques des systèmes dynamiques. Un algorithme génétique simpliste qui ne comporte pas d'opérateurs de variations et effectue une sélection dégénérée est perturbé selon un paramètre ℓ qui gouverne la probabilité de mutation ($p_m = \ell^{-a}/N$), la probabilité de recombinaison ($p_c = \ell^{-b}$) et aussi le mode de sélection. Cette approche fournit un seuil pour la taille de la population, dans le cas inhomogène où ℓ dépend du temps.

Théorème 3. *On suppose que la suite* (ℓ_t) *tend vers* $+\infty$ *et que la taille* μ *de la population est supérieure à une valeur critique* μ^* *qui s'exprime à l'aide de la performance et des paramètres du modèle. Il y a alors équivalence entre les deux assertions :*

- *il existe des exposants* α *et* β $(0 < \alpha < \beta)$ *satisfaisant à* $\sum_{t=0}^{+\infty} \frac{1}{\ell_t^\alpha} = +\infty$ *et* $\sum_{t=0}^{+\infty} \frac{1}{\ell_t^\beta} < +\infty$ *;*
- *pour toute population originelle dans l'espace de recherche, en un temps fini, la population est presque sûrement toute entière dans l'ensemble des individus qui fournissent l'optimum global de la performance.*

Ce résultat n'a pas de conséquence pratique mais il a comme corollaire que la taille critique de la population est de l'ordre de N, ce que Goldberg avait obtenu expérimentalement.

5.3. **Convergence des stratégies évolutionnaires.** En s'appuyant sur la théorie des surmartingales, Rudolph [8] a prouvé un énoncé qui donne en particulier le résultat suivant.

Théorème 4. *Une stratégie évolutionnaire, de type* $(1, \lambda)$ *avec* $\lambda \geq 2$ *et des mutations sphériques (les mutations font passer d'un point* x *de* $\mathbb{R}^N$ *à un point* $x + \varepsilon$ *où* ε *est une variable aléatoire uniforme sur une sphère de centre* 0 *de rayon adapté à la performance) et une performance strictement convexe au voisinage du point* X^* *qui fournit l'optimum, converge presque sûrement et en moyenne vers* X^**. De plus la convergence est géométrique.*

5.4. **Problèmes.** Les énoncés qui viennent d'être cités ne donnent qu'une faible idée de ce qui a été produit. Cependant il faut conclure que l'approche évolutionnaire manque de fondement théorique. Ceci est dû au fait que dans chaque cas les algorithmes sont adaptés à la situation par des choix de paramètres et des variantes que ne couvrent pas la théorie. Il en résulte par exemple que les dynamiques associées à des représentations différentes d'un même problème sont différentes et ne sont pas comparables faute d'un cadre adapté.

Bibliography

[1] Bäck (Thomas), Hoffmeister (Frank), and Schwefel (Hans-Paul). – A survey of evolution strategies. In Belew (Richard K.) and Booker (Lashon B.) (editors), *Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, CA, USA, July 1991*. pp. 2–9. – Morgan Kaufmann, 1991.

[2] Cerf (Raphaël). – *Une théorie asymptotique des algorithmes génétiques*. – Thèse de doctorat, Université de Montpelier II, 1994.

[3] Evolution@work. – Disponible en ligne à `http://www.evonet.polytechnique.fr/evoweb/resources/evolution_work/`.

[4] Holland (John H.). – *Adaptation in natural and artificial systems*. – University of Michigan Press, Ann Arbor, Mich., 1975, ix+183p. An introductory analysis with applications to biology, control, and artificial intelligence.

[5] Koza (John R.), Andre (David), Bennett, III (Forrest H.), and Keane (Martin A.). – Evolution of a low-distortion, low-bias 60 decibel op amp with good frequency generalization using genetic programming. In Koza (John R.) (editor), *Late breaking papers at the Genetic Programming 1996 conference, Stanford University, July 28-31, 1996*, pp. 94–100. – 1996. Disponible en ligne à `http://www.genetic-programming.com/jkpubs96.html`.

[6] Liepins (Gunar E.) and Vose (Michael D.). – Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 2, n° 2, 1990, pp. 4–30.

[7] Nix (Allen E.) and Vose (Michael D.). – Modelling genetic algorithms wit Markov chains. *Annals of Mathematics and Artificial Intelligence*, vol. 5, n° 1, 1992, pp. 79–88.

[8] Rudolph (Günter). – Convergence of non-elitist strategies. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, June 27-29, 1994, Orlando, Florida, USA*, pp. 63–66. – 1994.

[9] Schoenauer (Marc) and Michalewicz (Zbigniew). – Evolutionary computation. *Control and Cybernetics*, vol. 26, n° 3, 1997, pp. 307–338. – Disponible en ligne à `http://www.eeaax.polytechnique.fr/papers.html`.

Part VII

ALEA'2002 Lecture Notes

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 121–150.

Available online at the URL
http://algo.inria.fr/seminars/.

Systèmes dynamiques et algorithmique[†]

Viviane Baladi[(a)] *and Brigitte Vallée*[(b)]

[(a)]Institut Mathématique de Jussieu (France) and [(b)]GREYC, Université de Caen (France)

March 18 and 19, 2002

Summary by Frédéric Chazal[‡]*, Véronique Maume-Deschamps*[§]*, and Brigitte Vallée*[¶]

L'analyse en moyenne d'algorithmes vise à déterminer le comportement « moyen » des algorithmes. Par opposition à la complexité dans le pire des cas, la complexité moyenne d'un algorithme permet d'appréhender les performances de l'algorithme de manière « réaliste ». Il est maintenant classique, en analyse d'algorithmes, de travailler avec un outil essentiel, celui des séries génératrices. Les principales opérations algébriques sur les structures de données ou les algorithmes se traduisent en opérations formelles sur les séries génératrices. Quand les séries génératrices sont vues comme des fonctions de variable complexe, leur singularité dominante permet d'obtenir des renseignements précieux sur le comportement asymptotique moyen de l'algorithme. Cette méthodologie est décrite par exemple dans les livres de Flajolet et Sedgewick [22, 25].

Cependant, quand les algorithmes sont trop « corrélés », cette méthodologie ne peut plus s'appliquer, car les opérations sur les algorithmes ne se traduisent plus aisément en opérations sur les séries génératrices. C'est alors une idée tout à fait naturelle que de considérer un algorithme et l'ensemble de ses données comme un système dynamique. Les données sont alors les particules du système qui sont soumises au « champ » créé par les opérations que leur font subir l'algorithme. À un système dynamique, on associe classiquement, depuis Ruelle, un opérateur appelé opérateur de transfert, ou opérateur de Ruelle, [38, 39] qui permet de décrire l'évolution du système. Cet opérateur dépend d'un paramètre s, est désigné par $\mathbf{H}_s$, et agit sur un espace de fonctions d'une variable.

Opérateur de transfert = opérateur générateur. L'idée originale consiste à détourner l'opérateur de transfert de son usage habituel et à le considérer comme un opérateur « super-générateur », en ce sens qu'il engendre lui-même les séries génératrices associées à l'algorithme. Les opérations sur les algorithmes continuent à se traduire en opérations sur ces opérateurs générateurs. Par ailleurs, aussitôt que le système dynamique possède de « bonnes propriétés », cet opérateur a des propriétés spectrales dominantes : il existe une valeur propre dominante $\lambda(s)$ positive qui est séparée du reste du spectre par un saut spectral. Cette valeur propre dominante joue ainsi un rôle essentiel car c'est elle qui concentre les propriétés essentielles du système. C'est elle qui va jouer le même rôle que la singularité dominante dans le cadre classique des séries génératrices, et va ainsi permettre d'appréhender le comportement asymptotique moyen de l'algorithme, même quand celui-ci est « corrélé ». C'est la philosophie générale (voir Figure 1). De fait, l'opérateur de

[†]Notes de cours pour le cours donné pendant le groupe de travail ALÉA'02 au CIRM à Luminy (France).

[‡]Université de Bourgogne, B. P. 47870, 21078 Dijon Cedex, France ; email: fchazal@u-bourgogne.fr.

[§]Université de Bourgogne, B. P. 47870, 21078 Dijon Cedex, France ; email: vmaume@u-bourgogne.fr.

[¶]GREYC, Université de Caen, 14032 Caen Cedex, France ; email: brigitte.vallee@info.unicaen.fr.

transfert ne peut pas vraiment être utilisé « tel que » en analyse d'algorithmes, il a souvent besoin d'être généralisé, afin d'opérer sur des fonctions de plusieurs variables. Cet opérateur généralisé, désigné par $\mathfrak{H}_s$, étend d'ailleurs dans un sens fort l'opérateur classique $\mathbf{H}_s$, puisqu'il a la même valeur propre dominante.

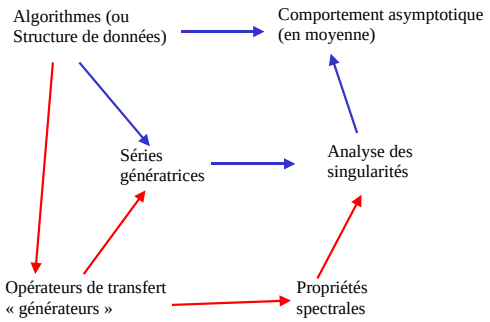


FIGURE 1. Analyse classique, analyse dynamique.

Les domaines d'application. Cette méthodologie, qu'on appelle « analyse dynamique des algorithmes » s'est installée relativement récemment en analyse d'algorithmes (1995). Elle peut déjà s'appliquer à deux domaines algorithmiques larges, l'algorithmique arithmétique et l'algorithmique du texte. Dans chacun de ces domaines, la méthode prouve son efficacité en permettant de résoudre des problèmes inaccessibles à la méthode classique. La démarche est différente dans les deux domaines : en algorithmique arithmétique, on cherche à analyser des algorithmes existants et utilisés. En algorithmique du texte, il y a une double volonté : on cherche à modéliser le concept de source, qui est le mécanisme sous-jacent à tous les algorithmes de texte, puisque c'est lui qui produit le texte ; on cherche ensuite à analyser les algorithmes quand les textes sont produits sous ce modèle. Bien que les deux domaines soient *a priori* disjoints, il y a de fait un transfert de méthodes de l'un des domaines à l'autre : en algorithmique arithmétique, le concept a été utilisé pour des systèmes dynamiques de plus en plus complexes qui se sont « spontanément » présentés, lors de l'analyse d'algorithmes classiques existant. Ces systèmes qui apparaissent naturellement en algorithmique, apparaissent souvent comme non classiques aux dynamiciens. Il était alors tentant d'utiliser cette expérience pour élargir la possible modélisation dans le contexte de l'algorithmique du texte, et pour généraliser progressivement la définition des sources dynamiques.

Plan. On commence par rappeler, dans la Section 1, les propriétés de base des systèmes dynamiques. Puis, la Section 2 présente les opérateurs qui seront utilisés dans les analyses et qui se situent dans la lignée des opérateurs de transferts des dynamiciens. La Section 3 décrit le cadre d'analyse fonctionnelle nécessaire à l'obtention des propriétés spectrales. Alors, tout est prêt pour décrire l'analyse dynamique, et ce, à travers deux champs d'application : le texte dans la Section 4 et l'arithmétique dans la Section 5.

Ces notes visent à introduire le sujet de l'analyse dynamique des algorithmes, et à donner quelques exemples clés. Elles sont complétées par une bibliographie assez exhaustive. On pourra aussi consulter la page du groupe d'Analyse dynamique à l'adresse `http://users.info.unicaen.fr/~daireaux/ANADY/index.html`.

Ces notes correspondent à un cours donné par Viviane Baladi et Brigitte Vallée lors des journées annuelles du groupe de travail ALÉA en mars 2002. Les Sections 1 et 3 résument plutôt le cours donné par Viviane, tandis que les Sections 2, 4, 5 sont relatives au cours de Brigitte. Ces notes résument en quelque sorte l'activité du groupe d'Analyse dynamique entre 1995 et ce jour. Brigitte Vallée tient à remercier tous ceux qui ont contribué à ce travail : en tout premier lieu, Philippe Flajolet, mais aussi tous ceux qui font partie ou ont, à un moment ou un autre, fait partie du groupe caennais : (par ordre alphabétique) Ali Akhavi, Jérémie Bourdon, Julien Clément, Benoît Daireaux, Hervé Daudé, Julien Fayolle, Charlie Lemée, Loïck Lhote. Un grand merci à Jérémie Bourdon pour le prêt des figures tirées de son mémoire de thèse ..., aux relecteurs attentifs de ce texte et tout particulièrement à l'éditeur de ce volume.

1. **Systèmes dynamiques**

Ici, on donne la définition des systèmes dynamiques et on insiste sur leurs principales caractéristiques. Le lecteur intéressé à la problématique générale des systèmes dynamiques pourra consulter le livre [4]. Les livres [10, 34] constituent une très bonne introduction élémentaire aux systèmes dynamiques de l'intervalle.

1.1. **Système dynamique.** Un système dynamique (de l'intervalle) est défini par les éléments suivants (voir un exemple Figure 2) :

1. un alphabet $\mathcal{M}$ inclus dans $\mathbb{N}$, fini ou dénombrable.
2. une partition topologique de $I :=]\,0,1\,[$ en intervalles ouverts disjoints I_m, pour $m \in \mathcal{M}$, *i. e.* $\bar{I} = \bigcup_{m\in\mathcal{M}} \bar{I}_m$.
3. une application de codage σ, constante et égale à m sur chaque I_m.
4. une application de décalage $T : I \to I$ inversible et de classe $\mathcal{C}^2$ sur chaque I_m. On désigne par $J_m = TI_m$ l'image par T de l'intervalle I_m, par $h_m : J_m \to I_m$ l'inverse local (appelé encore branche inverse) de T restreint à I_m, et par $\mathcal{H}$ l'ensemble $\mathcal{H} := \{\, h_m \mid m \in \mathcal{M} \,\}$ des branches inverses de T.

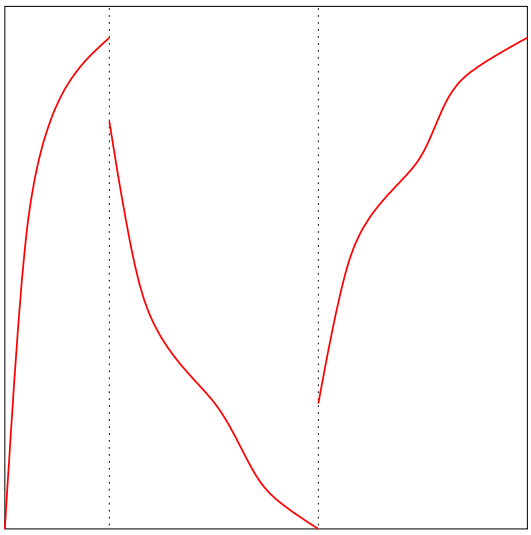


FIGURE 2. Exemple de source dynamique avec un alphabet $\mathcal{M}$ de cardinal 3.

Il y a plusieurs caractéristiques importantes d'un système dynamique, liées en particulier à la régularité des branches h_m, à leur géométrie (c'est-à-dire à la position des intervalles J_m par rapport aux intervalles I_m), au nombre de branches, fini ou infini, aux propriétés d'expansion du système (le décalage T sera dit expansif s'il existe $\Delta > 1$ pour lequel $\left|T'(x)\right| \geq \Delta > 1$).

La trajectoire (ou l'orbite) d'un élément $x \in I$ est la suite :

$$\mathcal{T}(x) := (x, Tx, \dots, T^k x, \dots).$$

Si on utilise l'application de codage σ, on peut associer au réel x le mot infini $M(x)$ construit sur l'alphabet $\mathcal{M}$,

$$M(x) = \big(\sigma(x), \sigma(Tx), \dots, \sigma(T^k x), \dots\big).$$

On pourra se reporter à la Figure 3 pour un exemple de ces deux notions.

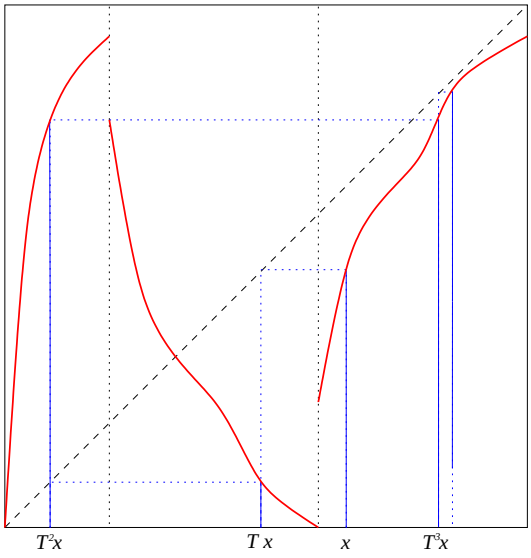


FIGURE 3. Une orbite créée par une source dynamique et le mot émis associé *cbac*...

1.2. **Utilisation en algorithmique.** En algorithmique, les systèmes dynamiques interviennent naturellement dans deux types de contextes : les algorithmes arithmétiques et les algorithmes du texte.

Les algorithmes arithmétiques. Un certain nombre d'algorithmes de type « algorithmes d'Euclide » suivent le schéma suivant.

Entrée : $x \in I$
Tant que $x \notin \mathcal{F}$ faire $x := T(x)$
Renvoyer x

Ici, $\mathcal{F}$ désigne l'ensemble des états finaux de l'algorithme. La trace d'une exécution de l'algorithme sur l'entrée x est alors la trajectoire tronquée $\widetilde{\mathcal{T}}(x)$ qui s'arrête dès que x entre dans $\mathcal{F}$. Le système associé à la transformation T (qu'on appelle le système sous-jacent à l'algorithme) peut être très varié. Pour cette classe d'algorithmes, le système dynamique de référence est associé à la transformation T défini par

$$T(x) := \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$$

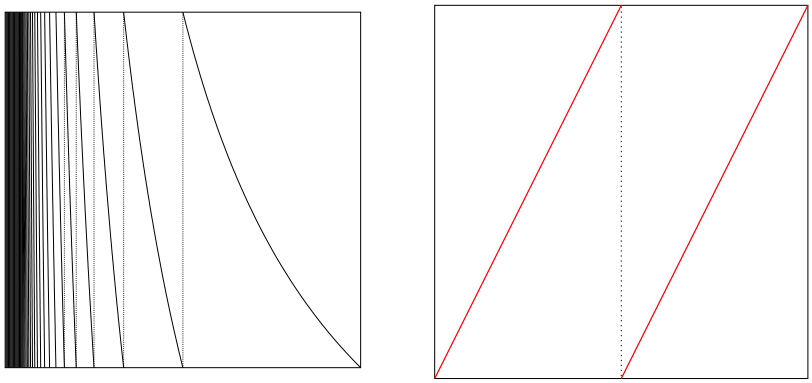


FIGURE 4. Les deux systèmes dynamiques de référence.

(voir Figure 4 gauche), mais la Section 5 donnera des exemples d'algorithmes « naturels » qui font intervenir des systèmes dynamiques assez complexes.

Les algorithmes du texte. Le système dynamique intervient ici fortement car c'est lui qui produit le texte. Plus précisément, on considère le modèle probabiliste suivant : on se donne une densité sur I et on étudie l'ensemble des mots de $\mathcal{M}^{\mathbf{N}}$ de la forme

$$M(x) = \big(\sigma(x), \sigma(Tx), \dots, \sigma(T^k x), \dots\big)$$

lorsque $x \in I$ est choisi suivant la densité f. Le système dynamique de référence (voir Figure 4 droite) est alors associé à la transformation T définie par

$$T(x) := 2x - \lfloor 2x \rfloor$$

qui donne lieu aux suites de chiffres binaires indépendants et équiprobables. La Section 4 donnera des exemples d'analyse d'agorithme de texte, quand le texte est produit par une source dynamique.

1.3. **Première caractéristique des systèmes dynamiques : la géométrie des branches.** La géométrie du système décrit la position des intervalles $J_m := TI_m$ par rapport aux intervalles I_m. Elle permet de caractériser l'ensemble $\mathcal{S}_m$ successeur du symbole m, formé de tous les symboles qui peuvent être émis après le symbole m. La géométrie du système donne ainsi un premier accès à la corrélation entre les symboles successifs émis.

Système complet. On dira que le système est *complet* si pour tout $m \in \mathcal{M}$, l'intervalle J_m est l'intervalle I tout entier. Tous les symboles de l'alphabet $\mathcal{M}$ sont possiblement émis après tout symbole m et donc $\mathcal{S}_m = \mathcal{M}$ pour tout symbole m. Ces systèmes-là sont (dans un sens à préciser) les moins corrélés.

Système markovien. Pour ces systèmes, l'ensemble $\mathcal{S}_m$ des symboles émis après un symbole m ne dépend que de m, et non de ce qui s'est passé avant. Par définition, et dans le cas d'un alphabet fini, on dit qu'un système est *markovien* si tout intervalle $J_m := TI_m$ est réunion finie d'intervalles I_ℓ. Plus précisément, pour tout $m \in \mathcal{M}$, il existe un sous ensemble $\mathcal{L}_m \subset \mathcal{M}$ tel que

$$J_m = \bigcup_{\ell \in \mathcal{L}_m} I_\ell,$$

et dans ce cas, on a $\mathcal{S}_m = \mathcal{L}_m$. La Figure 5 donne un exemple où

$$J_1 = I_1 \cup I_2, \qquad J_2 = I_2 \cup I_3, \qquad J_3 = I.$$

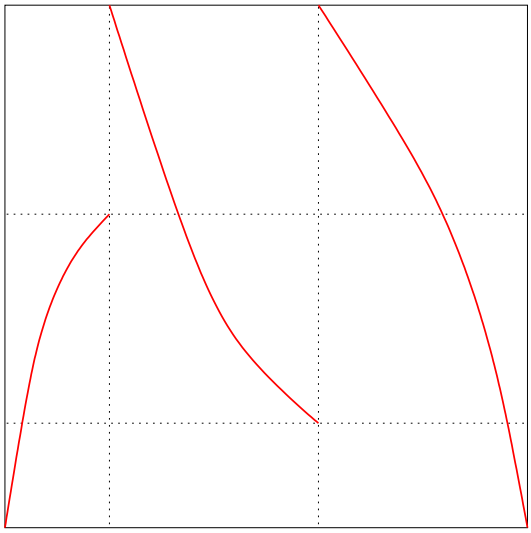


FIGURE 5. Une source dynamique markovienne.

Dans le cas d'un alphabet infini, il faut être un peu plus précis. On dit qu'un système est markovien s'il existe une partition finie de I en intervalles K_ℓ ($\ell \in \mathcal{L}$ et $\mathcal{L}$ finie) telle que

1. tout intervalle J_m est réunion (nécessairement finie) d'intervalles K_ℓ, pour $\ell \in \mathcal{L}_m$;
2. tout intervalle K_ℓ est réunion (en général non finie) d'intervalles I_m, pour $m \in \mathcal{M}_\ell$.

Un élément ℓ de $\mathcal{L}$ joue un rôle similaire à celui d'un état dans une chaîne de Markov. Pour deux états k et ℓ, on désigne par $\mathcal{M}_{k|\ell}$ l'ensemble des symboles de $\mathcal{M}$ qui permettent de passer de l'état ℓ à l'état k,

$$\mathcal{M}_{k|\ell} := \{\, m \in \mathcal{M} \mid I_m \subset K_\ell \quad \text{et} \quad K_k \subset J_m \,\} = \{\, m \in \mathcal{M} \mid m \in \mathcal{M}_\ell \quad \text{et} \quad k \in \mathcal{L}_m \,\}.$$

La matrice sous-jacente au système dynamique est la matrice booléenne P dont le coefficient $p_{k,\ell}$ est défini par

$$p_{k,\ell} = 1 \qquad \text{si et seulement si} \qquad \mathcal{M}_{k|\ell} \neq \emptyset. \tag{1}$$

Elle décrit les transitions possibles entre symboles, et le cas particulier où P est une matrice irréductible est important, puisqu'il traduit une propriété de mélange entre les symboles. (Une matrice irréductible est une matrice dont tous les coefficients sont positifs et qui possède une puissance dont tous les coefficients sont strictement positifs.)

Parfois, la partition de départ $(I_m)_{m\in\mathcal{M}}$ ne donne pas lieu à un système markovien, mais il se peut qu'un raffinement de la partition y donne lieu. La définition plus générale d'un système markovien est finalement la suivante : on construit, à partir de l'ensemble $\mathcal{S}$ des extrémités des intervalles I_m de la partition initiale, les ensembles

$$\mathcal{S}^{[p]} := \bigcup_{i=1}^{p} T^i(\mathcal{S}) \ ; \tag{2}$$

le système est markovien si la suite des $\mathcal{S}^{[p]}$ débute par un premier terme $\mathcal{S}^{[1]}$ fini et est stationnaire.

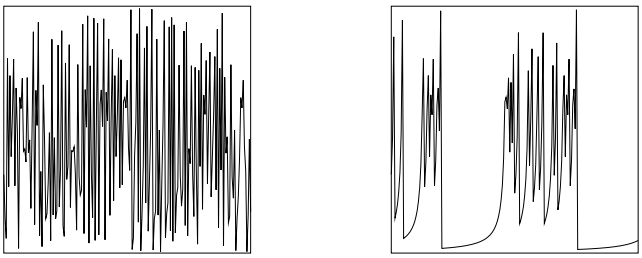


FIGURE 6. À gauche, orbite chaotique ; à droite, orbite avec intermittence.

Système non markovien. Dans ce cas, les symboles qui peuvent être émis à un moment donné ne peuvent être caractérisés en ne considérant qu'une partie bornée de l'histoire précédente : ce sont les systèmes les plus complexes.

1.4. **Importance du caractère expansif.** Rappelons qu'un système est *expansif* si le nombre $\Delta := \inf \left|T'(x)\right|$ est strictement plus grand que 1. La grandeur $\delta := 1/\Delta$ est le coefficient de contraction des branches inverses, et toute branche inverse h de T vérifie $\left|h'(x)\right| \leq \delta$. À première vue, le caractère expansif du décalage (ou, de manière équivalente, le caractère contractant des branches inverses) n'apparaît pas essentiel. Pour se persuader de l'importance de ce facteur, il suffit de comparer le comportement des orbites de deux systèmes : l'un est associé à un décalage T pour lequel T^2 est expansif ; l'autre est « presque » expansif, puisqu'il existe un point fixe indifférent x_0 (*i. e.* un point x_0 pour lequel $T(x_0) = x_0$, $\left|T'(x_0)\right| = 1$), alors que tous les autres points vérifient $\left|T'(x)\right| > 1$ (voir Figure 6). Dans le premier cas, la trajectoire est chaotique ; dans l'autre, elle présente des phénomènes d'intermittence, et quand la trajectoire s'approche de ce point fixe indifférent, elle s'en éloigne à grand peine ... Ces deux systèmes créeront une algorithmique vraiment différente, le premier donnant lieu à un algorithme rapide, et le second, qui perd beaucoup de temps près de son point fixe, donnant lieu à un algorithme lent. Nous reviendrons à cette situation dans les paragraphes 3.4 et 5.6.

2. Le principal outil de l'analyse dynamique : l'opérateur de transfert et sa descendance

Ici, on définit les principaux opérateurs qui sont les outils privilégiés de l'analyse dynamique. Ils proviennent tous de l'opérateur transformateur de densité, qui est leur ancêtre commun.

2.1. **Opérateur transformateur de densité.** Nous venons de décrire comment la possibilité d'émettre à un instant donné tel ou tel symbole était liée à la géométrie du système. Maintenant, nous nous posons une question plus fine : avec quelle probabilité un symbole — s'il peut être émis — va-t-il être émis ? Cette question est très liée à la manière dont le décalage T déforme les mesures sur l'intervalle I. Plus précisément, la densité de probabilité sur I évolue lorsqu'on itère la transformation de décalage T, et c'est l'opérateur *transformateur de densité*, désigné par $\mathbf{H}$, qui quantifie ce phénomène. Pour une densité initiale f, on désigne par $\mathbf{H}[f]$ la densité après une itération de T. On a ainsi :

$$\mathbf{H}[f](x) = \sum_{m\in\mathcal{M}} \left|h'_m(x)\right| f \circ h_m(x)\, 1_{J_m}(x), \tag{3}$$

où 1_A représente la fonction indicatrice de l'ensemble A. Informellement, si f est la densité initiale, la densité en un point x, après une itération, est apportée par tous les antécédents possibles de x.

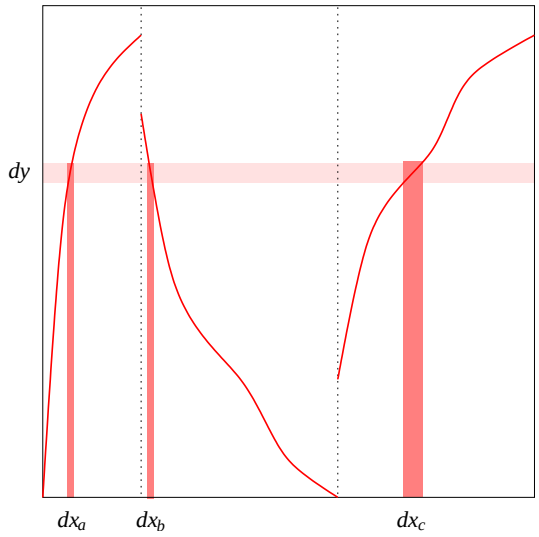


FIGURE 7. L'évolution de la densité.

L'antécédent de x provenant de la branche d'indice m existe si x appartient à J_m, et dans ce cas, il apporte la densité $f \circ h_m(x)$ distordue par le terme $\big|h'_m(x)\big|$ (lié à la formule de changement de variable). La composante $\mathbf{H}_{[m]}$ de l'opérateur relative au symbole m

$$\mathbf{H}_{[m]}[f](t) := \big|h'_m(t)\big|\, f \circ h_m(t)\, 1_{J_m}(t)$$

désigne ainsi la contribution apportée par la branche d'indice m (voir Figure 7).

C'est cette distorsion possible par le facteur $\big|h'_m(x)\big|$ qui va constituer le deuxième facteur de corrélation. Si les branches sont affines, avec donc une dérivée constante, cette distorsion n'existera pas. Pour une géométrie de branches fixée, ce sont donc les systèmes dynamiques à branches affines qui seront les moins corrélés. À l'opposé, ceux dont les branches ont une dérivée seconde grande (en valeur absolue) donneront lieu à des sources fortement corrélées. En particulier, c'est plutôt la dérivée de $x \mapsto \log\big|h'(x)\big|$ qui va intervenir, et la *condition de distorsion bornée*,

$$\exists c > 0,\ \forall x \in I,\ \forall h \in \mathcal{H}, \qquad \big|h''(x)\big| \le c\big|h'(x)\big|, \tag{4}$$

toujours vérifiée lorsque le nombre de branches est fini, intervient de manière fréquente.

Le k-ième itéré de l'opérateur $\mathbf{H}$ a aussi une forme très simple ; grâce à la propriété de multiplicativité des dérivées de fonctions composées, il s'exprime comme une somme qui fait intervenir tous les mots w de $\mathcal{M}^k$,

$$\mathbf{H}^k[f](x) = \sum_{w \in \mathcal{M}^k} |h'_w(x)|\, f \circ h_w(x)\, 1_{J_w}(x). \tag{5}$$

Ici, pour un mot w de $\mathcal{M}^k$ de la forme $w := m_1 m_2 \dots m_k$, la notation h_w désigne la *branche inverse* de T^k de la forme $h_w := h_{m_1} \circ \cdots \circ h_{m_k} \in \mathcal{H}^k$ et J_w désigne l'intervalle de définition de la branche h_w.

Cas particulier des systèmes complets et markoviens. Comme nous le verrons plus loin, la présence des fonctions indicatrices apporte un certain nombre de complications. Le cas le plus simple est donc celui des systèmes complets où ces fonctions indicatrices n'existent pas.

Dans le cas d'un système markovien, quitte à travailler avec une matrice d'opérateurs, on peut faire « disparaître » ces fonctions indicatrices, en procédant comme suit : à une fonction f définie

sur I, on associe la suite (finie) des fonctions f_ℓ, où f_ℓ est la restriction de f à l'intervalle K_ℓ. Au lieu de faire agir l'opérateur $\mathbf{H}$ sur f, et de considérer le transformé $g := \mathbf{H}[f]$, on considère qu'il agit sur la suite $\tilde{f}$ des f_ℓ et on désigne par g_k la k-ième composante de $\tilde{g}$ (*i. e.* la restriction de g à K_k) On a clairement

$$g_k = \sum_{\ell\in\mathcal{L}} \sum_{m\in\mathcal{M}_{k|\ell}} \mathbf{H}_{[m]}[f_\ell],$$

de sorte que $\mathbf{H}$ est maintenant (à conjugaison près) une matrice d'opérateurs, désignée par $\tilde{\mathbf{H}}$, de dimension $|\mathcal{L}| \times |\mathcal{L}|$ dont le coefficient situé en position (k,ℓ) est l'opérateur

$$\tilde{\mathbf{H}}_{k,\ell} := \mathbf{H}_{[k|\ell]} = \sum_{m\in\mathcal{M}_{k|\ell}} \mathbf{H}_{[m]} \ ;$$

En remplaçant ainsi l'égalité $g := \mathbf{H}[f]$ par l'égalité $\tilde{g} := \tilde{\mathbf{H}}[\tilde{f}]$, on a supprimé toutes les fonctions indicatrices ...

2.2. **Opérateur transformateur de densité, intervalles fondamentaux et probabilités fondamentales.** Si w est un mot fini, on désigne par p_w la probabilité qu'un mot produit par la source commence par w.

Associons à un mot w de longueur finie k la branche inverse h_w ; l'intervalle $h_w(I)$ est alors l'ensemble des réels x pour lesquels le mot $M(x)$ débute par le préfixe w : c'est ce que nous appelons *l'intervalle fondamental* associé au mot w, et que nous désignons par I_w ; pour un mot réduit à un symbole m, c'est exactement l'intervalle I_m de la partition initiale. Considérons une densité de probabilité f sur l'intervalle I. La mesure de l'intervalle $I_w = h_w(I)$ est exactement la probabilité p_w et

$$p_w := \int_{h_w(I)} f(t)\,dt = \int_I |h'_w(t)|\, f\circ h_w(t)\, 1_{J_w}(t)\,dt.$$

La composante de l'opérateur $\mathbf{H}^k$ relatif à la branche h_w, désignée par $\mathbf{H}_{[w]}$ et définie par

$$\mathbf{H}_{[w]}[f](t) := \left|h'_w(t)\right| f\circ h_w(t)\, 1_{J_w}(t) \tag{6}$$

permet donc d'exprimer la probabilité p_w, via la relation

$$p_w = \int_I \mathbf{H}_{[w]}[f](t)\,dt, \tag{7}$$

de sorte que cet opérateur $\mathbf{H}_{[w]}$ peut être considéré comme l'opérateur « générateur » de la probabilité p_w. De plus, la concaténation ww' entre deux mots se traduit par la *propriété de composition*

$$\mathbf{H}_{[ww']} = \mathbf{H}_{[w']} \circ \mathbf{H}_{[w]}, \tag{8}$$

qui est essentielle car elle permet de généraliser la propriété multiplicative

$$p_{ww'} = p_w p_{w'}$$

qui n'est vérifiée que par les sources sans mémoire.

2.3. **Sources classiques simples : sources sans mémoire, chaînes de Markov.** Pour une géométrie donnée, les systèmes dynamiques les plus simples sont ceux dont les branches sont affines.

Une source sans mémoire est modélisée par un système dynamique complet à branches affines, initialisé avec la densité uniforme. La Figure 8 donne un exemple de modélisation possible d'une source sans mémoire qui produit trois symboles suivant les probabilités $1/2, 1/6, 1/3$.

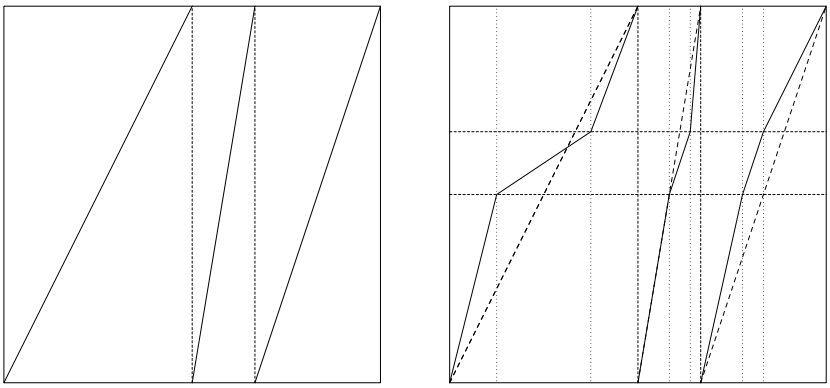


FIGURE 8. Une source sans mémoire, une chaîne de Markov.

Une chaîne de Markov est modélisée par un système dynamique markovien à branches affines, initialisé avec une densité constante sur chaque K_ℓ. La Figure 8 montre un exemple de modélisation d'une chaîne de Markov d'ordre 1.

Une chaîne de Markov d'ordre k s'obtient en cassant en morceaux les branches (affines) d'une chaîne de Markov d'ordre $k-1$. La Figure 8 montre comment on peut passer du cas $k=0$ au cas $k=1$. C'est pour cela, que, informellement du moins, un système général markovien peut être considéré comme une limite de chaînes de Markov d'ordre de plus en plus élevé.

2.4. **L'opérateur de transfert.** Dans l'étude des systèmes dynamiques, il est très utile de généraliser l'opérateur transformateur de densité $\mathbf{H}$ (défini en (3) en lui adjoignant un paramètre s. On obtient alors *l'opérateur de transfert*, désigné par $\mathbf{H}_s$ et défini par

$$\mathbf{H}_s[f](x) = \sum_{m\in\mathcal{M}} \left|h'_m(x)\right|^s f\circ h_m(x)\, 1_{J_m}(x). \tag{9}$$

Ici, l'ajout du paramètre s permettra de relier cet opérateur à des séries génératrices et plus précisément à des séries génératrices de Dirichlet.

Comme en (5), le k-ième itéré de l'opérateur $\mathbf{H}_s$ a aussi une forme très simple, et s'exprime comme une somme qui fait intervenir tous les mots w de $\mathcal{M}^k$,

$$\mathbf{H}_s^k[f](x) = \sum_{w\in\mathcal{M}^k} \left|h'_w(x)\right|^s f\circ h_w(x)\, 1_{J_w}(x). \tag{10}$$

Nous aurons besoin des composantes de tels opérateurs, et nous désignerons par $\mathbf{H}_{s,[w]}$ l'opérateur associé à la branche h_w et défini par

$$\mathbf{H}_{s,[w]}[f](t) := \left|h'_w(t)\right|^s f\circ h_w(t)\, 1_{J_w}(t).$$

Remarquons cependant que cet opérateur, qui vérifie une propriété de composition analogue à (8),

$$\mathbf{H}_{s,[ww']} = \mathbf{H}_{s,[w']}\circ\mathbf{H}_{s,[w]} \tag{11}$$

ne permet pas d'exprimer simplement la quantité p_w^s.

L'opérateur qui fait intervenir l'ensemble $\mathcal{M}^\star$ de tous les mots (finis) produits par la source est alors la somme de tous les itérés k-ième de l'opérateur définis en (10) : c'est ce que nous appelons le quasi-inverse ou l'étoile,

$$(\mathbf{1}-\mathbf{H}_s)^{-1} := \sum_{k\geq 0} \mathbf{H}_s^k, \tag{12}$$

et qui jouera un rôle si important dans la suite ...

2.5. **Pondération de l'opérateur de transfert.** Dans les applications aux algorithmes (et tout particulièrement aux algorithmes arithmétiques), on désire souvent pondérer chaque branche du décalage par une quantité qui mesure le coût de l'algorithme associé quand l'exécution « passe par » la branche. Ce coût peut dépendre de manière assez variée de la branche, mais, très souvent, comme nous le verrons dans les applications, ce coût est « additif », et le coût total d'une exécution est la somme des coûts dûs à l'emprunt de chaque branche. On remplace alors chaque opérateur composant $\mathbf{H}_{s,[w]}$ par un opérateur pondéré par un coût c,

$$\mathbf{H}^{[c]}_{s,u,[w]} := u^{c(h_w)}\, \mathbf{H}_{s,[w]},$$

et l'additivité des coûts montre que la propriété de composition se prolonge aux opérateurs pondérés.

2.6. **Opérateur de transfert généralisé.** Il est nécessaire ici de considérer des sources dynamiques complètes ou markoviennes. Commençons par le cas complet. Les quantités p^s_w s'expriment alors en fonction de l'opérateur de transfert généralisé, appelé encore opérateur sécant. Si F désigne la fonction de répartition de f, la quantité p^s_w s'exprime comme

$$p^s_w = \big|F\circ h_w(0) - F\circ h_w(1)\big|^s,$$

et fait donc intervenir la valeur de la fonction $F\circ h_w$ en les deux points $x=0$ et $x=1$. C'est pourquoi on introduit un opérateur de transfert $\mathfrak{H}_{s,[w]}$ qui agit sur des fonctions de deux variables en utilisant la « sécante » de la branche h_w (d'où son nom d'opérateur sécant)

$$\mathfrak{H}_{s,[w]}[\Phi](u,v) := \left|\frac{h_w(u)-h_w(v)}{u-v}\right|^s \Phi\big(h_w(u),h_w(v)\big),$$

ce qui résout le problème puisque

$$p^s_w = \mathfrak{H}_{s,[w]}[L^s](0,1) \qquad \text{avec} \qquad L(x,y) = \left|\frac{F(x)-F(y)}{x-y}\right|. \tag{13}$$

La multiplicativité de la « sécante » permet de prouver la propriété de composition

$$\mathfrak{H}_{s,[ww']} = \mathfrak{H}_{s,[w']}\circ \mathfrak{H}_{s,[w]},$$

qui, comme en (8) généralise la relation $p^s_{ww'} = p^s_w p^s_{w'}$.

Les opérateurs qui généralisent respectivement $\mathbf{H}_s$, ses itérés $\mathbf{H}^k_s$ et son quasi-inverse $(\mathbf{1}-\mathbf{H}_s)^{-1}$ sont alors les opérateurs $\mathfrak{H}_s$, $\mathfrak{H}^k_s$ et $(\mathbf{1}-\mathfrak{H}_s)^{-1}$ définis par

$$\mathfrak{H}_s := \sum_{m\in\mathcal{M}} \mathfrak{H}_{s,[m]}, \qquad \mathfrak{H}^k_s = \sum_{w\in\mathcal{M}^k} \mathfrak{H}_{s,[w]}, \qquad (\mathbf{1}-\mathfrak{H}_s)^{-1} = \sum_{w\in\mathcal{M}^\star} \mathfrak{H}_{s,[w]}. \tag{14}$$

Ce formalisme peut se transporter aisément dans le cas d'une source markovienne : la matrice $\mathfrak{H}_s$ a pour coefficient l'opérateur $\mathfrak{H}_{s,[k|\ell]}$.

2.7. **Problèmes à longueur fixée, ou à longueur quelconque.** Comme le montrent les relations (10), (12) et (14), les k-ième itérés des opérateurs font intervenir l'ensemble $\mathcal{M}^k$ des mots de longueur k et les quasi-inverses l'ensemble $\mathcal{M}^\star$ de tous les mots finis. Si on travaille sur des problèmes à taille fixée (longueur des textes fixée pour les algorithmes de texte, nombre d'itérations fixé pour les algorithmes arithmétiques), c'est donc le comportement asymptotique de ces k-ième itérés qu'on utilisera (pour $k\to\infty$). Si le problème fait intervenir toutes les tailles possibles, les opérateurs adéquats seront les opérateurs quasi-inverses, et on s'intéressera à leurs singularités.

Pour une matrice M, le comportement asymptotique de M^k ou les singularités de $(\mathrm{Id}-M)^{-1}$ sont très liés aux propriétés spectrales de la matrice M, et en particulier aux propriétés spectrales

dominantes (correspondant aux valeurs propres ayant le plus grand module). Nous sommes donc conduits à étudier l'analogue, mais en dimension infinie.

3. **Analyse fonctionnelle et propriétes spectrales**

Cette section est dédiée à l'étude des propriétés spectrales des opérateurs de transfert. Un livre de référence est celui de V. Baladi [2].

Pour un opérateur $\mathbf{L}$ qui agit sur un espace de Banach $\mathcal{F}$, le spectre $\operatorname{Sp}\mathbf{L}$ de $\mathbf{L}$ est l'ensemble des nombres complexes z pour lesquels $\mathbf{L} - z\mathbf{1} : \mathcal{F} \to \mathcal{F}$ n'est pas inversible. Un élément z de $\operatorname{Sp}\mathbf{L}$ est une valeur propre si $\mathbf{L} - z\mathbf{1}$ n'est pas injective. En dimension finie, le spectre d'une matrice est l'ensemble de ses valeurs propres. L'espace sur lequel agit l'opérateur est fondamental car le spectre d'un opérateur dépend beaucoup de l'espace sur lequel il opère. (Plus l'espace est « gros », plus il contient de possibles fonctions propres, et plus le spectre est lui-même « gros ».) Ainsi, un opérateur peut avoir de « bonnes » propriétés spectrales sur un espace et de moins bonnes sur un autre. Le choix de cet espace est fondamental et constitue généralement un des points délicats de l'analyse.

3.1. **Critères de choix pour l'espace fonctionnel.** Ce choix résulte en général d'un compromis : On veut que l'espace fonctionnel $\mathcal{F}$ soit suffisamment « gros » pour que l'opérateur de transfert $\mathbf{H}_s$ opère sur $\mathcal{F}$ (*i. e.* $\mathbf{H}_s[\mathcal{F}] \subset \mathcal{F}$). Mais on veut aussi qu'il ne soit pas trop gros pour que le spectre reste discret (formé de points isolés), ou du moins que la partie « supérieure » du spectre reste discrète.

Ce choix va dépendre des caractéristiques du système dynamique. Il sera dicté en tout premier lieu par la géométrie du système, et modulé par la régularité des branches. Dans la formule (9) apparaissent les fonctions caractéristiques 1_{J_m}. En fonction de la géométrie du système, ces fonctions caractéristiques peuvent introduire des discontinuités, et $\mathbf{H}_s[f]$ peut être discontinue même si f est très régulière.

1. Si le système est complet, les opérateurs $\mathbf{H}_s$ n'introduisent pas de discontinuités et on peut travailler sur des espaces de fonctions régulières (fonctions C^r sur I, fonctions analytiques, etc.) adaptés à la régularité des branches h_m.
2. Si le système est markovien, les opérateurs $\mathbf{H}_s^p$ introduisent des discontinuités uniquement au bord des K_ℓ et on peut travailler sur des espaces de fonctions régulières sur chacun des K_ℓ, ayant donc un nombre fini de discontinuités.
3. Enfin, si le système n'est pas markovien, on introduit à chaque itération de nouvelles discontinuités, de sorte que l'ensemble des discontinuités introduites est dénombrable et peut être dense dans I. On est alors conduit à travailler sur l'espace des fonctions à variation bornée.

3.2. **Le bon comportement désiré.** On considère d'abord le cas où $s = 1$. L'opérateur étudié est donc le transformateur de densité $\mathbf{H}$.

Sur un espace fonctionnel adéquat $\mathcal{F}$, les propriétés

(P1) la valeur 1 est valeur propre simple dominante unique de $\mathbf{H}$,

(P2) il y a un saut spectral : le reste du spectre de $\mathbf{H}$ est contenu dans un disque de rayon strictement inférieur à 1,

entrainent un certain nombre de conséquences. Tout d'abord, il existe alors un disque Γ du plan complexe, de frontière γ, qui contient comme seul point du spectre la valeur 1. De plus,

l'opérateur $\mathbf{P}$ défini par

$$\mathbf{P} := \frac{1}{2i\pi} \int_\gamma (z\mathbf{1} - \mathbf{H})^{-1}\, dz$$

est le projecteur sur le sous-espace propre dominant, et l'opérateur $\mathbf{H}$ se décompose en $\mathbf{H} = \mathbf{P} + \mathbf{N}$ où $\mathbf{N}$ est un opérateur dont le spectre est le même que celui de $\mathbf{H}$, excepté la valeur 1. Le rayon spectral de $\mathbf{N}$ est ainsi strictement inférieur à 1. Enfin, on a aussi $\mathbf{H}^k = \mathbf{P} + \mathbf{N}^k$ de sorte que

$$(\mathbf{1} - z\mathbf{H})^{-1} = \frac{\mathbf{P}}{1-z} + \mathbf{R}(z), \tag{15}$$

avec une fonction reste $\mathbf{R}$,

$$\mathbf{R}(z) := (\mathbf{1} - z\mathbf{N})^{-1} - \mathbf{P} = \sum_{k\geq 1} z^k (\mathbf{H}^k - \mathbf{P})$$

qui décrit les corrélations du système dynamique. De plus, le projecteur $\mathbf{P}$ s'exprime en fonction de la fonction propre dominante ϕ, normalisée par $\int_I \phi(u)\, du = 1$, sous la forme :

$$\mathbf{P}[f](t) = \phi(t) \int_I f(u)\, du.$$

Si, de plus, la condition (P3) suivante est satisfaite,

(P3) l'application $s \mapsto \mathbf{H}_s$ est analytique sur un voisinage de $s = 1$,

la théorie de la perturbation s'applique alors [32] et montre l'existence de fonctions $s \mapsto \lambda(s)$, $s \mapsto \mathbf{P}_s$, $s \mapsto \mathbf{N}_s$ analytiques dans un voisinage de $s = 1$. Ici, $\lambda(s)$ est la valeur propre dominante de $\mathbf{H}_s$, $\mathbf{P}_s$ est le projecteur sur le sous-espace propre dominant et $\mathbf{N}_s$ est un opérateur dont le rayon spectral est strictement inférieur à $\big|\lambda(s)\big|$. La décomposition $\mathbf{H}_s^k = \lambda(s)^k \mathbf{P}_s + \mathbf{N}_s^k$ perdure et finalement, la décomposition spectrale

$$(\mathbf{1} - \mathbf{H}_s)^{-1} = \frac{\mathbf{P}_s}{1-\lambda(s)} + \mathbf{N}_s(\mathbf{1} - \mathbf{N}_s)^{-1} \tag{16}$$

montre que $(\mathbf{1} - \mathbf{H}_s)^{-1}$ possède un pôle d'ordre 1 en $s = 1$, dont le résidu est $-\lambda'(1)\mathbf{P}$. Cette dernière valeur $-\lambda'(1)$ est l'entropie du système dynamique, comme nous le verrons plus loin.

3.3. **Compacité et quasi-compacité.** La propriété (P1) est une propriété de type Perron–Frobenius : elle est liée à des propriétes de forte positivité. Rappelons que la propriété (P1) est vérifiée pour une matrice M stochastique qui a une puissance k-ième dont tous les coefficients sont strictement positifs.

La propriété (P2) est toujours vraie en dimension finie, car le spectre est alors fini. Plus généralement, la validité de (P2) est assurée aussitôt que le spectre de $\mathbf{H}$ est discret, ou, du moins, aussitôt que la partie « supérieure » du spectre est discret.

Les opérateurs *compacts* sont les opérateurs qui, en dimension infinie, ressemblent le plus aux opérateurs de la dimension finie. Leur spectre est discret à ceci près qu'un point d'accumulation est possible en 0, et la validité de (P2) est alors assurée. Mais, on ne peut pas toujours trouver un espace fonctionnel $\mathcal{F}$ sur lequel l'opérateur $\mathbf{H}$ soit compact, et l'on ne peut donc toujours assurer que la totalité du spectre soit discret. On considère alors la propriété de quasi-compacité, plus générale. Le rayon spectral $R(\mathbf{L})$ d'un opérateur $\mathbf{L}$ est la borne supérieure des modules des éléments du spectre $\operatorname{Sp}\mathbf{L}$, de sorte que $\operatorname{Sp}\mathbf{L} \subset \{\, \lambda \;\big|\; |\lambda| \leq R(\mathbf{L}) \,\}$. Le rayon spectral essentiel $R_e(\mathbf{L})$ d'un opérateur $\mathbf{L}$ est le plus petit réel $r > 0$ pour lequel tout élément λ de $\operatorname{Sp}\mathbf{L}$ ayant un module $|\lambda| > r$ est une valeur propre isolée et de multiplicité finie. Pour un opérateur compact, on a $R_e(\mathbf{L}) = 0$. Un opérateur pour lequel $R_e(\mathbf{L}) < R(\mathbf{L})$ est appelé *quasi-compact*. Son spectre se décompose en

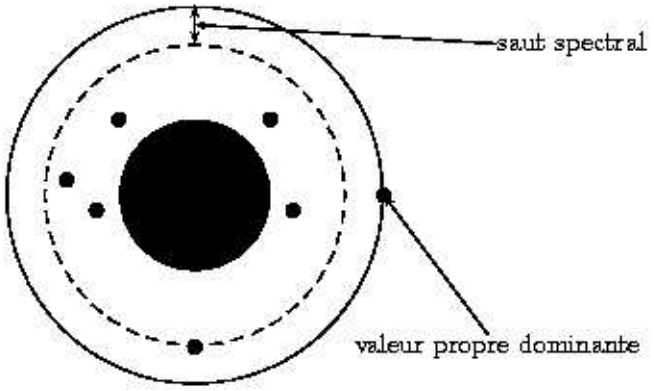


FIGURE 9. Saut spectral.

deux parties, une partie supérieure discrète et une partie inférieure qui peut être quelconque (voir Figure 9).

3.4. **Des espaces fonctionnels adéquats.** L'espace fonctionnel où les propriétés (P1), (P2) et (P3) sont vérifiées dépend des caractéristiques du système. Nous donnons ici quelques exemples d'espaces fonctionnels adaptés à certaines classes de systèmes dynamiques.

Type 1 : Systèmes complets (ou markoviens) avec branches uniformément holomorphes et contractantes. Ce sont d'abord les systèmes complets, bien décrits dans [36], qui vérifient ce qui suit :

> Il existe un disque complexe $\mathcal{V}$ sur lequel toutes les branches inverses $h \in \mathcal{H}$ se prolongent en des fonctions holomorphes sur $\mathcal{V}$, envoyant $\mathcal{V}$ strictement dans lui-même, (*i. e.* $h\left(\bar{\mathcal{V}}\right) \subset \mathcal{V}$) et contractantes (*i. e.* $\left|h'(z)\right| \leq \delta_h < 1$ avec la série $\sum_h \delta_h^\alpha$ convergente pour un réel $\alpha < 1$).

Dans ce cas, l'opérateur $\mathbf{H}$ agit sur l'espace $\mathcal{A}_\infty(\mathcal{V})$ des fonctions holomorphes définies sur $\mathcal{V}$ et continues sur $\bar{\mathcal{V}}$. Comme tous les opérateurs composants (qui sont des opérateurs de « composition » de la forme $f \mapsto f \circ h$) y sont compacts, l'opérateur $\mathbf{H}$ y est aussi compact. Un théorème dû à Krasnoselsky [33] généralise les résultats à la Perron–Frobenius et prouve que (P1) est aussi vérifiée ; (P3) est également vérifiée sans problème, par perturbation analytique, dès que $\Re(s) > \alpha$, ce pour un certain $\alpha > 1$.

Si de plus, le système a une distorsion bornée, les propriétés citées ci-dessus se généralisent à l'opérateur $\mathfrak{H}_s$ (voir [13, 43]), à condition de le faire opérer sur l'espace $\mathcal{B}_\infty(\mathcal{V})$ des fonctions holomorphes définies sur $\mathcal{V} \times \mathcal{V}$ et continues sur $\bar{\mathcal{V}} \times \bar{\mathcal{V}}$.

On peut aussi considérer la version « markovienne » du début de la condition précédente (on reprend les notations des paragraphes 1.3 et 2.1) :

> Pour tout k et tout ℓ de $\mathcal{L}$, il existe un disque complexe $\mathcal{V}_k$, voisinage de K_k sur lequel toutes les branches inverses $h \in \mathcal{H}_{[k|\ell]}$ ont leurs restrictions à K_k qui se prolongent en des fonctions holomorphes sur $\mathcal{V}_k$, envoyant $\mathcal{V}_k$ strictement dans $\mathcal{V}_\ell$.

Cette dernière condition assure que chaque opérateur $\mathbf{H}_{[k|\ell]}$ a de bonnes propriétés de compacité et de positivité. Si, de plus, la matrice de transition P définie en (1) est irréductible et apériodique, alors l'opérateur matriciel a toutes les bonnes propriétés souhaitées.

Type 2 : Systèmes à géométrie quelconque, contractants. Cas du nombre de branches fini. Dans ce cas (voir [16]), l'espace fonctionnel adapté est l'espace $BV(I)$ des fonctions à variation bornée sur l'intervalle I. Cet espace est un espace de Banach dense dans $\mathcal{L}^1(I)$ dont la boule unité est précompacte dans $\mathcal{L}^1(I)$. L'opérateur $\mathbf{H}$ agit sur $BV(I)$ et le théorème suivant [29] permet de montrer sa quasi-compacité.

Théorème. *Soit* $\mathbf{L}$ *un opérateur qui agit sur* $\mathcal{L}^1$. *Supposons qu'il existe deux suites* (r_n) *et* (t_n) *de nombres positifs pour lesquelles, pour tout* $n \geq 1$, *et pour tout* $f \in BV(I)$, *on a*

$$\left\|\mathbf{L}^n[f]\right\|_{BV} \leq r_n \|f\|_{BV} + t_n \|f\|_1. \tag{17}$$

Alors l'opérateur $\mathbf{L}$ *est borné sur* $BV(I)$ *et son rayon spectral essentiel vérifie*

$$R_e(\mathbf{L}) \leq r := \liminf_{n\to\infty} (r_n)^{1/n}.$$

On applique le théorème en montrant que r peut être choisi égal au coefficient de contraction $\delta < 1$ et que l'opérateur $\mathbf{H}$ a une valeur propre égale à 1.

Type 3 : Cas du nombre infini de branches. Systèmes à géométrie pseudo-markovienne, contractants, à distorsion bornée. Quand le nombre de branches est infini, ce qui arrive très souvent dans les applications arithmétiques, on peut aussi travailler sur $BV(I)$, à condition d'exiger des propriétés supplémentaires pour le système dynamique. En particulier (voir [7, 12]), on exige que le système ait une distorsion bornée, et aussi qu'il ne soit pas trop différent d'un système markovien. Dans le cas d'un système markovien, l'ensemble $\mathcal{S}^{[p]}$, défini en (2), et formé des extrémités des intervalles J_w associés à l'ensemble $\{\, w \mid |w| \leq p \,\}$ est fini pour tout p. Là, on lui laisse la possibilité d'être infini, mais on exige que les intervalles J_w, quand ils sont non vides, ne soient pas trop petits, *i. e.*

$$\ell_p := \inf \left\{\, |J_w| \;\middle|\; J_w \neq \emptyset,\ |w| \leq p \,\right\} > 0.$$

C'est une condition qui a été donnée au départ par Rychlick. Dans ces conditions, les propriétés (P1), (P2) et (P3) sont vérifiées pour l'opérateur $\mathbf{H}_s$ agissant sur $BV(I)$.

3.5. **La méthode d'induction.** Dans tout le paragraphe précédent, le système était supposé expansif. On peut traiter relativement aisément des systèmes complets où la condition d'expansion est seulement violée en un point, et qui sont « presque expansifs » avec seulement un point indifférent (voir paragraphe 1.4). Dans ce cas, il y a une seule « mauvaise » branche (*i. e.* non expansive), et on va la grouper avec des bonnes branches, pour tenter d'améliorer son comportement. Supposons que cette branche soit la branche correspondant au symbole a, et corresponde donc à un intervalle I_a.

Considérons le système dynamique (J, U) où l'intervalle J est $J := I \setminus I_a$ et le décalage U est défini par le premier retour à J : pour $x \in J$, on désigne par $n(x)$ le plus petit entier pour lequel $T^{n(x)} \in J$, et on pose $U(x) := T^{n(x)}(x)$. Ce système dynamique est appelé le système induit. La partition fondamentale sur J est maintenant formée des intervalles fondamentaux de l'ancien système de la forme

$$I_w \quad \text{avec} \quad w \in \mathcal{N} := \big(\mathcal{M} \setminus \{a\}\big)\{a\}^\star,$$

et le nouvel alphabet $\mathcal{N}$ est ainsi infini.

Il y a une autre manière d'induire, un peu différente, en restant dans l'intervalle I, et en remplaçant la partition initiale par la partition formée des anciens intervalles fondamentaux de la forme

$$I_w \quad \text{avec} \quad w \in \mathcal{Q} := \{a\}^\star\big(\mathcal{M} \setminus \{a\}\big).$$

C'est celle-là qu'on utilisera plutôt en algorithmique, et qui remplace l'alphabet $\mathcal{M}$ initial par l'alphabet $\mathcal{Q}$. La Figure 10 représente un système dynamique (à gauche) et son système dynamique

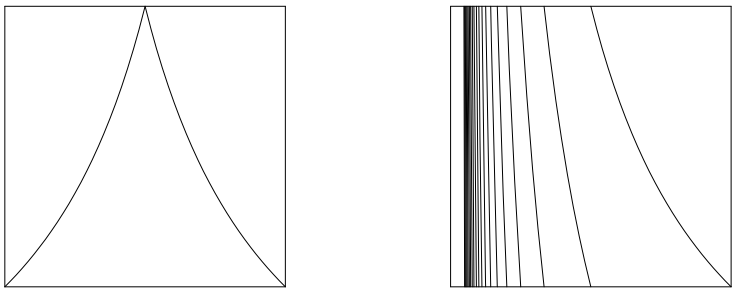


FIGURE 10. Un système dynamique et son système induit.

induit associé (on induit ici par rapport à la première branche, car c'est elle qui possède un point indifférent).

Grâce aux propriétés de dictionnaire dues à la propriété de composition (11), l'opérateur de transfert $\tilde{\mathbf{H}}_s$ du système dynamique induit fait intervenir l'opérateur de transfert $\mathbf{H}_s$ et l'opérateur $\mathbf{A}_s := \mathbf{H}_{s,[a]}$ relatif au symbole a sous la forme

$$\tilde{\mathbf{H}}_s = \sum_{k\geq 0} (\mathbf{H}_s - \mathbf{A}_s)\mathbf{A}_s^k = (\mathbf{H}_s - \mathbf{A}_s)(\mathbf{1} - \mathbf{A}_s)^{-1}. \tag{18}$$

Puisque le nouveau décalage regroupe une suite de « mauvaises » branches avec une « bonne » branche, le nouveau système dynamique sera expansif, et le quasi-inverse $(\mathbf{1} - \tilde{\mathbf{H}}_s)^{-1}$ vérifiera souvent des propriétés de type (16). Alors, la relation $\mathcal{M}^\star = \mathcal{Q}^\star\{a\}^\star$, qui se traduit par une relation entre les deux quasi-inverses,

$$(\mathbf{1} - \mathbf{H}_s)^{-1} = (\mathbf{1} - \mathbf{A}_s)^{-1}(\mathbf{1} - \tilde{\mathbf{H}}_s)^{-1} \tag{19}$$

permet de « revenir » au quasi-inverse initial, en y intégrant les propriétés de la « mauvaise » branche.

4. **Analyse dynamique des algorithmes du texte**

Le comportement de tout algorithme qui travaille sur du texte est très influencé par la manière dont le texte est produit. Il y a d'abord un premier fait qui est vrai pour une source $\mathcal{S}$ quelconque :

1. L'ensemble des probabilités $\{\, p_w \mid w \in \mathcal{M}^\star \,\}$, ou plus généralement, pour un complexe s, l'ensemble des quantités $\{\, p_w^s \mid w \in \mathcal{M}^\star \,\}$ joue un rôle essentiel dans l'analyse des algorithmes du texte, lorsque le texte est produit par une source quelconque $\mathcal{S}$.

L'intérêt des sources dynamiques provient du caractère explicite de ces probabilités, que nous avons décrit dans la Section 2 :

2. Pour une source dynamique, les probabilités p_w s'expriment en fonction des composantes de l'opérateur transformateur de densité (voir Section 2.2).
3. Pour une source dynamique complète (ou markovienne), les quantités p_w^s s'expriment en fonction de l'opérateur de transfert généralisé (voir l'opérateur sécant de la Section 2.6).

Nous allons maintenant décrire quelques exemples d'application de ces trois faits.

4.1. **Les problèmes de mots qui font intervenir des langages.** Un langage $\mathcal{L}$ défini sur l'alphabet $\mathcal{M}$ est un sous-ensemble de $\mathcal{M}^\star$. À un langage $\mathcal{L}$, on associe classiquement la série génératrice

$$L(z) := \sum_{w\in\mathcal{L}} p_w z^{|w|} \tag{20}$$

où la variable z « marque » la taille $|w|$ du mot w. Cette série génératrice s'avère essentielle dans l'analyse des proriétés du langage $\mathcal{L}$.

Pour une source sans mémoire, la propriété de multiplicativité des probabilités permet de traduire les opérations sur les langages en opérations sur les séries génératrices associées. Ce n'est plus possible dès que la source garde « de la mémoire ». On remplace alors, dans la série génératrice du langage définie en (20), la probabilité p_w par l'opérateur générateur $\mathbf{H}_{[w]}$ défini en (6), et on obtient ce qu'on appelle l'opérateur générateur du langage $\mathcal{L}$ défini par

$$\mathbf{L}(z) := \sum_{w\in\mathcal{L}} \mathbf{H}_{[w]} z^{|w|}. \tag{21}$$

La propriété de composition (8) sur les opérateurs permet de traduire les opérations sur les langages en opérations sur les opérateurs générateurs associés. Grâce à (7), on peut alors revenir à la série génératrice par la relation

$$L(z) = \int_I \mathbf{L}(z)[f](t)\,dt. \tag{22}$$

Exemple d'application : les motifs généralisés. (Le cadre est celui des sources de type 2 ou 3 de la Section 3.4). On pourra se reporter à [9] pour plus de précisions.

Un motif généralisé $\mathcal{L}$ est une suite finie de langages construits sur le même alphabet $\mathcal{M}$, de la forme $\mathcal{L} := (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_r)$. Chacun des langages $\mathcal{L}_i$ est de longueur finie (c'est-à-dire que pour chacun des langages, on a une borne uniforme sur la longueur des mots). On dit que le motif $\mathcal{L}$ apparaît dans le texte $T \in \mathcal{M}^\star$ si le texte contient comme sous-séquence un élément $\ell = (\ell_1, \ell_2, \dots, \ell_r)$ de $\mathcal{L}$. Dans ce cas, T est de la forme

$$T = w_0\ell_1 w_1 \ell_2 \dots w_i \ell_i w_{i+1} \dots w_r \ell_r w_{r+1} \qquad \text{avec} \qquad w_i \in \mathcal{M}^\star \quad \text{et} \quad \ell_i \in \mathcal{L}_i.$$

Cette notion de motif généralisé recouvre beaucoup de problèmes de recherche de motifs, tout particulièrement les motifs cachés, qui apparaissent naturellement dans des contextes divers (bioinformatique, détection d'intrusions) et a déjà été étudiée dans le contexte des sources sans mémoire [24].

L'ensemble de toutes les *occurrences* du motif généralisé $\mathcal{L}$ est alors la collection $\rho(\mathcal{L})$ (avec répétitions) donnée par concaténation,

$$\rho(\mathcal{L}) = \mathcal{M}^\star \times \mathcal{L}_1 \times \mathcal{M}^\star \times \mathcal{L}_2 \times \cdots \times \mathcal{M}^\star \times \mathcal{L}_r \times \mathcal{M}^\star. \tag{23}$$

Cette opération ρ transforme une suite finie de langages en une collection de mots (par opposition à un langage qui est un ensemble de mots, une collection est un multi-ensemble de mots), et dans la collection $\rho(\mathcal{L})$, un texte T est présent autant de fois qu'il contient d'occurrences de $\mathcal{L}$. Pour un texte T de longueur n, on désigne par $\Omega_n(\mathcal{L}, T)$ le nombre d'occurrences de $\mathcal{L}$ dans T, et la remarque précédente permet de montrer que la série génératrice des espérances coïncide exactement avec la série génératrice $L(z)$ de la collection $\rho(\mathcal{L})$,

$$L(z) := \sum_{w\in\rho(\mathcal{L})} p_w z^{|w|} = \sum_{n\geq 1} \mathbf{E}\big[\Omega_n(\mathcal{L}, T)\big]\, z^n.$$

Grâce aux règles de transfert citées précédemment, l'opérateur générateur $\mathbf{L}(z)$ de la collection $\rho(\mathcal{L})$ s'écrit facilement en fonction des opérateurs générateurs $\mathbf{L}_i(z)$ des langages et de l'opérateur $(\mathbf{1} - z\mathbf{H})^{-1}$ associé au langage $\mathcal{M}^\star$,

$$\mathbf{L}(z) = (I - z\mathbf{H})^{-1} \circ \mathbf{L}_r(z) \circ (I - z\mathbf{H})^{-1} \circ \cdots \circ \mathbf{L}_1(z) \circ (I - z\mathbf{H})^{-1}. \tag{24}$$

Cet opérateur contient $r+1$ occurrences du quasi-inverse $(I - z\mathbf{H})^{-1}$, qui « apportent » chacune un pôle en $z = 1$. Elles sont « mélangées » avec les opérateurs $\mathbf{L}_i(z)$ des langages $\mathcal{L}_i$ qui sont des polynômes en z (et n'apportent pas de pôles). Via la relation (22), on caractérise alors aisément les singularités de la série $L(z)$ et on obtient ainsi le résultat suivant :

Proposition. *Le nombre moyen $\mathbf{E}\big[\Omega_n(\mathcal{L}, T)\big]$ d'occurrences du motif généralisé $\mathcal{L}$ dans un texte de longueur n produit par une source dynamique de type 2 ou 3 vérifie :*

$$\mathbf{E}\big[\Omega_n(\mathcal{L}, T)\big] = \binom{n+r}{r}\pi(\mathcal{L}) + \binom{n+r-1}{r-1}\pi(\mathcal{L})\big(C(\mathcal{L}) - N(\mathcal{L})\big) + O(n^{r-2}).$$

Ici, $\pi(\mathcal{L})$ est le poids total du motif

$$\pi(\mathcal{L}) := \prod_{i=1}^{r} p(\mathcal{L}_i)$$

où, pour une collection $\mathcal{M}$, on pose

$$p(\mathcal{M}) := \sum_{w \in \mathcal{M}} p_w,$$

et $N(\mathcal{L})$ est sa longueur moyenne. Le coefficient $C(\mathcal{L})$ décrit la corrélation entre deux composantes successives du motif et s'exprime en fonction de l'opérateur $\mathbf{R}$ défini en (15).

À l'aide des mêmes techniques, utilisées cette fois pour des collections associées aux doubles occurrences, on peut avoir accès à la variance du nombre d'occurrences. On démontre ainsi un phénomène de concentration autour de la valeur moyenne [9].

4.2. **Les grandeurs fondamentales d'une source (cas d'une source de type 1).** Pour plus de précisions, on peut consulter [43]. Les séries de Dirichlet des probabilités fondamentales font intervenir les quantités p_w^s et sont définies par

$$\Lambda_k(s) := \sum_{|w|=k} p_w^s, \qquad \Lambda(s) := \sum_{k \geq 0} \Lambda_k(s) = \sum_{w \in \mathcal{M}^\star} p_w^s. \tag{25}$$

La plupart des grandeurs fondamentales associées à la source $\mathcal{S}$ s'expriment à l'aide de ces séries. Nous en donnons quatre exemples.

Entropie. L'entropie $h(\mathcal{S})$ de la source satisfait à la relation

$$h(\mathcal{S}) := \lim_{k \to \infty} \frac{-1}{k} \sum_{|w|=k} p_w \log p_w = \lim_{k \to \infty} \frac{-1}{k} \left(\frac{d}{ds} \Lambda_k(s) \right) \Big|_{s=1}.$$

Probabilité de coïncidence. La coïncidence $C(x, y)$ entre les deux mots $M(x)$ et $M(y)$ pour deux réels x et y tirés indépendamment selon une même loi est la longueur du plus long prefixe commun. La probabilité pour que $M(x)$ et $M(y)$ aient le même préfixe de longueur k est donc la probabilité

de l'événement $\big[C(x,y] \geq k\big]$. Cet événement se produit si (et seulement si) les deux réels x et y appartiennent à un même intervalle fondamental I_w de profondeur k (voir Section 2.2). On a ainsi

$$\mathbf{P}\big[C(x,y) \geq k\big] = \sum_{|w|=k} p_w^2,$$

et la probabilité de coïncidence $c(\mathcal{S})$ vérifie la relation

$$c(\mathcal{S}) := \lim_{k\to\infty} \left(\sum_{|w|=k} p_w^2 \right)^{1/k} = \lim_{k\to\infty} \Lambda_k(2)^{1/k}.$$

Équirépartition des mots de longueur k. On cherche à décrire les probabilités possibles de tous les mots de longueur k. Plus précisément, on veut décrire la distribution de l'ensemble

$$\mathcal{P}_k := \{\, p_w \mid w \in \mathcal{M}^k \,\}.$$

On définit sur $\mathcal{M}^k$ une variable aléatoire ℓ_k par $\ell_k(w) := \log p_w$, et on veut analyser la distribution de la variable ℓ_k. Un outil important pour l'analyse d'une variable aléatoire X est la série génératrice des moments,

$$M(X)(s) := \mathbf{E}\big[\exp(sX)\big] = \sum_{n\geq 0} \frac{s^n}{n!}\, \mathbf{E}[X^n].$$

Ici, la série génératrice des moments de la variable ℓ_k, désignée par $M_k(s)$, vérifie

$$M_k(s) := \mathbf{E}[p_w^s] = \sum_{w\in\mathcal{M}^k} p_w p_w^s = \Lambda_k(1+s). \tag{26}$$

Nombre de préfixes assez probables. La quantité $B(\rho)$ désigne le nombre de préfixes w dont la probabilité est au moins égale à ρ ($\rho \to 0$). Un outil principal est ici une transformation intégrale, la transformée de Mellin (voir [23]), que nous utiliserons aussi en Section 4.3. La transformée de Mellin de la fonction B est reliée à la fonction $\Lambda(s)$, via la relation

$$\Lambda(s) = s \int_0^\infty B(x) x^{s-1}\, dx.$$

Dans les quatre exemples, les grandeurs caractéristiques $h(\mathcal{S})$, $c(\mathcal{S})$ et les fonctions B et $M_k(s)$ s'expriment donc en fonction des séries de Dirichlet $\Lambda_k(s)$ et $\Lambda(s)$ définies en (25).

Transcription algébrique. Dans le cas des sources dynamiques complètes (ou markoviennes), et grâce à la relation (13), les séries de Dirichlet (25) ont une autre expression en fonction de l'opérateur de transfert sécant,

$$\Lambda_k(s) = \mathfrak{H}_s^k[L^s](0,1) \quad \text{et} \quad \Lambda(s) = (\mathbf{1} - \mathfrak{H}_s)^{-1}[L^s](0,1) \tag{27}$$

où L est aussi définie en (13).

Traitement analytique. Dans le cas des sources dynamiques de type 1, les bonnes propriétés spectrales de l'opérateur de transfert sécant induisent un bon comportement des séries de Dirichlet, et tous les résultats vont s'exprimer en fonction de la valeur propre dominante $s \mapsto \lambda(s)$, omniprésente dans ce cadre. Remarquons que $s \mapsto \lambda(s)$ ne dépend que du système dynamique et non pas de la

densité (analytique) initiale f choisie. Au voisinage de l'axe réel, la série $\Lambda_k(s)$ se comporte comme une quasi-puissance : il existe $a(s)$ tel que, sur un voisinage complexe d'un point s_0 réel, on ait

$$\Lambda_k(s) \sim a(s)\lambda(s)^k$$

pour $k \to \infty$ uniformément en s sur ce voisinage. Par ailleurs, $\Lambda(s)$ est analytique sur le demi-plan $\Re(s) > 1$, avec un pôle simple en $s = 1$, et pour s au voisinage de 1 sur ce domaine,

$$\Lambda(s) \sim \frac{a(s)}{1-\lambda(s)} \sim \frac{-1}{\lambda'(1)}\frac{1}{s-1}.$$

Dans le cas où la fonction $s \mapsto \lambda(s)$ est périodique, il peut y avoir d'autres pôles régulièrement espacés sur la droite $\Re(s) = 1$. Ce phénomène de périodicité se produit en particulier pour certaines classes de sources simples, mais ce sont essentiellement les seuls cas où il se produit.

On déduit d'abord aisément les deux relations

$$h(\mathcal{S}) = -\lambda'(1), \qquad c(\mathcal{S}) = \lambda(2). \tag{28}$$

Des techniques classiques d'analyse (transformée de Mellin, théorème taubérien) permettent d'obtenir le comportement de la fonction B au voisinage de 0. Par exemple, si la fonction $s \mapsto \lambda(s)$ n'est pas périodique, on obtient

$$B(\rho) \sim \frac{-1}{\lambda'(1)\rho} \quad \text{pour} \quad \rho \to 0.$$

Enfin, la série génératrice des moments (26) se comporte presque exactement comme la fonction $a(s)\lambda(1+s)^k$, ce comportement étant uniforme en s sur un voisinage de 0. Alors des résultats classiques, dûs en particulier à Hwang [31], montrent que la variable aléatoire ℓ_k suit asymptotiquement (quand $k \to \infty$) une loi gaussienne, avec

$$\mathbf{E}[\ell_k] \sim \lambda'(1)k, \qquad \mathbf{Var}[\ell_k] \sim \big(\lambda''(1) - \lambda'(1)^2\big)k, \tag{29}$$

la convergence vers la loi normale étant en $O\big(1/\sqrt{k}\,\big)$. (Là encore, il y a quelques exceptions, essentiellement liées à des sources simples.) Ce résultat est une version forte d'un théorème célèbre en Théorie de l'Information, dû à Shannon–Macmillan–Breiman qui montre que pour de « bonnes sources », l'ensemble $\mathcal{M}^k$ des mots de longueur k se répartit en deux sous-ensembles : les mots probables, qui ont à peu près tous la même probabilité, égale à $\exp\big(-kh(\mathcal{S})\big)$ et un ensemble de mots très peu probables. Le résultat obtenu ici démontre en plus un phénomène de concentration autour de la valeur moyenne.

4.3. **Comportement des arbres dictionnaires (cas des sources de type 1).** Pour plus de précisions, on peut consulter [6, 8, 15, 14, 21].

Une structure de données essentielle dans les algorithmes de traitement du texte est l'arbre digital, ou trie (le mot « trie » est obtenu par contraction des deux mots « tree » et « retrieval »), et ses variations (le patricia-trie et le suffix-trie). Un trie est tout simplement un arbre qui implante un dictionnaire : un dessin suffit à comprendre comment il fonctionne (voir Figure 11). Les nœuds internes servent à diriger la recherche, et ce sont les feuilles qui contiennent les mots du dictionnaire. Il y a en particulier (et par définition) autant de feuilles que de mots dans le trie. Un nœud du trie (interne ou feuille) peut être étiqueté par le chemin qui le lie à la racine. Pour obtenir le patricia-trie associé, on supprime simplement les noeuds internes qui ne sont pas des points de branchement (voir Figure 11).

Les atouts du trie sont sa facilité d'implantation et son dynamisme : il est facile à modifier (insertion, suppression, etc.). L'efficacité de la structure de données associée est liée à la compacité de la forme de l'arbre, qu'on peut quantifier par les paramètres usuels d'un arbre : longueur de

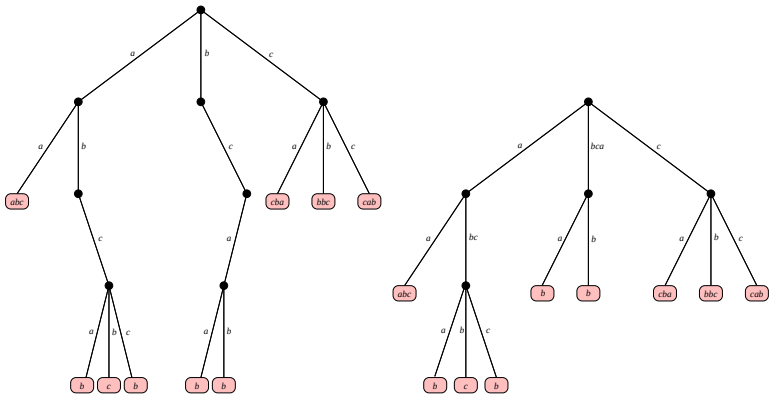


FIGURE 11. Un exemple de trie et du patricia-trie associé.

cheminement externe, nombre de nœuds (internes) — encore appelé taille —, hauteur, ... Ici, la mesure de la donnée est le nombre n de mots présents dans le dictionnaire.

L'analyse de la structure de trie et des arbres de sa descendance a été largement étudiée dans le cadre des sources classiques. On peut consulter à ce sujet le livre de W. Szpankowski [40]. Ici, nous cherchons à faire l'analyse dans le cadre « dynamique ». Il y a une très grande affinité entre les propriétés du trie et celles de la source dynamique. Un trie construit sur un ensemble de n mots est défini par l'ensemble $X := \{x_1, x_2, \ldots, x_n\}$ des n réels qui ont donné naissance aux n mots. On le désigne par la suite par $T(X)$. Un tel trie est complètement déterminé par les nœuds internes qui sont effectivement présents. Or ces nœuds-là sont étiquetés par les préfixes w pour lesquels l'intervalle fondamental I_w contient au moins deux éléments de X. Pour que les contributions de l'ensemble X dans deux intervalles fondamentaux disjoints I_w et $I_{w'}$ soient indépendants, on est alors conduit à travailler dans un modèle de Poisson : on tire la cardinalité N de l'ensemble X suivant une loi de Poisson de paramètre z,

$$\mathbf{P}[N = n] = e^{-z}\,\frac{z^n}{n!},$$

puis on tire les n réels de l'ensemble X indépendamment suivant une loi de densité f. Alors la variable aléatoire N_w qui mesure la cardinalité de l'ensemble $I_w \cap X$ suit une loi de Poisson de paramètre $p_w z$: et, crac, voilà la probabilité fondamentale p_w qui intervient de nouveau ! La probabilité d'existence du nœud interne n_w d'étiquette w est égale à $\mathbf{P}[N_w \geq 2]$, tandis que la contribution de ce noeud n_w à la longueur moyenne de cheminement externe est $\mathbf{E}[\, N_w \mid N_w \geq 2\,]$.

On obtient ainsi l'expression des valeurs moyennes des deux variables taille, S, et longueur de cheminement externe, P. L'indice z fait référence au paramètre du modèle de Poisson.

$$\mathbf{E}[P_z] = \sum_{w\in\mathcal{M}^\star} p_w z(1 - e^{-p_w z}), \qquad \mathbf{E}[S_z] = \sum_{w\in\mathcal{M}^\star} \left(1 - e^{-p_w z}(1 + p_w z)\right).$$

Ces deux expressions sont des sommes harmoniques, et l'instrument pour étudier le comportement asymptotique de telles expressions (pour $z \to \infty$) est la transformée de Mellin [23]

$$\hat{A}(s) := \int_0^\infty A(x) x^{s-1}\, dx$$

car la transformée d'une somme harmonique $A(z)$ se factorise en un produit de deux facteurs : si

$$A(z) = \sum_{w\in\mathcal{M}^\star} g(p_w z),$$

alors

$$\hat{A}(s) = \hat{g}(s) \sum_{w\in\mathcal{M}^\star} p_w^{-s}.$$

En particulier, les transformées de Mellin des espérances de la taille et de la longueur de cheminement externe font intervenir la série de Dirichlet $\Lambda(s)$ définie en (25), et, tout particulièrement son comportement singulier autour de son pôle dominant en $s = 1$ (qui s'exprime à l'aide de l'entropie $-\lambda'(1)$).

Par ailleurs, la hauteur de $T(X)$ est au plus égale à k pourvu qu'il n'existe pas de noeuds internes n_w associés à des préfixes de longueur k. Compte tenu du phénomène d'indépendance induit par le modèle de Poisson, on a donc :

$$\mathbf{P}[H_z \le k] = \prod_{w\in\mathcal{M}^k} \mathbf{P}[N_w \le 1] = \prod_{w\in\mathcal{M}^k} e^{-p_w z}(1 + p_w z)$$

de sorte que

$$\log \mathbf{P}[H_z \le k] = -z + \sum_{w\in\mathcal{M}^k} \log(1 + p_w z). \tag{30}$$

Supposons dans un premier temps que l'on puisse utiliser dans (30), pour tous les couples (w, z) et successivement, les deux approximations suivantes

$$-p_w z + \log(1 + p_w z) \sim -\frac{p_w^2 z^2}{2} \qquad \text{puis} \qquad \sum_{w\in\mathcal{M}^k} p_w^2 = \Lambda_k(2) \sim a\lambda(2)^k.$$

Alors, la valeur moyenne de la hauteur s'écrit

$$\mathbf{E}[H_z] \sim \sum_{k\ge 0} \left(1 - \exp\left(-\frac{az^2}{2}\lambda^k(2)\right)\right)$$

et c'est encore une somme harmonique ! La série de Dirichlet associée,

$$\sum_{k\ge 0} \lambda(2)^{-ks} = \frac{1}{1 - \lambda(2)^{-s}}$$

a un pôle simple en $s = 0$, avec un résidu qui fait intervenir $\big|\log \lambda(2)\big|$.

On peut d'abord rendre rigoureux tout ce qui est dit précédemment. Ensuite, il faut revenir au modèle dit de Bernoulli où le nombre des mots est fixé égal à n. Dans ce cas les paramètres étudiés se notent $S^{[n]}, P^{[n]}, H^{[n]}$. On obtient finalement :

Théorème. *Dans une source dynamique de type 1, les trois paramètres de forme du trie (taille, longueur de cheminement, hauteur) construit sur n mots de la source tirés indépendamment ont pour valeur moyenne asymptotique (pour $n \to \infty$) les quantités suivantes qui font intervenir l'entropie et la probabilité de coïncidence,*

$$\mathbf{E}\left[S^{[n]}\right] \sim \frac{n}{h(\mathcal{S})}, \qquad \mathbf{E}\left[P^{[n]}\right] \sim \frac{n\log n}{h(\mathcal{S})}, \qquad \mathbf{E}\left[H^{[n]}\right] \sim \frac{\log n}{2\big|\log c(\mathcal{S})\big|}.$$

Dans le cas de sources périodiques, le terme principal de $\mathbf{E}\left[S^{[n]}\right]$ fait intervenir un facteur supplémentaire, qui contient une fonction de n oscillante, avec de faibles amplitudes.

Cette approche est suffisamment robuste pour s'adapter à l'analyse des tries plus compliqués (patricia-tries, tries hybrides) ou pour étudier d'autres paramètres de tries simples (par exemple, la hauteur de pile, qui fait intervenir une source induite au sens du paragraphe 3.4). Le suffix-trie est d'un abord plus complexe : c'est par définition un trie construit sur l'ensemble des suffixes d'un mot, et la propriété d'indépendance entre les mots du dictionnaire n'est plus préservée.

5. **Analyse dynamique des algorithmes arithmétiques**

L'objet de cette section est d'illustrer sur un exemple simple l'utilisation des opérateurs de transfert pour l'analyse d'algorithmes arithmétiques. Nous commençons par traiter le cas de l'algorithme d'Euclide standard du calcul du p. g. c. d. de deux entiers. Le résultat que nous exposons ici n'est pas original, puisque le nombre moyen d'itérations de l'algorithme d'Euclide classique a été déterminé autour de 1970 indépendamment par Heilbronn [28] et Dixon [20]. La méthode décrite est, elle, typique de l'analyse dynamique et peut être facilement généralisée dans de multiples directions (voir paragraphes 5.5 et 5.6).

À partir d'une entrée (v_1, v_0) formée de deux entiers positifs vérifiant $v_1 \leq v_0$ l'algorithme effectue une suite de divisions euclidiennes,

$$v_0 = a_1 v_1 + v_2, \qquad v_1 = a_2 v_2 + v_3, \qquad \dots \qquad v_{k-1} = a_k v_k + 0. \tag{31}$$

L'algorithme s'arrête dès qu'apparaît un reste nul. Le coût étudié ici est le nombre k de divisions successives effectuées.

5.1. **Le système dynamique sous-jacent à l'algorithme.** Une étape de l'algorithme remplace une paire (v_1, v_0) par la paire (v_2, v_1) avec

$$\frac{v_2}{v_1} = \frac{v_0}{v_1} - a_1.$$

Si, à la place des paires d'entiers (v_1, v_0), on considère les rationnels de la forme v_1/v_0, la transformation T définie par

$$T(x) = \left\{ \frac{1}{x} \right\} := \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$$

où $\lfloor x \rfloor$ désigne la partie entière de x, exprime (v_2/v_1) en fonction de (v_1/v_0). Le système sous-jacent (voir Figure 12) est complet et l'ensemble des branches inverses de la transformation T est

$$\mathcal{H} = \left\{ h : z \mapsto \frac{1}{z+m} \;\middle|\; m \in \mathbb{N},\ m \neq 0 \right\}.$$

5.2. **Les séries génératrices des coûts.** L'ensemble des entrées possibles de l'algorithme est

$$\tilde{\Omega} = \left\{ (u, v) \;\middle|\; 0 \leq u \leq v \right\},$$

et l'ensemble des entrées de taille N est

$$\tilde{\Omega}_N := \left\{ (u, v) \;\middle|\; 0 \leq u \leq v \leq N \right\}.$$

Pour simplifier l'étude, nous travaillons sur des ensembles d'entrées possibles plus restreints,

$$\Omega = \left\{ (u, v) \in \tilde{\Omega} \;\middle|\; \mathrm{pgcd}(u, v) = 1 \right\}, \qquad \Omega_N := \left\{ (u, v) \in \tilde{\Omega}_N \;\middle|\; \mathrm{pgcd}(u, v) = 1 \right\},$$

formés des entrées pour lesquelles la réponse de l'algorithme est connue à l'avance ..., mais nous reviendrons ensuite aux ensembles $\tilde{\Omega}$, $\tilde{\Omega}_N$ plus « naturels ».

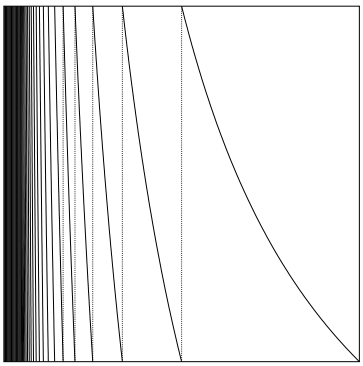


FIGURE 12. Le système dynamique euclidien standard.

Désignons par $C(u,v)$ la fonction de coût correspondant au nombre de divisions effectuées par l'algorithme d'Euclide sur l'entrée $(u,v) \in \Omega$. L'analyse en moyenne du coût C est l'étude du comportement asymptotique de la valeur moyenne du coût C sur Ω_N ou sur $\tilde{\Omega}_N$

$$\mathrm{E}_N[C] = \frac{\sum_{(u,v)\in\Omega_N} C(u,v)}{\sum_{(u,v)\in\Omega_N} 1}, \qquad \tilde{\mathrm{E}}_N[C] = \frac{\sum_{(u,v)\in\tilde{\Omega}_N} C(u,v)}{\sum_{(u,v)\in\tilde{\Omega}_N} 1},$$

lorsque N tend vers ∞. Les séries génératrices de Dirichlet

$$G_1(s) := \sum_{(u,v)\in\Omega} \frac{1}{v^{2s}} = \sum_{v\geq 1} \frac{a_v}{v^{2s}}, \qquad G_C(s) := \sum_{(u,v)\in\Omega} \frac{C(u,v)}{v^{2s}} = \sum_{v\geq 1} \frac{c_v}{v^{2s}}$$

et leurs homologues « tildées »

$$\tilde{G}_1(s) := \sum_{(u,v)\in\tilde{\Omega}} \frac{1}{v^{2s}} = \sum_{v\geq 1} \frac{\tilde{a}_v}{v^{2s}}, \qquad \tilde{G}_C(s) := \sum_{(u,v)\in\tilde{\Omega}} \frac{C(u,v)}{v^{2s}} = \sum_{v\geq 1} \frac{\tilde{c}_v}{v^{2s}}$$

sont relatives aux coûts intervenant au numerateur et au dénominateur. Les relations

$$\tilde{G}_C(s) = \zeta(s) G_C(s), \qquad \tilde{G}_1(s) = \zeta(s) G_1(s)$$

(où $\zeta(s)$ est la série zeta de Riemann) montrent qu'il suffit de travailler sur Ω, comme il était annoncé. Ici, a_v désigne le nombre d'éléments de Ω ayant un dénominateur égal à v et c_v désigne la somme des coûts associés aux éléments de Ω ayant un dénominateur égal à v. Remarquons que $E_N[C]$ s'exprime en fonction des sommes partielles des coefficients des séries précédentes :

$$E_N[C] = \frac{\sum_{v\leq N} c_v}{\sum_{v\leq N} a_v}.$$

Le comportement asymptotique des sommes partielles est lié au comportement des fonctions G_1 et G_C via le théorème taubérien suivant [19, 41].

Théorème taubérien. *Soit une série de Dirichlet $F(s)$ à coefficients positifs ou nuls*

$$F(s) = \sum_{n\geq 1} \frac{a_n}{n^{2s}}$$

telle que :

1. *$F(s)$ converge dans un demi-plan $\Re(s) > \sigma > 0$ et est analytique sur $\Re(s) = \sigma$, $s \neq \sigma$,*

2. *il existe* $\gamma \geq 0$ *tel que* $F(s) = A(s)(s-\sigma)^{-\gamma-1} + C(s)$ *où* A *et* C *sont analytiques en* σ *et* $A(\sigma) \neq 0$.

Alors, lorsque $N \longrightarrow \infty$,

$$\sum_{n \leq N} a_n = \frac{2^\gamma A(\sigma)}{\sigma \Gamma(\gamma+1)} N^{2\sigma} \log^\gamma N \left(1 + \epsilon(N)\right), \quad \textit{avec} \quad \epsilon(N) \longrightarrow 0.$$

Pour appliquer le théorème précédent, il faut exprimer les deux séries de Dirichlet en fonction de l'opérateur de transfert associé au système dynamique. Ce seront alors les propriétés spectrales de cet opérateur qui permettront d'étudier le comportement de G_1 et G_C au voisinage de $\sigma = 1$.

5.3. **Lien entre la fonction de coût et les opérateurs de transfert.** Les couples (u, v) de Ω sur lesquels l'algorithme effectue exactement k divisions sont ceux qui s'écrivent

$$\frac{u}{v} = h(0) \qquad \text{avec} \qquad h = h_1 \circ \cdots \circ h_k \in \mathcal{H}^k.$$

Puisque toutes les branches inverses $h \in \mathcal{H}^\star$ sont des homographies de déterminant 1, la dérivée $h'(z)$ s'exprime simplement en fonction du carré de son dénominateur : pour $(u, v) \in \Omega$ tel que $u/v = h(0)$ avec $h \in \mathcal{H}^\star$, on a $1/v^2 = \left|h'(0)\right|$.

Ceci permet d'exprimer différemment les séries de Dirichlet

$$G_1(s) = \sum_k \sum_{h \in \mathcal{H}^k} \left|h'(0)\right|^s, \qquad G_C(s) = \sum_k k \sum_{h \in \mathcal{H}^k} \left|h'(0)\right|^s. \tag{32}$$

Et c'est maintenant que l'opérateur de transfert $\mathbf{H}_s$ associé au système dynamique intervient. Ici, il est appelé opérateur de Ruelle–Mayer et prend la forme bien connue suivante

$$\mathbf{H}_s[f](x) = \sum_{m \geq 1} \left(\frac{1}{m+x}\right)^{2s} f\left(\frac{1}{m+x}\right).$$

La comparaison des relations (10) et (32) montre que

$$G_1(s) = \sum_{k \geq 0} \mathbf{H}_s^k[\mathbf{1}](0) = (\mathbf{1} - \mathbf{H}_s)^{-1}[\mathbf{1}](0),$$
$$G_C(s) = \sum_{k \geq 1} k \mathbf{H}_s^k[\mathbf{1}](0) = \mathbf{H}_s(\mathbf{1} - \mathbf{H}_s)^{-2}[\mathbf{1}](0).$$

Les séries de Dirichlet des coûts s'expriment donc à l'aide du quasi-inverse $(\mathbf{1} - \mathbf{H}_s)^{-1}$ de l'opérateur de transfert.

5.4. **Analyse spectrale.** Comme le système dynamique associé est de type 1, l'espace fonctionnel adéquat est l'espace $\mathcal{A}_\infty(\mathcal{V})$ et l'opérateur $\mathbf{H}_s$ est compact et vérifie les propriétés (P1), (P2) et (P3) de la Section 3.2. La quantité $(\mathbf{1} - \mathbf{H}_s)^{-1}[\mathbf{1}](0)$ possède un pôle d'ordre 1 en $s = 1$ dont le résidu est $-1/\lambda'(1)$. Comme on l'a vu en (28), la valeur $-\lambda'(1)$ est l'entropie du système dynamique T. Ici, cette entropie fait intervenir des constantes classiques et vaut

$$h = \frac{\pi^2}{6 \log 2} \approx 2.3731.$$

Le théorème taubérien s'applique en $\sigma = 1$, avec $\gamma = 0$ pour G_1 et $\gamma = 1$ pour G_C. Il permet d'obtenir le comportement asymptotique de $\mathrm{E}_N[C]$ et $\tilde{\mathrm{E}}_N[C]$,

$$\mathrm{E}_N[C] \sim \tilde{\mathrm{E}}_N[C] \sim \frac{-2}{\lambda'(1)} \log N = \frac{12 \log 2}{\pi^2} \log N.$$

Cet exemple montre, dans un cas simple, la démarche suivie lors de l'analyse dynamique d'algorithmes arithmétiques, tout à fait conforme au schéma décrit en Figure 1. De fait, l'analyse dynamique a permis d'obtenir de nombreux autres résultats sur les algorithmes euclidiens (autres coûts, autres algorithmes).

5.5. **D'autres coûts.** On peut d'abord s'intéresser à d'autres paramètres, plus précis. L'un d'entre eux est la complexité en bits, désignée par B, qui compte le nombre d'opérations binaires effectuées par l'algorithme. Des méthodes similaires à celles décrites ici (mais plus subtiles ...) permettent d'évaluer la complexité moyenne en bits $E_N[B]$

$$E_N[B] \sim \frac{6\log^2 2}{\pi^2}\left(2+\log_2\prod_{k=0}^{\infty}\left(1+\frac{1}{2^k}\right)\right)\log_2^2 N.$$

On utilise (voir [1, 46]) à la fois des opérateurs pondérés (voir 2.5) et les dérivés des opérateurs par rapport à la variable s.

On peut aussi se poser beaucoup d'autres questions sur les rationnels, analogues à celles qu'on se pose classiquement sur le développement en fraction continue des nombres réels. Par exemple : quelle est la fréquence d'un chiffre donné dans le développement en fraction continue d'un rationnel ? Pour les réels, on répond à cette question à l'aide des théorèmes ergodiques. Ici, on remplace l'utilisation des théorèmes ergodiques par les théorèmes taubériens, et on peut montrer que vis-à-vis d'une classe très large de paramètres, les rationnels se comportent « en moyenne » comme les réels le font presque sûrement [46].

5.6. **La classe des algorithmes euclidiens.** Il existe toute une classe d'algorithmes d'Euclide, car il y a autant d'algorithmes d'Euclide que de divisions possibles : on peut effectuer des divisions caractérisées par la classe des quotients (quelconques, pairs, impairs), par la position du reste (division par défaut, par excès, centrée, ou plus généralement α-division), par la parité du reste (on peut vouloir un reste impair, qu'on obtient en enlevant les puissances de 2 du reste classique, ce qui se justifie tout particulièrement quand on veut calculer le symbole de Jacobi à l'aide de la loi de réciprocité quadratique). On peut aussi éviter les divisions et les remplacer par des opérations plus simples (soustractions et décalages binaires) : c'est le cas de l'algorithme binaire, de l'algorithme Plus-Moins et des algorithmes binaires généralisés. On peut aussi éviter les divisions entre grands entiers, et les remplacer par des divisions entre des entiers plus petits : c'est le principe de l'algorithme de Lehmer–Euclide.

À ce jour, les méthodes d'analyse dynamique ont permis d'établir un cadre très général où l'on a pu analyser (presque) tous les algorithmes cités. La démarche décrite dans les paragraphes 5.2 et 5.3 se généralise aisément, car, bien que les systèmes dynamiques « euclidiens » puissent être extrêmement divers, ils ont un point commun important : toutes leurs branches sont des homographies. Comme la dérivée d'une homographie s'exprime en fonction du carré de son dénominateur (avec une intervention possible du déterminant qui n'est plus toujours égal à 1), on peut relier les séries de Dirichlet des coûts et les opérateurs de transfert.

Mais la géométrie des branches et les propriétés d'expansion peuvent vraiment varier d'un algorithme à l'autre, et cette classe dite euclidienne regroupe (presque) toute la diversité possible des systèmes dynamiques. En particulier, les algorithmes pseudo-euclidiens (*i. e.* ceux où l'on « enlève » du reste les éventuelles puissances de 2) obligent à travailler avec des systèmes dynamiques probabilistes, où l'on prolonge la valuation dyadique, bien définie sur les rationnels, en une variable aléatoire sur les réels. Les « bons » espaces fonctionnels ne sont pas alors toujours faciles à trouver, et ils peuvent être autres que ceux qui sont décrits en 3.4. L'analyse fonctionnelle devient alors assez délicate, et moins standard. En particulier, dans [42], en utilisant un espace

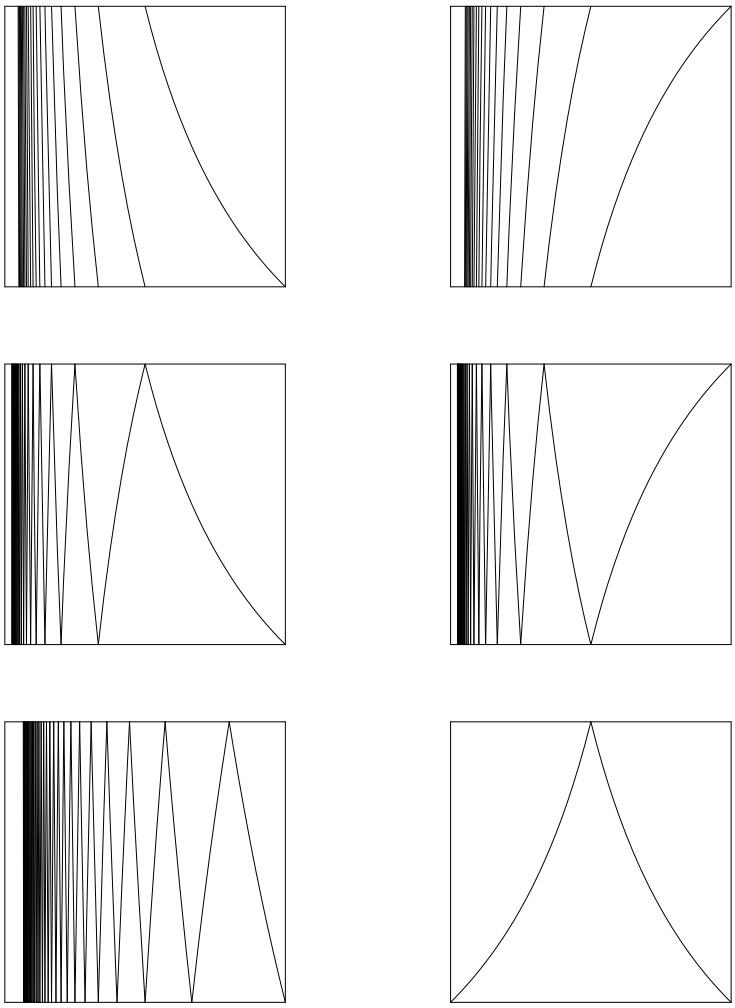


FIGURE 13. Les six systèmes euclidiens classiques ; à gauche, les « rapides » : Standard, Impair, Centré ; à droite, les « lents » : Par Excès, Pair, Soustractif.

fonctionnel bien adapté à l'algorithme binaire, qui est alors un espace de Hardy, on a pu analyser cet algorithme et répondre ainsi à une conjecture de Brent [11]. L'espace fonctionnel adapté à l'algorithme Plus-Moins [17], lui, reste encore à trouver !

Même pour les systèmes liés à des algorithmes euclidiens plus simples, la présence d'un point indifférent complique aussi l'analyse : il faut alors travailler avec le système dynamique induit (voir paragraphe 3.4), en utilisant des idées de Prellberg et Slawny [37]. C'est le cas des systèmes liés aux algorithmes Par Excès, Pair ou Soustractif. On obtient alors des algorithmes lents avec un nombre d'itérations quadratique (en $\log^2 N$) [47]. Par exemple, la Figure 13 represente six systèmes dynamiques euclidiens ; selon les colonnes, on obtient deux comportements bien différents ; la première colonne (qui contient les systèmes Standard, Impair et Centré) donne lieu à des algorithmes rapides ; la seconde colonne contient les systèmes Par Excès, Pair, et Soustractif qui ont chacun un point indifférent ; elle donne lieu à des algorithmes lents (comme annoncé en 1.4).

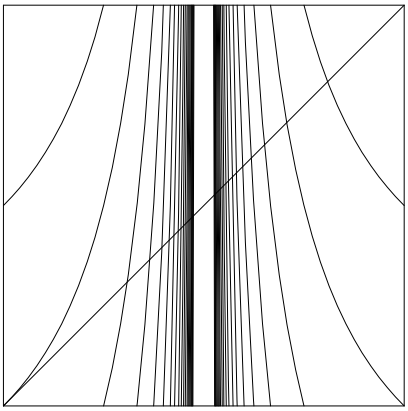


FIGURE 14. La famille des systèmes japonais.

La famille des algorithmes japonais est liée à une α-division de la forme $a = bq + r$ avec un reste $r \in]\, b(\alpha-1), b\alpha\,]$. Elle est représentée Figure 14. Le carré total est le carré $[\,-1,1\,]\times[\,-1,1\,]$. Pour obtenir la représentation du système japonais lié au paramètre $\alpha \in [\,0,1\,]$, on se limite à la fenêtre délimitée par le carré $[\,\alpha-1,\alpha\,]\times[\,\alpha-1,\alpha\,]$. Les systèmes dynamiques japonais sont le plus souvent non complets, en général non markoviens (sauf pour des cas très particuliers du paramètre α) et sont associés à des systèmes de type 3 [7].

On peut aussi chercher à analyser des extensions des algorithmes euclidiens en dimension supérieure : l'algorithme de Gauss qui réduit les réseaux en dimension 2 [18], l'algorithme qui calcule le signe d'un déterminant en utilisant deux développements « parallèles » en fraction continue [44]. De manière un peu inattendue, l'analyse de ces deux algorithmes se révèle proche et fait intervenir la grandeur $\lambda(2)$. L'analyse dynamique de l'algorithme LLL, si utilisé en théorie algorithmique des nombres et en cryptographie, reste aussi un problème ouvert à ce jour ...

5.7. **Les constantes euclidiennes.** Un certain nombre de constantes qui apparaissent dans l'analyse des algorithmes d'Euclide sont liés à des objets spectraux des opérateurs de transfert, et s'expriment en fonction de la valeur propre dominante $s \mapsto \lambda(s)$. Il s'agit tout particulièrement de l'entropie $-\lambda'(1)$, omniprésente, de la valeur $\lambda''(1)$ qui intervient dans les moments d'ordre 2, et de la valeur $\lambda(2)$ qui intervient dans la coïncidence (voir 4.2). Dans l'algorithme d'Euclide standard, la fonction propre dominante est explicite, et donc, l'entropie l'est aussi. Mais, même dans ce cas-là, les deux autres valeurs ne sont pas explicites. Il s'agit de préciser le statut de la calculabilité de ces constantes, pour les algorithmes d'Euclide généraux (où même l'entropie n'est plus explicite), et de les calculer, si leur statut le permet. Il s'agit aussi de calculer de manière exacte la dimension de Hausdorff de réels dont les fractions continues sont « contraintes ». On pourra consulter pour plus de détails [26, 27, 35, 45].

6. Un problème encore ouvert : l'analyse en distribution

Ici, nous avons décrit principalement des analyses en moyenne, où l'on cherche à déterminer principalement les valeurs moyennes de certains paramètres, ou plus généralement leurs moments d'ordre supérieur. C'est alors le comportement de l'opérateur de transfert (ou de ses généralisés) autour de la valeur $s = 1$ qui joue un rôle essentiel. Mais le rêve de l'algorithmicien consiste à effectuer une analyse en distribution de ces paramètres (*i. e.* chercher la distribution limite de ces paramètres quand la taille du problème devient grande). Ce sont alors les propriétés de l'opérateur de transfert à gauche de la droite $\Re(s) = 1$ qui vont intervenir. Plus précisément, une situation

favorable est celle où le quasi-inverse $(\mathbf{1} - \mathbf{H}_s)^{-1}$ possède une région sans pôle à gauche de la droite $\Re(s) = 1$. Dans ce cas, on peut espérer obtenir une distribution limite gaussienne pour une certaine classe de paramètres liés au système dynamique. Cela permettrait tout particulièrement d'obtenir une nouvelle preuve, plus simple, du résultat d'Hensley [30] qui montre que le nombre d'itérations de l'algorithme d'Euclide suit une loi asymptotiquement gaussienne. C'est l'objet d'un travail en cours [3].

Bibliography

[1] Akhavi (Ali) and Vallée (Brigitte). – Average bit-complexity of Euclidean algorithms. In *Automata, languages and programming (Geneva, 2000)*, pp. 373–387. – Springer, Berlin, 2000. Proceedings of ICALP'00.

[2] Baladi (Viviane). – *Positive transfer operators and decay of correlations*. – World Scientific Publishing Co., River Edge, NJ, 2000, *Advanced Series in Nonlinear Dynamics*, vol. 16, x+314p.

[3] Baladi (Viviane) and Vallée (Brigitte). – Analyse dynamique en distribution. – En préparation.

[4] Bedford (Tim), Keane (Michael), and Series (Caroline) (editors). – *Ergodic theory, symbolic dynamics, and hyperbolic spaces*. – The Clarendon Press Oxford University Press, New York, 1991, *Oxford Science Publications*, xvi+369p. Comptes rendus de l'atelier *Hyperbolic Geometry and Ergodic Theory*, Trieste, 17 au 28 avril 1989.

[5] Bourdon (Jérémie). – Size and path length of Patricia tries: dynamical sources context. *Random Structures & Algorithms*, vol. 19, n° 3-4, 2001, pp. 289–315. – Analysis of algorithms (Krynica Morska, 2000).

[6] Bourdon (Jérémie). – *Analyse dynamique des algorithmes : exemples en algorithmique du texte et en algorithmique arithmétique*. – Thèse, Université de Caen, 2002.

[7] Bourdon (Jérémie), Daireaux (Benoit), and Vallée (Brigitte). – Dynamical analysis of α-Euclidean algorithms. *Journal of Algorithms*, vol. 44, n° 1, 2002, pp. 246–285. – Analysis of algorithms.

[8] Bourdon (Jérémie), Nebel (Markus), and Vallée (Brigitte). – On the stack-size of general tries. *Theoretical Informatics and Applications*, vol. 35, n° 2, 2001, pp. 163–185.

[9] Bourdon (Jérémie) and Vallée (Brigitte). – Generalized pattern matching statistics. In *Mathematics and computer science, II (Versailles, 2002)*, pp. 249–265. – Birkhäuser, Basel, 2002.

[10] Boyarsky (Abraham) and Góra (Paweł). – *Laws of chaos*. – Birkhäuser Boston, Boston, MA, 1997, *Probability and its Applications*, xvi+399p. Invariant measures and dynamical systems in one dimension.

[11] Brent (Richard P.). – Analysis of the binary Euclidean algorithm. In Traub (J. F.) (editor), *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976)*, pp. 321–355. – Academic Press, New York, 1976. New directions and recent results.

[12] Broise (Anne). – Transformations dilatantes de l'intervalle et théorèmes limites. *Astérisque*, n° 238, 1996, pp. 1–109. – Études spectrales d'opérateurs de transfert et applications.

[13] Chazal (F.), Maume-Deschamps (V.), and Vallée (B.). – Erratum to "Dynamical sources in information theory: fundamentals intervals and word prefixes". – 2002. Soumis à *Algorithmica*. Disponible dans les *Cahiers du GREYC* et les *Prépublications du Laboratoire de Topologie de Dijon*.

[14] Clément (J.), Flajolet (P.), and Vallée (B.). – Dynamical sources in information theory: a general analysis of trie structures. *Algorithmica*, vol. 29, n° 1-2, 2001, pp. 307–369. – Average-case analysis of algorithms (Princeton, NJ, 1998).

[15] Clément (Julien). – *Arbres digitaux et sources dynamiques*. – Thèse de doctorat, Université de Caen, 2000.

[16] Collet (Pierre). – Some ergodic properties of maps of the interval. In *Dynamical systems (Temuco, Chile, 1991/1992)*, pp. 55–91. – Hermann, Paris, 1996. Comptes rendus de la première École CIMPA de l'UNESCO *Dynamical and Disordered Systems*.

[17] Daireaux (Benoît). – *Analyse d'algorithmes du PGCD*. – Mémoire de DEA, Université de Caen, 2001.

[18] Daudé (Hervé), Flajolet (Philippe), and Vallée (Brigitte). – An average-case analysis of the Gaussian algorithm for lattice reduction. *Combinatorics, Probability and Computing*, vol. 6, n° 4, 1997, pp. 397–433.

[19] Delange (Hubert). – Généralisation du théorème de Ikehara. *Annales scientifiques de l'École normale supérieure. Troisième série*, vol. 71, 1954, pp. 213–242.

[20] Dixon (John D.). – The number of steps in the Euclidean algorithm. *Journal of Number Theory*, vol. 2, 1970, pp. 414–422.

[21] Fayolle (Julien). – *Paramètres des arbres suffixes dans le cas des sources simples*. – Mémoire de DEA, Université de Paris VI, 2002.

[22] Flajolet (Philippe). – Analytic analysis of algorithms. In *Automata, languages and programming (Vienna, 1992)*, pp. 186–210. – Springer, Berlin, 1992.

[23] Flajolet (Philippe), Gourdon (Xavier), and Dumas (Philippe). – Mellin transforms and asymptotics: harmonic sums. *Theoretical Computer Science*, vol. 144, n° 1-2, 1995, pp. 3–58. – Volume spécial sur l'analyse mathématique des algorithmes.

[24] Flajolet (Philippe), Guivarc'h (Yves), Szpankowski (Wojciech), and Vallée (Brigitte). – Hidden pattern statistics. In Orejas (Fernando), Spirakis (Paul G.), and Van Leeuwen (Jan) (editors), *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings. Lecture Notes in Computer Science*, vol. 2076, pp. 152–165. – Springer-Verlag, 2001. Comptes rendus de ICALP'01.

[25] Flajolet (Philippe) and Sedgewick (Robert). – Analytic combinatorics. – Livre en préparation. Voir aussi les Rapports recherche INRIA, n° 1888, 2026, 2376 et 2956.

[26] Flajolet (Philippe) and Vallée (Brigitte). – Continued fraction algorithms, functional operators, and structure constants. *Theoretical Computer Science*, vol. 194, n° 1-2, 1998, pp. 1–34.

[27] Flajolet (Philippe) and Vallée (Brigitte). – Continued fractions, comparison algorithms, and fine structure constants. In *Constructive, experimental, and nonlinear analysis (Limoges, 1999)*, pp. 53–82. – American Mathematical Society, Providence, RI, 2000.

[28] Heilbronn (H.). – On the average length of a class of finite continued fractions. In *Number Theory and Analysis (Papers in Honor of Edmund Landau)*, pp. 87–96. – Plenum, New York, 1969.

[29] Hennion (Hubert). – Sur un théorème spectral et son application aux noyaux lipchitziens. *Proceedings of the American Mathematical Society*, vol. 118, n° 2, 1993, pp. 627–634.

[30] Hensley (Doug). – The number of steps in the Euclidean algorithm. *Journal of Number Theory*, vol. 49, n° 2, 1994, pp. 142–182.

[31] Hwang (Hsien-Kuei). – *Théorèmes limite pour les structures combinatoires et les fonctions arithmétiques*. – Thèse, École polytechnique, 1994.

[32] Kato (Tosio). – *Perturbation theory for linear operators*. – Springer-Verlag, Berlin, 1995, *Classics in Mathematics*, xxii+619p. Réimpression de l'édition de 1980.

[33] Krasnosel'skiĭ (M. A.). – *Positive solutions of operator equations*. – P. Noordhoff, Groningen, 1964, 381p.

[34] Lasota (Andrzej) and Mackey (Michael C.). – *Chaos, fractals, and noise*. – Springer-Verlag, New York, 1994, *Applied Mathematical Sciences*, vol. 97, xiv+472p. Stochastic aspects of dynamics. Deuxième édition.

[35] Lhote (Loïck). – *Modélisation et approximation de sources complexes*. – Mémoire de DEA, Université de Caen, 2002.

[36] Mayer (Dieter H.). – Continued fractions and related transformations. In *Ergodic theory, symbolic dynamics, and hyperbolic spaces (Trieste, 1989)*, pp. 175–222. – Oxford University Press, New York, 1991.

[37] Prellberg (Thomas) and Slawny (Joseph). – Maps of intervals with indifferent fixed points: thermodynamic formalism and phase transitions. *Journal of Statistical Physics*, vol. 66, n° 1-2, 1992, pp. 503–514.

[38] Ruelle (David). – *Thermodynamic formalism*. – Addison-Wesley Publishing Co., Reading, Mass., 1978, *Encyclopedia of Mathematics and its Applications*, vol. 5, xix+183p. The mathematical structures of classical equilibrium statistical mechanics.

[39] Ruelle (David). – *Dynamical zeta functions for piecewise monotone maps of the interval*. – American Mathematical Society, Providence, RI, 1994, *CRM Monograph Series*, vol. 4, viii+62p.

[40] Szpankowski (Wojciech). – *Average-case analysis of algorithms on sequences*. – John Wiley & Sons, Chichester, New York, 2001, *Wiley-Interscience Series in Discrete Mathematics*.

[41] Tenenbaum (Gérald). – *Introduction à la théorie analytique et probabiliste des nombres*. – Institut Élie Cartan, Nancy, France, 1990, vol. 13.

[42] Vallée (B.). – Dynamics of the binary Euclidean algorithm: functional analysis and operators. *Algorithmica*, vol. 22, n° 4, 1998, pp. 660–685. – Average-case analysis of algorithms.

[43] Vallée (B.). – Dynamical sources in information theory: fundamental intervals and word prefixes. *Algorithmica*, vol. 29, n° 1-2, 2001, pp. 262–306. – Average-case analysis of algorithms (Princeton, NJ, 1998).

[44] Vallée (Brigitte). – Algorithms for computing signs of 2×2 determinants: dynamics and average-case analysis. In *Algorithms—ESA'97 (Graz)*, pp. 486–499. – Springer, Berlin, 1997.

[45] Vallée (Brigitte). – Dynamique des fractions continues à contraintes périodiques. *Journal of Number Theory*, vol. 72, n° 2, 1998, pp. 183–235.

[46] Vallee (Brigitte). – Digits and continuants in Euclidean algorithms. Ergodic versus Tauberian theorems. *Journal de Théorie des Nombres de Bordeaux*, vol. 12, n° 2, 2000, pp. 531–570. – Colloque international de Théorie des Nombres (Talence, 1999).

[47] Vallée (Brigitte). – Dynamical analysis of a class of Euclidean algorithms. *Theoretical Computer Science*, vol. 297, n° 1-3, 2003, pp. 447–486. – Latin American theoretical informatics (Punta del Este, 2000).

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 151–166.

Available online at the URL
http://algo.inria.fr/seminars/.

Martingales discrètes et applications à l'analyse d'algorithmes[†]

Brigitte Chauvin
Université de Versailles–Saint-Quentin (France)

March 20 and 21, 2002

Summary by Brigitte Chauvin

1. Les martingales discrètes

1.1. **Définitions.** Soit $(\Omega, \mathcal{A}, \mathbf{P})$ un espace probabilisé. Une filtration est une suite croissante de sous-tribus de $\mathcal{A}$.

Définition 1. Soit $(\mathcal{F}_n)$ une filtration. Une suite de variables aléatoires réelles (X_n) est une *$\mathcal{F}_n$-martingale* si pour tout n :

1. X_n est $\mathcal{F}_n$-mesurable (on dit que la suite (X_n) est *adaptée*)
2. X_n est intégrable : $\mathbf{E}\,|X_n| < \infty$
3. $\mathbf{E}(X_{n+1} \,/\, \mathcal{F}_n) = X_n$, p. s.

Le mot « martingale » vient du cas, au siècle de Pascal, où X_n représente la fortune d'un joueur après la $n$ième partie et $\mathcal{F}_n$ représente son information à propos du jeu à ce moment-là. L'égalité du point 3 de la définition dit que sa fortune espérée après la prochaine partie est la même que sa fortune actuelle. Une martingale est ainsi un jeu équitable.

Définition 2. Si le point 3 est remplacé par $\mathbf{E}(X_{n+1} \,/\, \mathcal{F}_n) \leq X_n$, p. s., on obtient une *surmartingale*, le jeu est défavorable pour le joueur. S'il est remplacé par $\mathbf{E}(X_{n+1} \,/\, \mathcal{F}_n) \geq X_n$, p. s., on obtient une *sous-martingale*.

Remarque. Si l'on n'a pas de filtration sous la main, on peut toujours prendre $\mathcal{F}_n = \sigma(X_1, \ldots, X_n)$, la σ-algèbre engendrée par $X_1, \ldots, X_n$.

Quelques conséquences.

1. Pour une martingale (X_n), la suite des espérances $(\mathbf{E}\,X_n)$ est constante, pour une surmartingale, $(\mathbf{E}\,X_n)$ est décroissante et pour une sous-martingale, $(\mathbf{E}\,X_n)$ est croissante.
2. Pour tout entier $k \geq 1$, on a $\mathbf{E}(X_{n+k} \,/\, \mathcal{F}_n) = X_n$ p. s. si (X_n) est une martingale, et de même $\mathbf{E}(X_{n+k} \,/\, \mathcal{F}_n) \leq X_n$ p. s. si (X_n) est une surmartingale, $\mathbf{E}(X_{n+k} \,/\, \mathcal{F}_n) \geq X_n$ p. s. si (X_n) est une sous-martingale.
3. Si (X_n) est une martingale, $\Delta X_n := X_n - X_{n-1}$ s'appelle *accroissement* de X_n. Alors $\mathbf{E}(\Delta X_n \,/\, \mathcal{F}_{n-1}) = 0$ et $X_n = X_0 + \sum_{k=1}^{n} \Delta X_k$. Inversement, on peut se donner des « différences de martingale » ΔX_n, avec la propriété $\mathbf{E}(\Delta X_n \,/\, \mathcal{F}_{n-1}) = 0$ et obtenir une martingale en posant $X_n = X_0 + \sum_{k=1}^{n} \Delta X_k$. Ces deux modes d'exposition sont fréquemment utilisés.

1.2. **Comment trouver des martingales ?**

[†]Notes de cours pour le cours donné pendant le groupe de travail ALÉA'02 au Cirm à Luminy (France).

1.2.1. *Convexité.* Si (M_n) est une martingale et si φ est une fonction convexe de $\mathbb{R}$ dans $\mathbb{R}$, alors $\varphi(M_n)$ est une sous-martingale. Par exemple, dès que (M_n) est une martingale, (M_n^2) est une sous-martingale.

1.2.2. *Compensateur.* Partons d'une suite (X_n), adaptée, d'accroissement $\Delta X_n := X_n - X_{n-1}$, et posons

$$\Delta \tilde{X}_n := \mathbf{E}(X_n - X_{n-1} \,/\, \mathcal{F}_{n-1}) = \mathbf{E}(X_n \,/\, \mathcal{F}_{n-1}) - X_{n-1}.$$

A priori, pour une suite quelconque (X_n), cette quantité n'est pas nulle, c'est le « défaut de martingale ». Il s'ensuit que la suite $(\tilde{X}_n)$ définie par $\tilde{X}_0 = 0$ et

$$\tilde{X}_n = \Delta \tilde{X}_1 + \cdots + \Delta \tilde{X}_n$$

est $\mathcal{F}_{n-1}$-mesurable (on dit qu'elle est *prévisible*), et

$$\begin{aligned}\mathbf{E}(X_n - \tilde{X}_n \,/\, \mathcal{F}_{n-1}) &= \mathbf{E}(X_{n-1} + \Delta \tilde{X}_n - \tilde{X}_n \,/\, \mathcal{F}_{n-1}) \\ &= X_{n-1} + \mathbf{E}(-\tilde{X}_{n-1} \,/\, \mathcal{F}_{n-1}) = X_{n-1} - \tilde{X}_{n-1},\end{aligned}$$

donc la suite $(X_n - \tilde{X}_n)$ est une martingale. On dit que $(\tilde{X}_n)$ est le « compensateur » de (X_n). Le compensateur est l'unique suite prévisible, nulle en 0, telle que $(X_n - \tilde{X}_n)$ soit une martingale (s'il y en avait une autre, la différence serait une martingale prévisible et nulle en 0, c'est-à-dire nulle).

Cas particulier important : la décomposition de Doob. Si l'on applique la méthode du compensateur ci-dessus à une sous-martingale (X_n), on obtient

$$\Delta \tilde{X}_n = \mathbf{E}(X_n \,/\, \mathcal{F}_{n-1}) - X_{n-1} \geq 0$$

c'est-à-dire que le compensateur est un processus croissant. Autrement dit, toute sous-martingale se décompose de façon unique en la somme d'une martingale M_n et d'un processus croissant noté A_n :

$$X_n = M_n + A_n$$

avec $A_0 = 0$ et

$$A_n = \sum_{k=1}^{n} \mathbf{E}(X_k - X_{k-1} \,/\, \mathcal{F}_{k-1}).$$

Dans le cas encore plus particulier où l'on part d'une martingale (M_n), ceci s'applique à la sous-martingale (M_n^2) qui se décompose donc

$$M_n^2 = \text{martingale} + \langle M \rangle_n$$

où $\langle M \rangle_n$ est la notation habituellement utilisée pour le processus croissant de (M_n^2). On appelle *processus croissant d'une martingale* (M_n) le processus croissant de la décomposition de Doob de la sous-martingale (M_n^2).

1.2.3. *Exemple de base : les sommes de variables i. i. d.* Soit (X_n) une suite de variables aléatoires indépendantes et équidistribuées (i. i. d.) de moyenne $m = \mathbf{E}(X)$ et de variance $\sigma^2 = \mathbf{Var}(X)$. On s'intéresse au comportement asymptotique de

$$S_n := X_1 + \cdots + X_n.$$

On constate immédiatement que

$$M_n := S_n - n\,\mathbf{E}(X)$$

est une $\mathcal{F}_n$-martingale. Cherchons son processus croissant :

$$\langle M\rangle_n = \sum_{k=1}^{n} \mathbf{E}\big(M_k^2 - M_{k-1}^2 \ / \ \mathcal{F}_{k-1}\big) = \sum_{k=1}^{n} \mathbf{E}\big((M_k - M_{k-1})^2 \ / \ \mathcal{F}_{k-1}\big)$$
$$= \sum_{k=1}^{n} \mathbf{E}\Big(\big(X_k - \mathbf{E}(X)\big)^2 \ \Big/ \ \mathcal{F}_{k-1}\Big) = n\sigma^2.$$

Finalement, dans ce cas des sommes de variables i. i. d., la décomposition de Doob s'écrit :

$$(S_n - nm)^2 = \text{martingale} + n\sigma^2.$$

1.2.4. *Renormalisation.* Supposons que la suite de variables aléatoires (X_n) adaptée vérifie

$$\mathbf{E}(X_n \,/\, \mathcal{F}_{n-1}) = A_{n-1}X_{n-1}$$

où A_{n-1} est $\mathcal{F}_{n-1}$-mesurable et différent de 0 presque sûrement. Il suffit alors de renormaliser la suite (X_n) pour obtenir une $\mathcal{F}_n$-martingale

$$Y_n := \frac{X_n}{\prod_{k=0}^{n-1} A_k}.$$

Exemple (Arbre de Galton–Watson). Prenons le plus simple des processus de branchement : le processus de Galton–Watson, dans lequel un ancêtre à l'instant 0 donne naissance à k individus à l'instant 1 avec probabilité p_k. Puis les individus continuent à se reproduire à des instants discrets, indépendamment les uns des autres et toujours suivant la loi (p_k). On suppose que la moyenne m de cette loi est finie :

$$m := \sum_{k\geq 0} kp_k < +\infty.$$

On s'intéresse au comportement asymptotique du nombre Z_n de nœuds à la génération n.

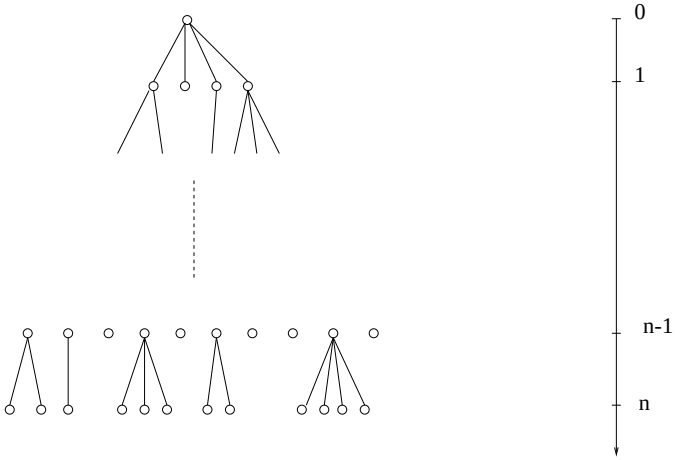


FIGURE 1. Arbre de Galton–Watson.

À tout instant n,

$$Z_n = \sum_{|u|=n-1} N_u$$

où N_u désigne le nombre de descendants de u, de sorte que

$$\mathbf{E}(Z_n \,/\, \mathcal{F}_{n-1}) = \sum_{|u|=n-1} \mathbf{E}(N_u \,/\, \mathcal{F}_{n-1}) = mZ_{n-1} \ ;$$

en renormalisant, on obtient que

$$M_n := \frac{Z_n}{m^n}$$

est une martingale.

1.2.5. *Transformée.* Soit (M_n) une martingale et soit (c_n) une suite de variables aléatoires $\mathcal{F}_{n-1}$-mesurables (prévisibles). Les (c_n) peuvent évidemment parfois être réduites à des constantes. Les accroissements de martingale sont $\Delta M_n = M_n - M_{n-1}$ et il est immédiat que

$$Z_n := c_0 M_0 + c_1 \Delta M_1 + \cdots + c_n \Delta M_n$$

est encore une martingale. Z_n s'appelle la « transformée » de M_n.

1.2.6. *Chaines de Markov.* Partons d'une chaine de Markov (X_n) à temps discret, à valeurs dans un espace d'états S dénombrable et de matrice de transition $P = (p_{i,j})$:

$$\mathbf{P}(X_{n+1} = j \,/\, X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \mathbf{P}(X_{n+1} = j \,/\, X_n = i) = p_{i,j}$$

Soit $\psi : S \longrightarrow S$ une fonction harmonique au sens où

$$\forall i \in S, \qquad \sum_{j \in S} p_{i,j} \psi(j) = \psi(i).$$

Alors $\big(\psi(X_n)\big)_{n\in\mathbb{N}}$ est une martingale pour la filtration associée à (X_n), car

$$\begin{aligned}\mathbf{E}\big(\psi(X_{n+1}) \ / \ X_1, X_2, \ldots, X_n\big) &= \mathbf{E}\big(\psi(X_{n+1}) \ / \ X_n\big) \\ &= \sum_{j\in S} \mathbf{E}\big(\psi(j)\, 1_{\{X_{n+1}=j\}} \ / \ X_n\big) = \sum_{j\in S} \psi(j)\, \mathbf{P}(X_{n+1} = j \,/\, X_n) = \sum_{j\in S} p_{X_n,j}\psi(j) = \psi(X_n).\end{aligned}$$

Dans le cas un peu plus général où ψ est un vecteur propre de la matrice de transition P, pour la valeur propre λ, c'est-à-dire

$$\forall i \in S, \qquad \sum_{j \in S} p_{i,j} \psi(j) = \lambda \psi(i),$$

alors

$$\mathbf{E}\big(\psi(X_{n+1}) \ / \ X_1, X_2, \ldots, X_n\big) = \lambda \psi(X_n),$$

et en renormalisant, on obtient que

$$M_n := \frac{\psi(X_n)}{\lambda^n}$$

est une martingale.

1.3. **Règles d'arrêt.**

Définition 3. Un *temps d'arrêt* T pour la filtration $(\mathcal{F}_n)$ est une variable aléatoire à valeurs dans $\mathbb{N} \cup \{+\infty\}$ telle que pour tout entier n, l'événement $\{T \le n\}$ est $\mathcal{F}_n$-mesurable (c'est la même chose que de demander que pour tout n, $\{T = n\}$ soit $\mathcal{F}_n$-mesurable).

Si T est un temps d'arrêt, la tribu « arrêtée » à T est

$$\mathcal{F}_T := \big\{\, A \in \mathcal{F}_\infty \ \big|\ \forall n \in \mathbb{N},\ A \cap \{T = n\} \in \mathcal{F}_n \,\big\}.$$

Exemple 1. $T =$ constante.

Exemple 2. Les temps d'atteinte ou temps de premier passage d'une suite adaptée (X_n) par un borélien : soit B un borélien, posons

$$T := \inf \{ n \in \mathbb{N} \mid X_n \in B \} ;$$

alors

$$\{T = n\} = \{X_1 \notin B, \ldots, X_{n-1} \notin B, X_n \in B\} \in \mathcal{F}_n.$$

On se demande dans la suite si la propriété de martingale peut passer à des instants aléatoires ayant la propriété de temps d'arrêt. Il est commode de noter $a \wedge b$ pour le minimum de deux réels a et b.

Théorème 1 (Premier théorème d'arrêt)**.** 1. *Si (X_n) est une martingale et si T est un temps d'arrêt, alors*

$$Y_n := X_{T \wedge n}$$

est une martingale appelée martingale arrêtée au temps T.

2. *Si (X_n) est une martingale et si S et T sont deux temps d'arrêt* bornés *tels que $S < T$, alors*

$$\mathbf{E}(X_T \,/\, \mathcal{F}_S) = X_S$$

c'est-à-dire que l'on a la propriété de martingale à des instants aléatoires.

L'hypothèse « temps d'arrêt bornés » est très forte, nous allons chercher à la relaxer, après avoir vu comment convergent les martingales.

Ce théorème vaut également pour les sous-martingales et les surmartingales.

1.4. **Inégalités.** Les inégalités suivantes concernent le maximum d'une suite de variables aléatoires, et sont appelées *inégalités maximales de Doob.*

Proposition 1. *Soit (X_n) une* sous-*martingale* positive *telle que $\mathbf{E}(X_0) < \infty$, alors*

$$\mathbf{P}\left(\max_{k \leq n} X_k \geq \lambda\right) \leq \frac{\mathbf{E}(X_n)}{\lambda}.$$

Quelques remarques.

1. La démonstration consiste à introduire T, le temps d'atteinte de λ et à appliquer l'inégalité de Markov à $X_{T \wedge n}$.
2. Si l'on dispose seulement d'une sous-martingale de signe quelconque, il est possible d'appliquer la proposition à $X_n^+ = \max(0, X_n)$ qui est encore une sous-martingale.
3. Si (X_n) est une martingale de signe quelconque, la proposition s'applique à $|X_n|$ qui est une sous-martingale. Le même argument appliqué au carré d'une martingale conduit au corollaire suivant :

   Corollaire 1. *Soit (X_n) une martingale de carré intégrable. Alors*

$$\mathbf{P}\left(\max_{k \leq n} X_k \geq \lambda\right) \leq \frac{\mathbf{E}(X_n^2)}{\lambda^2}.$$

1.5. **Convergence.**

Définition 4. On dit qu'une suite de variables aléatoires (X_n) est *intégrable* (respectivement *de carré intégrable*) si et seulement si

$$\mathbf{E}\big(|X_n|\big) < \infty, \qquad \text{respectivement,} \quad \mathbf{E}(X_n^2) < \infty.$$

On dit qu'une suite de variables aléatoires (X_n) est *bornée dans* L^p $(p > 0)$, si et seulement si

$$\sup_n \mathbf{E}\big(|X_n|^p\big) < \infty.$$

On dit qu'une suite de variables aléatoires (X_n) est *équi-intégrable* ou *uniformément intégrable* si et seulement si

$$\sup_n \mathbf{E}\left(|X_n|\, 1_{\{|X_n|>a\}}\right) \longrightarrow 0$$

quand $a \to +\infty$.

1.5.1. *Convergence dans* L^2.

Théorème 2 (Convergence L^2 des martingales)**.** *Toute martingale (X_n) bornée dans L^2 converge dans L^2. Toute sous-martingale positive (X_n) bornée dans L^2 converge dans L^2.*

La démonstration de ce théorème repose sur la décomposition de Doob.

Ce théorème est simple à utiliser et il a de nombreuses applications : par exemple, la martingale de l'arbre de Galton–Watson surcritique $(m > 1)$ converge dans L^2, dès que la loi de reproduction a un second moment fini.

1.5.2. *Convergence presque sûre.*

Théorème 3 (Convergence p. s. des martingales, théorème de Doob)**.** *Toute sous-martingale (X_n) vérifiant $\sup_n \mathbf{E}(X_n^+) < \infty$ converge p. s. vers une variable aléatoire X_∞ et $X_\infty \in L^1$.*

Ce théorème admet de nombreux sous-produits, comme :

- toute sous-martingale (X_n), bornée dans L^1, converge p. s. vers une variable aléatoire X_∞ et on a $X_\infty \in L^1$;
- toute martingale (X_n), bornée dans L^1, converge p. s. vers une variable aléatoire X_∞ et on a $X_\infty \in L^1$.

Un autre de ses corollaires est particulièrement simple et efficace :

Corollaire 2. *Toute surmartingale positive (X_n) converge p. s. vers une variable aléatoire X_∞, $X_\infty \in L^1$ et $\mathbf{E}(X_\infty) \leq \liminf \mathbf{E}(X_n)$.*

Démonstration. Regarder $-X_n$ et utiliser Fatou. □

1.5.3. *Convergence dans* L^1.

Attention : les hypothèses « (X_n) bornée dans L^1 » et « (X_n) converge p. s. vers une limite $X_\infty \in L^1$ » ne suffisent pas à assurer que X_n converge dans L^1. Il y a de nombreux contre-exemples, les martingales exponentielles de la section 2.1 en sont un.

Théorème 4 (Convergence L^1 des martingales)**.** *Soit (X_n) une martingale. Les trois assertions suivantes sont équivalentes :*

1. *(X_n) converge dans L^1 (vers une variable aléatoire $X_\infty \in L^1$) ;*
2. *(X_n) est bornée dans L^1 et $\mathbf{E}(X_\infty \,/\, \mathcal{F}_n) = X_n$;*
3. *(X_n) est uniformément intégrable.*

Une martingale vérifiant l'une de ces propriétés est dite régulière. *Pour une martingale régulière, on aura en particulier $\mathbf{E}(X_\infty) = \mathbf{E}(X_0)$.*

Il arrive que l'on n'ait pas convergence dans L^2 et que la convergence L^1 soit suspectée mais difficile à obtenir via le théorème précédent (c'est le cas pour des processus de branchement et pour les arbres binaires de recherche par exemple). Une convergence L^p pour $p \in]\,1, 2\,[$ pourra alors être utile (outre son intérêt propre).

Corollaire 3 (Convergence L^p, $p > 1$, des martingales)**.** *Toute martingale (X_n) bornée dans L^p pour $p > 1$ converge dans L^p (et aussi en probabilité et p. s. par le théorème de Doob).*

Démonstration. On montre que $|X_n|^p$ est uniformément intégrable. □

1.5.4. *Temps d'arrêt et martingales régulières.*

Définition 5 (Temps d'arrêt régulier)**.** On dit qu'un temps d'arrêt T est *régulier* pour la martingale M_n quand la martingale arrêtée $M_{T\wedge n}$ est régulière.

Cette nouvelle notion permet de raffiner le premier théorème d'arrêt, en affaiblissant les conditions sur le temps d'arrêt :

Théorème 5 (Second théorème d'arrêt)**.** *Soit (M_n) une martingale dans L^1 (mais pas nécessairement régulière), soient T_1 et T_2 deux temps d'arrêt avec T_2 régulier. Alors*

$$\mathbf{E}(M_{T_2} \,/\, \mathcal{F}_{T_1}) = M_{T_1} \quad \textit{sur} \quad \{T_1 \le T_2\}.$$

Le corollaire suivant est évidemment utile en pratique :

Corollaire 4 (Identité de Wald)**.** *Soit (M_n) une martingale et soit T un temps d'arrêt régulier. Alors,*

$$\mathbf{E}(M_n) = \mathbf{E}(M_T)$$

1.6. **Théorème central limite.** Dans l'exemple de base où la martingale considérée est la somme de v. a. X_i i. i. d., $M_n = S_n - n\,\mathbf{E}(X)$, le théorème central limite classique s'applique. Dans une variante de l'exemple de base, la martingale considérée est la somme de v. a. X_i indépendantes mais pas nécessairement de même loi. Dans ce cas (voir par exemple [14]), le théorème central limite est valide sous une condition de type « condition de Lindeberg ». Il est aussi valide sous une condition un peu plus forte, de moment d'ordre $2 + \delta$, dite « condition de Lyapunov ».

Le cas des martingales est complètement analogue au cas des sommes de v. a. indépendantes et c'est le processus croissant de la martingale qui joue le rôle de la variance (*cf.* [6, 8]).

Théorème 6 (Théorème central limite pour les martingales, version Lindeberg)**.** *Soit (M_n) une martingale centrée et de carré intégrable. Sous les deux conditions*

1. $$\frac{\langle M\rangle_n}{\mathbf{E}(M_n^2)} \longrightarrow \Gamma \quad \textit{en probabilité}$$
   où Γ est une v. a. positive, finie p. s., et
2. *(condition de Lindeberg)*

$$\forall \varepsilon > 0, \quad \frac{1}{\mathbf{E}(M_n^2)} \sum_{k=1}^{n} \mathbf{E}\Big((M_k - M_{k-1})^2\, 1_{\left\{|M_k - M_{k-1}| > \varepsilon \sqrt{\mathbf{E}(M_n^2)}\right\}} \;\Big/\; \mathcal{F}_{k-1} \Big) \longrightarrow 0 \quad \textit{en probabilité},$$

on a

$$\frac{M_n}{\sqrt{\mathbf{E}(M_n^2)}} \longrightarrow \mathcal{N}(0, \Gamma).$$

On note $\mathcal{N}(0,\Gamma)$ pour un mélange de lois normales, au sens où on dira qu'une v. a. Z a pour loi $\mathcal{N}(0,\Gamma)$ si et seulement si sa fonction caractéristique est donnée par

$$\mathbf{E}\left(e^{-itZ}\right) = \mathbf{E}\left(\exp -\frac{1}{2}\Gamma t^2\right).$$

La version plus simple du théorème central limite pour les martingales, sous condition de moment $2+\delta$ est :

Théorème 7 (Théorème central limite pour les martingales, version Lyapunov)**.** *Soit (M_n) une martingale centrée et de carré intégrable. Sous les deux conditions*

1. $$\frac{\langle M\rangle_n}{\mathbf{E}(M_n^2)} \longrightarrow \Gamma \quad \textit{en probabilité}$$
   où Γ est une v. a. positive, finie p. s., et
2. *(condition de Lyapunov)*
$$\frac{1}{\left(\mathbf{E}(M_n^2)\right)^{1+\delta/2}} \sum_{k=1}^{n} \mathbf{E}\left((M_k - M_{k-1})^{2+\delta}\right) \longrightarrow 0 \quad \textit{en probabilité,}$$

on a

$$\frac{M_n}{\sqrt{\mathbf{E}(M_n^2)}} \longrightarrow \mathcal{N}(0,\Gamma).$$

Dans les cas particuliers suivants, les accroissements de martingale vérifient à chaque fois une condition plus forte que la condition de Lindeberg et le théorème central limite s'appliquera :

- condition 2 remplacée par : $\max_k \left(|M_k - M_{k-1}|/\sqrt{\mathbf{E}(M_n^2)}\right) \longrightarrow 0$ en probabilité ;
- condition 2 remplacée par : $\mathbf{E}\left(\max_k (M_k - M_{k-1})^2\right) / \mathbf{E}(M_n^2)$ uniformément bornée ;
- condition 2 remplacée par : $|M_k - M_{k-1}|$ uniformément bornés (très fort).

Enfin, le théorème suivant est une version affaiblie, mais simple à utiliser.

Théorème 8. *Soit (M_n) une martingale centrée et de carré intégrable. Supposons qu'il existe une constante K telle que pour tout n,*

$$\mathbf{E}\Big(|M_{n+1} - M_n|^{2+\delta} \;\Big/\; \mathcal{F}_n\Big) \leq K.$$

Si la suite $\left(\langle M\rangle_n/n\right)$ converge en probabilité vers une constante σ^2, alors $\frac{M_n}{\sqrt{n}}$ converge en loi vers une variable gaussienne $\mathcal{N}(0,\sigma^2)$.

2. Application aux martingales exponentielles

2.1. **Une famille de martingales non régulières.** Soit (X_i) une suite de variables aléatoires indépendantes et équidistribuées (i. i. d.) à valeurs réelles, de loi μ qui ne soit pas une mesure concentrée en un seul point. On suppose que la transformée de Laplace de μ existe au moins sur un voisinage ouvert de 0 et on appelle l son logarithme :

$$e^{l(u)} := \int_{\mathbb{R}} e^{ux}\, d\mu(x).$$

On s'intéresse au comportement asymptotique de

$$S_n := X_1 + \cdots + X_n.$$

dont on va voir qu'il est très lié aux martingales exponentielles

$$M_n(u) := \exp\bigl(uS_n - nl(u)\bigr).$$

On peut comprendre heuristiquement le comportement de $M_n(u)$: quand n est grand, par la loi des grands nombres, $\frac{S_n}{n} \sim \mathbf{E}\,X$, et comme les moments de X se lisent avec la transformée de Laplace, $\mathbf{E}\,X = l'(0)$, on a $S_n \sim nl'(0)$, ce qui, en remplaçant dans $M_n(u)$ donne

$$M_n(u) \sim \exp\Bigl(nu\bigl(l'(0) - l(u)/u\bigr)\Bigr).$$

Comme $l(u)$ est convexe, la quantité $l'(0) - l(u)u$ est négative, ce qui indique que $M_n(u)$ converge exponentiellement vite vers 0. La démonstration rigoureuse de la proposition suivante repose effectivement sur la convexité de $l(u)$.

Proposition 2. *Pour tout réel u appartenant à l'ouvert de définition de $l(u)$,*

$$M_n(u) := \exp\bigl(uS_n - nl(u)\bigr)$$

est une martingale positive, d'espérance 1*, qui converge p. s. vers* 0*. Ce n'est pas une martingale régulière.*

Démonstration. Laissée en exercice. □

Exercice. 1. La fonction $u \mapsto f(u) := \exp\bigl(ux - nl(u)\bigr)$, pour x et n fixés, est analytique sur V, voisinage de 0 où $l(u)$ est analytique. Son développement est

$$\exp\bigl(ux - nl(u)\bigr) = \sum_{k=0}^{\infty} \frac{u^k}{k!} f_k(n, x).$$

Montrer que pour tout entier $k \geq 0$, la suite $\bigl(f_k(n, S_n)\bigr)_n$ est une $\mathcal{F}_n$-martingale intégrable.

2. Regarder ce que l'on obtient pour $k = 1$ et $k = 2$.
3. Montrer que

$$Y_n := \int_{\mathbb{R}} \exp\bigl(uS_n - nl(u)\bigr)\, du$$

est encore une martingale positive donc p. s. convergente vers une v. a. finie p. s. Qu'obtient-on dans le cas particulier où les X_i suivent des lois normales centrées réduites ?

2.2. ***Branching random walks.*** On trouve aussi des martingales exponentielles dans les marches aléatoires branchantes (*branching random walks*), puisque des v. a. i. i. d. s'ajoutent le long des branches de l'arbre. Rappelons que dans ce processus, un ancêtre (noté $\emptyset$) se trouve en 0 (ou ailleurs) à l'instant 0. Ses enfants forment la première génération et leurs positions (ainsi que leur nombre N) sont données par un processus ponctuel Z sur $\mathbb{R}$. Ensuite, chaque particule u donne naissance, indépendamment des autres particules et du passé, à des enfants selon un processus ponctuel copie de Z.

L'espace de probabilité considéré est celui des arbres marqués par les déplacements γ_u des particules. On définit la position X_u de la particule u par

$$X_u = X_\emptyset + \gamma_{i_1} + \gamma_{i_1 i_2} + \cdots + \gamma_{i_1 \ldots i_n}$$

pour $u = i_1 i_2 \ldots i_n$ avec $i_j \in \mathbb{N}^*$.

Si l'on appelle maintenant μ la mesure d'intensité du processus ponctuel Z, c'est-à-dire que pour toute fonction f mesurable positive

$$\mathbf{E}\bigl(Z(f)\bigr) = \mathbf{E}\Bigl(\sum_{j=1}^{N} f(X_j)\Bigr) = \int f(x)\mu(dx),$$

supposons que sa transformée de Laplace soit définie sur un voisinage de 0 et notons-la $m(\theta)$:

$$m(\theta) = \mathbf{E}\Big(\sum_{j=1}^{N} e^{\theta X_j}\Big) = \int e^{\theta x}\mu(dx) = \mathbf{E}\left(Z\big(e^{\theta\cdot}\big)\right),$$

son logarithme est toujours noté $l(\theta) = \log m(\theta)$.

Théorème 9 (Kingman, 1975)**.** *Pour tout θ,*

$$W_n(\theta) := \sum_{|u|=n} \exp\big(\theta X_u - nl(\theta)\big)$$

est une $\mathcal{F}_n$-martingale positive, d'espérance 1*. Elle converge presque sûrement vers $W(\theta)$ avec* a priori $\mathbf{E}\big(W(\theta)\big) \le 1$ *(par le corollaire 2).*

Remarquons qu'en faisant $\theta = 0$, on retrouve le processus de Galton–Watson sous-jacent.

C'est l'étude plus fine de cette martingale (*cf.* [2, 4]), qui permettra d'obtenir des résultats sur la particule la plus à gauche dans ce processus. La connection de Devroye entre un arbre binaire de recherche et une marche aléatoire branchante fournira la convergence presque sûre de la hauteur d'un arbre binaire de recherche.

2.3. **Utilisation de martingales exponentielles pour l'étude des queues de distribution de S_n. Identité de Wald.** Pour un réel a donné, l'objectif est d'étudier $\mathbf{P}(S_n \ge a)$. L'idée conductrice est d'appliquer le second théorème d'arrêt (Théorème 5) à la martingale exponentielle $M_n(u)$ pour le temps d'arrêt ν_a qui est le temps d'atteinte de a

$$\nu_a := \min\{\, n \in \mathbb{N} \mid S_n \ge a \,\}.$$

Pour cela, il faut voir si ce temps d'arrêt est régulier.

Lemme 1. *Pour tout u tel que $l'(u) > 0$, le temps d'arrêt ν_a est régulier pour la martingale $M_n(u)$.*

Démonstration. La démonstration repose sur un changement de probabilité classique en théorie des grandes déviations.

On doit montrer que $M_{n\wedge\nu_a}(u)$ (dont on sait déjà que c'est une martingale par le premier théorème d'arrêt) est une martingale régulière, c'est-à-dire qu'elle converge dans L^1. Écrivons

$$M_{n\wedge\nu_a}(u) = M_n(u)\,1_{\{\nu_a > n\}} + M_{\nu_a}(u)\,1_{\{\nu_a \le n\}}$$

et montrons que chacun des deux termes converge dans L^1.

Pour le second terme, il suffit de montrer par convergence dominée que

$$\mathbf{E}\left(M_{\nu_a}(u)\,1_{\{\nu_a<\infty\}}\right) < \infty.$$

Mais par le premier théorème d'arrêt, la propriété de martingale de $M_{n\wedge\nu_a}(u)$ donne

$$M_{\nu_a}(u) = M_{n\wedge\nu_a}(u) = \mathbf{E}(M_n \,/\, \mathcal{F}_{\nu_a}) \qquad \text{sur } \{\nu_a \le n\}$$

et donc

$$\mathbf{E}\left(M_{\nu_a}(u)\,1_{\{\nu_a\le n\}}\right) = \mathbf{E}\left(1_{\{\nu_a\le n\}}\,\mathbf{E}(M_n \,/\, \mathcal{F}_{\nu_a})\right) \le \mathbf{E}(M_n) = \text{constante} < \infty.$$

Pour le premier terme, on va montrer qu'il converge vers 0 dans L^1. Comme la martingale est positive, on va montrer que

$$\mathbf{E}\left(M_n(u)\,1_{\{\nu_a>n\}}\right) \longrightarrow 0.$$

Écrivons

$$\mathbf{E}\left(M_n(u)\,1_{\{\nu_a>n\}}\right) = \mathbf{E}\left(e^{u(X_1+\cdots+X_n)-nl(u)}\,1_{\{\forall m<n,\ S_m<a\}}\right)$$
$$= \int_{\mathbb{R}^n} e^{ux_1-l(u)}\,d\mu(x_1)\ldots e^{ux_n-l(u)}\,d\mu(x_n)\,1_{\{\max_{m<n} S_m<a\}},$$

d'où l'idée de changer de probabilité en posant pour tout u dans le bon voisinage de 0

$$d\mu_u(x) = e^{ux-l(u)}d\mu(x)$$

de sorte que

$$\mathbf{E}\left(M_n(u)\,1_{\{\nu_a>n\}}\right) = \int_{\mathbb{R}^n} d\mu_u(x_1)\ldots d\mu_u(x_n)\,1_{\{\max_{m<n} S_m<a\}} = \mathbf{P}_u\left(\max_{m<n} S_m < a\right)$$

où $S_m = X_1+\cdots+X_m$ pour des v. a. X_i i. i. d. de loi μ_u. Par la loi des grands nombres, quand $n\to\infty$,

$$\frac{S_n}{n} \longrightarrow \mathbf{E}_u(X) = l'(u) \quad \text{p. s.}$$

donc si $l'(u)>0$, S_n tend vers $+\infty$, $\mathbf{P}_u$-p. s. et la probabilité ci-dessus tend vers 0. □

Puisque d'après ce lemme, le temps d'arrêt ν_a est régulier, la martingale est d'espérance constante égale à 1, y compris au temps ν_a. On a ainsi démontré :

Proposition 3. *Pour tout u tel que $l'(u)>0$, le temps d'arrêt ν_a est régulier pour la martingale $M_n(u)$ et on a l'*identité de Wald

$$\mathbf{E}\,M_{\nu_a}(u) = 1$$

et comme cette martingale converge presque sûrement vers 0*, l'identité de Wald s'écrit aussi*

$$\mathbf{E}\left(M_{\nu_a}(u)\,1_{\nu_a<\infty}\right) = 1$$

Commentaire. Dans les cas où l'on a $S_{\nu_a}=a$, l'identité de Wald s'écrit

$$\mathbf{E}\exp\bigl(uS_{\nu_a}-\nu_a l(u)\bigr) = \mathbf{E}\exp\bigl(ua-\nu_a l(u)\bigr) = 1,$$

ce qui fournit une formule exacte pour $\mathbf{E}\left(e^{-\nu_a l(u)}\right)$, c'est-à-dire pour la transformée de Laplace du temps d'arrêt ν_a.

3. Martingales à horizon fini et inégalité d'Azuma

L'objectif est de montrer des inégalités de concentration de variables aléatoires X_n autour de leur moyenne $\mathbf{E}\,X_n$. Pour cela on va construire une martingale artificielle, seulement pour les instants $0,1,\ldots,n$ (on parle de martingale à horizon fini), de sorte que X_n soit la martingale à l'instant n et $\mathbf{E}\,X_n$ soit la martingale à l'instant 0. Puis on utilisera le théorème de concentration suivant pour les martingales ([12] est une référence agréable sur ce sujet).

Théorème 10 (Inégalité d'Hoeffding–Azuma)**.** *Soit M_n une martingale, donnée par les accroissements de martingale ΔM_n :*

$$M_n = M_0 + \Delta M_1 + \cdots + \Delta M_n.$$

Supposons qu'il existe une suite de constantes $(c_n)_{n\in\mathbb{N}}$ telle que

$$\forall k\in\mathbb{N}, \qquad |\Delta M_k| \le c_k \qquad \text{presque sûrement} \tag{1}$$

Alors, pour tout réel t,

$$\mathbf{P}\bigl(|M_n-M_0|\ge t\bigr) \le 2\exp\left(-\frac{t^2}{2\sum_{k=1}^n c_k^2}\right)$$

Exercice. Appliquer le théorème précédent pour montrer que pour une suite de variables aléatoires i. i. d. de loi uniforme sur $[\,0,1\,]$, on a l'inégalité de concentration :

$$\mathbf{P}\big(|S_n - n/2| > t\big) \leq 2e^{-\frac{2t^2}{n}}$$

Voyons sur un exemple d'application comment la méthode annoncée plus haut peut fonctionner.

Une application au bin packing. Appelons B_n le nombre de boîtes de taille 1 nécessaires pour ranger n objets de taille $X_1, X_2, \ldots, X_n$ i. i. d. de loi uniforme sur $[\,0,1\,]$. On souhaite étudier la variable aléatoire B_n. On sait par exemple que $\frac{B_n}{n}$ converge presque sûrement vers une constante. On cherche une inégalité de concentration de B_n autour de sa moyenne. Pour cela, on construit

$$\forall i = 0, 1, \ldots, n \qquad Y_i^{(n)} := \mathbf{E}(B_n \,/\, \mathcal{F}_i).$$

Alors $\left(Y_i^{(n)}\right)_{i=0,1,\ldots,n}$ est une $\mathcal{F}_i$-martingale et

$$Y_n^{(n)} = B_n, \qquad Y_0^{(n)} = \mathbf{E}(B_n).$$

Il suffit donc de montrer que les accroissements de cette martingale (Y_i) vérifient la condition (1) pour en déduire grâce au théorème d'Azuma une majoration de

$$\mathbf{P}\left(\left|Y_n^{(n)} - Y_0^{(n)}\right| > t\right) = \mathbf{P}\big(|B_n - \mathbf{E}(B_n)| > t\big).$$

Pour cela, remarquons que si l'on note $B_n(i)$ le nombre de boîtes nécessaires pour ranger les objets sauf X_i, on a toujours

$$B_n(i) \leq B_n \leq B_n(i) + 1$$

ce qui, en passant aux espérances conditionnelles par rapport à $\mathcal{F}_{i-1}$ et $\mathcal{F}_i$ donne

$$\mathbf{E}\big(B_n(i) \;/\; \mathcal{F}_{i-1}\big) \leq Y_{i-1} \leq \mathbf{E}\big(B_n(i) \;/\; \mathcal{F}_{i-1}\big) + 1$$

$$\mathbf{E}\big(B_n(i) \;/\; \mathcal{F}_i\big) \leq Y_i \leq \mathbf{E}\big(B_n(i) \;/\; \mathcal{F}_i\big) + 1.$$

Mais par définition de $B_n(i)$, les membres de gauche sont égaux, les membres de droite aussi. Donc

$$|Y_i - Y_{i-1}| \leq 1$$

ce qui est une condition de type (1) avec les constantes toutes égales à 1. L'inégalité d'Azuma fournit ainsi

$$\mathbf{P}\left(\left|Y_n^{(n)} - Y_0^{(n)}\right| > t\right) = \mathbf{P}\big(|B_n - \mathbf{E}(B_n)| > t\big) \leq 2e^{-\frac{t^2}{2n}}.$$

4. **Application aux arbres binaires de recherche**

Une référence sur le sujet est le livre de Mahmoud [10]. Rappelons d'abord quelques généralités sur Quicksort et les arbres binaires de recherche.

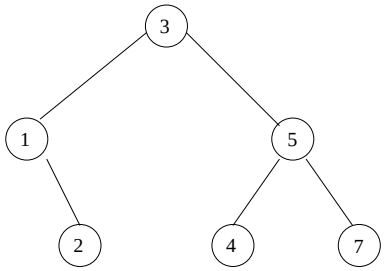
4.1. **Généralités.** Un *arbre binaire de recherche* est un arbre binaire dont chaque nœud interne est muni d'une « clé » (d'une étiquette, d'une marque, qui est dans un premier temps un entier) de telle sorte qu'à chaque nœud de l'arbre, toutes les clés du sous-arbre droit sont plus grandes que toutes les clés du sous-arbre gauche.

La définition (récursive) d'un *arbre binaire de recherche* est aussi la suivante : donnons-nous $x_1, x_2, \ldots x_n$ réels distincts. L'arbre binaire de recherche (abrégé abr) est l'arbre binaire dans lequel

- x_1 est à la racine,
- le sous-arbre gauche est l'abr associé à $\{x_2, \ldots x_n\} \cap]-\infty, x_1\,[$,

– le sous-arbre droit est l'abr associé à $\{x_2, \dots x_n\} \cap]\, x_1, +\infty\, [$.

Exemple. Tri de 3, 1, 5, 4, 7, 2.



Remarque. La lecture de l'arbre en ordre infixe donne la liste triée.

Insertion. Si l'on veut insérer une $(n+1)$ième donnée x_{n+1}, on la compare à la racine, puis à la clé racine du sous-arbre gauche (ou droit), etc, on ne la compare pas à toutes les clés mais seulement le long d'une branche, jusqu'à l'insérer sur un nœud externe. Appelons D_{n+1} (qui est donc un coût) le niveau d'insertion de la $(n+1)$ième donnée.

Modèle probabiliste. À tout ensemble de n données distinctes $x_1, x_2, \dots x_n$, on associe un arbre binaire de recherche T_n, à n nœuds internes et $n+1$ nœuds externes. Le modèle que nous étudions dans la suite, c'est-à-dire la loi de probabilité sur les arbres est l'image (par cette association) de la *loi uniforme sur les permutations* de n objets. Dans ce modèle, les arbres (T_n) ont même loi que ceux construits en choisissant $x_1, x_2, \dots x_n$ selon une densité uniforme sur l'intervalle $[\,0, 1\,]$. Dans ce modèle, l'insertion d'une $(n+1)$ième clé a la même probabilité d'être effectuée à chacun des $n+1$ nœuds externes de T_n.

4.2. **Répartition des nœuds externes, largeur d'un abr.**

4.2.1. *Mesure empirique. Polynôme de niveau.* Outre la hauteur et le niveau de saturation, si l'on veut connaître plus finement la répartition des nœuds par niveau dans l'arbre binaire de recherche, on introduit naturellement le nombre $U_k(n)$ de nœuds externes au niveau k dans l'arbre T_n. On pourrait travailler de façon analogue avec les nœuds internes et introduire le nombre $V_k(n)$ de nœuds internes au niveau k dans l'arbre T_n ainsi que le nombre total $Z_k(n) = U_k(n) + V_k(n)$ de nœuds au niveau k dans l'arbre T_n. Comme il y a $n+1$ nœuds externes dans l'arbre T_n de taille n, la mesure empirique de répartition des nœuds externes est

$$\nu_n := \sum_{k=1}^{+\infty} \frac{U_k(n)}{n+1} \delta_{\{k\}}.$$

Comme la $(n+1)$ième insertion se fait uniformément sur les $n+1$ nœuds externes de l'arbre T_n, la loi du niveau d'insertion, ou profondeur D_n est donnée par

$$\mathbf{P}(D_{n+1} = k \,/\, T_n) = \frac{U_k(n)}{n+1} = \nu_n(k), \tag{2}$$

et en prenant l'espérance

$$\mathbf{P}(D_{n+1} = k) = \mathbf{E}\left(\frac{U_k(n)}{n+1}\right) = \mathbf{E}\big(\nu_n(k)\big)$$

autrement dit la loi de D_{n+1} est ce qu'on appelle la mesure d'intensité (déterministe) du processus ponctuel ν_n, par conséquent *les résultats en moyenne sur la mesure ν_n donneront des résultats en loi sur D_n.*

Le comportement en moyenne des $U_k(n)$ est connu :

Théorème 11 (Lynch, 1965)**.**

$$\mathbf{E}\, U_k(n) = \frac{2^k}{n!} \begin{bmatrix} n \\ k \end{bmatrix}$$

où $\begin{bmatrix} n \\ k \end{bmatrix}$ sont les nombres de Stirling de première espèce, c'est-à-dire que $\begin{bmatrix} n \\ k \end{bmatrix}$ est le coefficient de x^k dans le développement de $x(x+1)\dots(x+n-1)$. Ces nombres vérifient la relation de récurrence

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + n \begin{bmatrix} n \\ k \end{bmatrix}.$$

Corollaire 5. *La loi de D_n est donnée par*

$$\mathbf{P}(D_{n+1} = k) = \frac{2^k}{(n+1)!} \begin{bmatrix} n \\ k \end{bmatrix}.$$

Le comportement p. s. des $U_k(n)$ peut, lui, se comprendre synthétiquement avec le « polynôme de niveau » qui est défini maintenant.

Définition 6. On appelle *polynôme de niveau* d'un arbre binaire de recherche T_n, le polynôme défini pour tout paramètre $z \in \mathbb{C}$ par

$$W_n(z) := \sum_{k=0}^{+\infty} U_k(n) z^k.$$

Il s'agit bien d'un polynôme, car pour $k > 1 + h_n$, $U_k(n) = 0$. Bien entendu, c'est une variable aléatoire, puisque les $U_k(n)$ sont aléatoires.

Les résultats en moyenne sur le polynôme de niveau se déduisent facilement des résultats en moyenne sur les $U_k(n)$:

$$\mathbf{E}\, W_n(z) = \sum_{k=0}^{+\infty} \mathbf{E}\, U_k(n) z^k$$

et d'après le Théorème 11,

$$\mathbf{E}\, W_n(z) = \frac{1}{n!} \sum_{k=0}^{+\infty} 2^k z^k \begin{bmatrix} n \\ k \end{bmatrix} = \frac{1}{n!} 2z(2z+1)\dots(2z+n-1)\, \mathbf{E}\, W_n(z) = \prod_{j=0}^{n-1} \frac{j+2z}{j+1} \tag{3}$$

La loi de la profondeur est alors immédiate, car si $D_n(z)$ est sa série génératrice

$$D_n(z) = \sum_{k \geq 0} \mathbf{P}(D_n = k) z^k = \mathbf{E}\left(z^{D_n}\right),$$

comme $\mathbf{P}(D_{n+1} = k) = \frac{1}{n+1}\, \mathbf{E}\, U_k(n)$ (*cf.* Équation (2)), on obtient

$$D_{n+1}(z) = \frac{1}{n+1}\, \mathbf{E}\, W_n(z) = \frac{1}{(n+1)!} \prod_{j=0}^{n-1} (j+2z)$$

qui est une expression assez explicite de la série génératrice de la profondeur d'insertion.

Pour obtenir davantage, c'est-à-dire des résultats p. s. sur le polynôme de niveau, nous allons décrire son évolution dans le temps, et c'est là qu'une martingale va apparaître.

4.3. **Résultats p. s. sur le polynôme de niveau : une martingale.** La relation évidente

$$U_k(n+1) - U_k(n) = -1_{\{D_{n+1}=k\}} + 2 \cdot 1_{\{D_{n+1}=k-1\}}$$

permet d'avoir une relation de récurrence sur le « polynôme de niveau » :

$$\mathbf{E}(W_{n+1}(z) \,/\, T_n) = \sum_{k\geq 0} z^k \big(U_k(n) + 2\,\mathbf{P}(D_{n+1} = k-1 \,/\, T_n) - \mathbf{P}(D_{n+1} = k \,/\, T_n)\big) = \frac{n+2z}{n+1} W_n(z),$$

ce qui signifie que le polynôme de niveau convenablement renormalisé est une martingale.

Théorème 12. *Pour tout nombre complexe $z \in \mathbb{C}$,*

$$M_n(z) := \frac{W_n(z)}{\mathbf{E}\big(W_n(z)\big)} = \frac{W_n(z)}{\prod_{j=0}^{n-1} \frac{j+2z}{j+1}}$$

est une $\mathcal{F}_n$-martingale qui :

1. *converge p. s. pour tout z réel positif ;*
2. *converge dans L^1 sur l'intervalle réel $]\, c'/2, c/2\,[$ (les constantes c et c' étant celles du théorème de Devroye sur la hauteur des arbres binaires de recherche). Elle converge vers 0 en dehors de l'intervalle $[\, c'/2, c/2\,]$;*
3. *converge dans L^2 sur la boule $B\left(1, \frac{1}{\sqrt{2}}\right)$ de $\mathbb{C}$.*

Idée de démonstration. La propriété de martingale vient du calcul ci-dessus, par renormalisation.

1. La convergence presque sûre dans $\mathbb{R}_+$ est celle de toute martingale positive.
2. La convergence dans L^1 est plus difficile à voir. Elle est obtenue en bornant $M_n(z)$ dans L^p pour $p > 1$, de façon assez analogue à la méthode utilisée pour un processus de branchement spatial. Les détails sont dans [9].
3. La convergence L^2 pour z dans $\mathbb{C}$ s'obtient en calculant la covariance de la martingale et avec un peu d'analyse complexe. La convergence L^1 (et L^p) est un problème ouvert dans $\mathbb{C}$. Les détails se trouvent dans [3].

□

Commentaire. Globalement, cette étude tire parti de l'égalité

$$W_n(z) = \mathbf{E}\big(W_n(z)\big) M_n(z)$$

où $M_n(z)$ est une martingale. L'égalité ci-dessus est très parlante : en effet, elle permet de séparer l'étude du pôlynome de niveau en deux parties beaucoup plus simples ; une partie déterministe, $\mathbf{E}\big(W_n(z)\big)$, donnée exactement par (3) et dont on connaît l'asymptotique ; et une partie aléatoire qui a la propriété de martingale. Tout l'aléa est concentré dans cette partie martingale et l'étude de la convergence est facilitée.

Par la suite, des résultats de type théorème central limite et grandes déviations sur la mesure ν_n peuvent être obtenus, ainsi que l'ordre de grandeur de la largeur d'un arbre binaire de recherche. Appelons en effet $\overline{Z}_n$ la largeur de T_n ; c'est le maximum sur tous les niveaux du nombre de nœuds à chaque niveau :

$$\overline{Z}_n := \max_{k\geq 0} Z_k(n)$$

Théorème 13 (largeur d'un arbre binaire de recherche)**.**

$$\frac{\overline{Z}_n}{n/\sqrt{\pi \log n}} = 1 + \mathcal{O}\left(\frac{1}{\sqrt{\log n}}\right) \quad p.\ s.$$

lorsque $n \to +\infty$.

Exercice. Soit E_n la longueur de cheminement externe, c'est-à-dire la somme des longueurs de u pour tous les u nœuds externes de l'arbre T_n. Montrer que

$$E_n = W_n'(1)$$

où W_n est le polynôme de niveau. En déduire le théorème de Régnier (1989) :

Théorème 14. $\frac{1}{n+1}\big(E_n - \mathbf{E}(E_n)\big)$ *est une* $\mathcal{F}_n$*-martingale qui converge dans* L^2.

Bibliography

[1] Athreya (Krishna B.) and Ney (Peter E.). – *Branching processes*. – Springer-Verlag, New York, 1972, xi+287p. Die Grundlehren der mathematischen Wissenschaften, Band 196.

[2] Biggins (J. D.). – How fast does a general branching random walk spread? In *Classical and modern branching processes (Minneapolis, MN, 1994)*, pp. 19–39. – Springer, New York, 1997.

[3] Chauvin (Brigitte), Drmota (Michael), and Jabbour-Hattab (Jean). – The profile of binary search trees. *The Annals of Applied Probability*, vol. 11, n° 4, 2001, pp. 1042–1062.

[4] Devroye (L.). – Branching processes in the analysis of the heights of trees. *Acta Informatica*, vol. 24, n° 3, 1987, pp. 277–298.

[5] Drmota (Michael). – The variance of the height of digital search trees. *Acta Informatica*, vol. 38, n° 4, 2002, pp. 261–276.

[6] Duflo (Marie). – *Méthodes récursives aléatoires*. – Masson, Paris, 1990, *Techniques Stochastiques*, xiv+361p.

[7] Grimmett (G. R.) and Stirzaker (D. R.). – *Probability and random processes*. – The Clarendon Press Oxford University Press, New York, 1992, xii+541p. Deuxième édition.

[8] Hall (P.) and Heyde (C. C.). – *Martingale limit theory and its application*. – Academic Press, New York, 1980, xii+308p. Probability and Mathematical Statistics.

[9] Jabbour-Hattab (J.). – *Martingales and large deviations for the binary search trees*. – Rapport n° 29, Prépublications du LAMA, mai 1999. À paraître dans *Random Structures & Algorithms*.

[10] Mahmoud (Hosam M.). – *Evolution of random search trees*. – John Wiley & Sons, New York, 1992, *Wiley-Interscience Series in Discrete Mathematics and Optimization*, xii+324p.

[11] Mahmoud (Hosam M.), Smythe (R. T.), and Szymański (Jerzy). – On the structure of random plane-oriented recursive trees and their branches. *Random Structures & Algorithms*, vol. 4, n° 2, 1993, pp. 151–176.

[12] McDiarmid (Colin). – Concentration. In *Probabilistic methods for algorithmic discrete mathematics*, pp. 195–248. – Springer, Berlin, 1998.

[13] Neveu (Jacques). – *Martingales à temps discret*. – Masson, Paris, 1972, vii+218p.

[14] Petrov (Valentin V.). – *Limit theorems of probability theory*. – The Clarendon Press Oxford University Press, New York, 1995, *Oxford Studies in Probability*, vol. 4, xii+292p. Sequences of independent random variables, Oxford Science Publications.

[15] Spencer (Joel). – Nine lectures on random graphs. In *École d'Été de Probabilités de Saint-Flour XXI—1991*, pp. 293–347. – Springer, Berlin, 1993.

[16] Williams (David). – *Probability with martingales*. – Cambridge University Press, Cambridge, 1991, *Cambridge Mathematical Textbooks*, xvi+251p.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 167–172.

Available online at the URL
http://algo.inria.fr/seminars/.

Phase Transitions and Satisfiability Threshold[†]

Olivier Dubois[(a)] *and Vincent Puyhaubert*[(b)]

[(a)]Laboratoire LIP6, UPMC (France) and [(b)]Projet Algo, Inria Rocquencourt (France)

[(a)]March 20, 2002 and [(b)]January 20, 2003

Summary by Vincent Puyhaubert

Abstract

The 3-SAT problem consists in determining if a boolean formula with 3 literals per clause is satisfiable. When the ratio between the number of clauses and the number of variables increases, a threshold phenomenon is observed: the probability of satisfiability appears to decrease sharply from 1 to 0 in the neighbourghood of a fixed threshold value, conjectured to be close to 4.25. Although a threshold value has been provably obtained for the similar problem 2-SAT and for closely related problems like 3-XORSAT, there is still no proof for the 3-SAT problem.

Recent works have so far provided only upper and lower bounds for the potential location of the threshold. We present here a survey of methods giving upper bounds. We also introduce generating functions as a new generic tool and rederive some of the most significant upper bounds in a simple uniform manner.

1. Introduction

We consider boolean formulæ over a set of variable $x_1, \ldots, x_n$ (where the x_j range over $\{0,1\}$ or $\{\text{true}, \text{false}\}$). A literal is either a variable x_j or a negated variable $\neg x_j$. It is known that each boolean formula admits a conjunctive normal form, being a conjunction of clauses, themselves disjunctions of literals. A 3-SAT formula is then such a formula with exactly 3 literals per clause. A typical formula is then for example:

$$\Phi = (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_5) \wedge (x_1 \vee \neg x_4 \vee \neg x_5) \wedge (x_3 \vee \neg x_4 \vee \neg x_5).$$

We will choose the model where each clause is composed of a set of three literals from distinct variables. There are then $8\binom{n}{3}$ distinct clauses and $8^m\binom{n}{3}^m$ formulæ with m clauses. Other models may be occasionally used for convenience in calculations, for example, the three literals may be ordered and not necessarily distinct so that there would be $8n^3$ clauses. All these models are easily proved to be equivalent with respect to the probability of satisfiability.

In Figure 1, a *phase transition phenomenon* can be observed regarding the satisfiability of these formulæ when they are drawn at random. As the ratio r of the number m of clauses to the number n of variables increases, the probability of satisfiability drops abruptly from nearly 1 to nearly 0.

From these experiments, it is believed that there exists a critical value r_3 such that for any $\epsilon > 0$, the probability of satisfiability tends to 1 for $r < r_3 - \epsilon$ (as m and n tend to infinity), and tends to 0 for $r > r_3 + \epsilon$. Experiments suggest for r_3 the value 4.25 ± 0.05. However, so far, only successive

[†]This text summarizes both the course given by Olivier Dubois at the ALEA'02 meeting in Luminy (France) and a seminar talk by Vincent Puyhaubert at the Algorithms seminar.

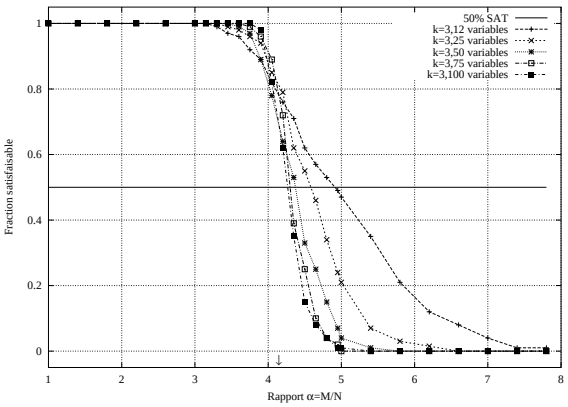


FIGURE 1. Ratio of satisfiable formulæ with respect to the parameter m/n.

upper and lower bounds of the potential location of the threshold have been obtained. The table below lists the bounds successively established for the 3-SAT threshold. The bounds marked with a star admit an extension to k-SAT for any k.

Lower bounds for 3-SAT threshold		Upper bounds for 3-SAT threshold	
2.9*	Chao and Franco (1986,1990) [4]	5.191*	Franco and Paull (1983) [8]
2/3*	Chvátal and Reed (1992)	5.081	El Mafthoui and Fernandez de la Vega (1993) [6]
1.63	Broder et al. (1993) [3]	4.762*	Kamath et al. (1995) [12]
3.003*	Frieze and Suen (1996) [10]	4.643*	Dubois and Boufkhad (1997) [5]
3.145	Achlioptas (2000) [1]	4.602	Kirousis et al. (1998) [13]
		4.596	Janson et al. (1999) [11]
		4.506	Dubois et al.

Apart from these works, Friedgut [9] also proved that there exists a sequence (γ_n) such that for any $\epsilon > 0$, the probability of satisfiability tends to 1 as m and n increase under the constraint $m/n < \gamma_n - \epsilon$, while it tends to 0 under the constraint $m/n > \gamma_n + \epsilon$. But it is not known whether the sequence (γ_n) converges. The limiting value γ would be the threshold r_3.

The aim of the present paper is to present some of the most significant upper bounds on the satisfiability threshold. We will specially focus on enumerative proofs, with the help of generating functions. For lower bounds, one can refer to the surveys by Franco [7] and Achlioptas [2].

2. Expectations of the Number of Solutions

The first bound for 3-SAT threshold has been obtained by several authors as a direct application of the first-moment method to the random variable giving the number of solutions of a random formula. Under an enumerative perspective, it can be seen as a direct application of the following simple remark: *Each positive integer k satisfies $k \geq 1$.* From there, one has the following inequality:

$$\big|\Phi \text{ satisfiable}\big| \leq \big|(\Phi, S) \text{ such that } \Phi \text{ is satisfied by } S\big|. \tag{1}$$

Let S be an assignment of the n variables to values in $\{0,1\}$ and $C = \pm x_i \vee \pm x_j \vee \pm x_k$ a clause. There is only one way to choose the signs of the three literals in order to have the value of C be false under S: each literal must have the opposite sign of its assignment. Then, there are 7 ways to choose the signs in order to render C true. The number of clauses satisfied by any given S is

then $7\binom{n}{3}$. Since S is a solution of a 3-SAT formula Φ if and only if all clauses of Φ are satisfied by S, for any assignment, there are exactly $7^m\binom{n}{3}^m$ formulas with m clauses which admit S as a solution.

The cardinality of the pairs (Φ, S) such that S is a solution of Φ is then given by $2^n 7^m \binom{n}{3}^m$. Dividing each term of (1) by the total number of formulæ $8^m\binom{n}{3}^m$ gives (with $r = m/n$):

$$\mathbf{P}(\Phi \text{ satisfiable}) \leq \left(2\left(\frac{7}{8}\right)^r\right)^n. \tag{2}$$

Hence, for $r > \ln(2)/\ln(8/7) \approx 5.191$, the right-hand side of (2) tends to 0 as n tends to infinity, and so does the probability of satisfiability. This gives the first upper bound obtained by Franco and Paull.

3. **Prime Implicants**

In the previous section, we have bounded the number of satisfiable formulæ by their number of solutions. Since a formula may have from 1 to almost 2^n solutions, the upper bound provided may be very coarse. The next idea is to group some of the solutions which look very close to each other and enumerate only these groups for each formula. In this way, it may be possible to get an improved upper bound on the satisfiability threshold.

This leads to the definition of partial assignments and prime implicants. A partial assignment A is simply an assignment of a subset of the n variables (possibly all, so that solutions are also partial assignments). Let us say that A satisfies a formula Φ if and only if all complete assignments A' extending A are solutions of Φ. A necessary and sufficient condition for this is that in each clause of Φ, there exists at least one of the three literals which is true under A. If there are k missing variables in a partial assignment A, then A "groups" 2^k solutions together.

A natural order may be placed on partial assignments. We say that A is smaller than B if we can remove some assigned variables from B to get A. A prime implicant is then a partial assignment which satisfies Φ and is *minimal* with respect to this order. Any satisfiable formula has then at least one prime implicant since it has at least one solution and the set of partial assignments is then non-empty. As in the previous section, we get from there the inequality (see (1)):

$$|\Phi \text{ satisfiable}| \leq |(\Phi, I) \text{ such that } I \text{ is a prime implicant of } \Phi|. \tag{3}$$

Note that the sets of solutions grouped together by two distinct partial assignments are not necessarily disjoint. Some formulæ may have more prime implicants than solutions. But in fact, the expectation of the number of prime implicants of a random formula appears to be smaller by an exponential factor.

Let I be a partial assignment of k variables. Then, all clauses in a formula Φ that admits I as a prime implicant must contain at least one literal satisfied by I. Let $A_{n,k}$ be the set of such clauses and $\alpha_{n,k}$ their number (it is clear that this quantity depends only on k and n and does not depend on the names or values of the variables assigned in I).

Let then Φ be a formula which admits I as a prime implicant. Recall that I has to be minimal with respect to the order defined earlier. Let I' be obtained from I by removing a variable x_i from the set of assigned variables. Then I' can not satisfy Φ, which means that at least one clause in Φ must be rendered false by I'.

Hence, at least one clause is of the form $\pm x_i \vee a \vee b$ where the sign of the literal x_i makes this literal positive under I and where a and b are literals from unassigned variables or false under I. Let C_{x_i} be the set of such clauses. Then all these sets have the same number of elements $\beta_{n,k}$ and

are mutually disjoint. In order to build a formula for which I is a prime implicant, we need to choose m clauses among $A_{n,k}$ so that in k subsets, we must pick at least one element. The number of such formulæ is then the number $I_{n,k,m}$ whose generating function is given by

$$I_{n,k}(z) = \sum_{m\geq 0} I_{n,k,m}\frac{z^m}{m!} = e^{z(\alpha_{n,k}-k\beta_{n,k})}\left(e^{z\beta_{n,k}}-1\right)^k.$$

Finally, since there are $\binom{n}{k}2^k$ partial assignments of k variables, the total number of pairs (Φ, I) such that I is a prime implicant for Φ is given by:

$$\left|(\Phi, I)\right| = \sum_{k=0}^{n}\binom{n}{k}2^k m!\,[z^m]\,e^{z(\alpha_{n,k}-k\beta_{n,k})}\left(e^{z\beta_{n,k}}-1\right)^k. \tag{4}$$

The next step depends on the following general remark: if (f_k) is a sequence of positive reals and $f(z) = \sum f_k z^k$ then for all $s > 0$ within the domain of convergence of $f(z)$:

$$f_k \leq \frac{f_0}{s^k} + \cdots + f_k + f_{k+1}s + \cdots = \frac{f(s)}{s^k} \quad \text{and thus} \quad f_k \leq \min_s \frac{f(s)}{s^k}.$$

From now on, we set $k = \alpha n$, $m = rn$ and make use of the upper bounds $\alpha_{n,k} - k\beta_{n,k} \leq \frac{1}{3}k^2(3n-k)$ and $\beta_{n,k} \leq \frac{1}{2}(2n-k)^2$. From (3) and (4) one determines:

$$\mathbf{P}(\Phi \text{ satisfiable}) \leq \sum_{\alpha\in\{0,1/n,\ldots,1\}} f(\alpha)^n \tag{5}$$

with

$$f(\alpha) = \left(\tfrac{3r}{4e}\right)^r \tfrac{2^\alpha}{\alpha^\alpha(1-\alpha)^{1-\alpha}} e^{\frac{u_\alpha}{3}\alpha^2(3-\alpha)}\left(e^{\frac{u_\alpha}{2}(2-\alpha)^2}-1\right)^\alpha {u_\alpha}^{-r}$$

where u_α makes $f(\alpha)$ minimal. For $r > 4.89$, one verifies that the maximum of f is strictly under 1. The probability of satisfiability in then bounded from above by $(n+1)\delta^n$ with $\delta < 1$ and thus, tends to 0 as n tends to infinity. The idea of prime implicant was first introduced by Olivier Dubois and an improvement of this idea led to the value 4.762 obtained by Kamath.

4. **Negatively Prime Solutions**

The next idea is to introduce a partial order on the set of solutions. Define B to be an assignment smaller than A if we can change the values of some of its variables from 0 to 1 to get A. We now propose to enumerate only pairs (Φ, S) where S is a maximal solution with respect to this order. In fact, it is very difficult to find for any given assignment a simple characterization of formulæ for which it is a maximal solution; consequently we have to deal with a weaker definition of *local maximal solution* (also called *negatively prime solution* or NPS). This is a solution for which changing the value of any variable from 0 to 1 no longer gives a solution of our formula. This amounts to considering solutions which do not admit a greater solution that differs in exactly one variable. Once more, we start from the inequality:

$$|\Phi \text{ satisfiable}| \leq |(\Phi, S) \text{ such that } S \text{ is a NPS of } \Phi|. \tag{6}$$

Let A be an assignment giving the value 0 to k variables and Φ a formula for which A is an NPS. Then, all clauses of Φ must belong to the set A_n of all $7\binom{n}{3}$ clauses satisfied by A (as seen in Section 2). Now, if any variable x_i assigned to 0 is changed to 1, there must be at least one clause in Φ that is no longer satisfied by this new assignment: at least one clause must be of the form $\neg x_i \vee a \vee b$ where a and b are false under A. If we denote by C_{x_i} this set of clauses (for each variable assigned to 0), then all these sets have the same number $\binom{n-1}{2}$ of elements and are

mutually disjoint. As in the previous section, since there are $\binom{n}{k}$ solutions with k variables assigned to 0, we get:

$$\left|(\Phi, A \text{ NPS})\right| = \sum_{k=0}^{n} \binom{n}{k} m! \, [z^m] \, e^{z\left(7\binom{n}{3} - k\binom{n-1}{2}\right)} \left(e^{z\binom{n-1}{2}} - 1\right)^k.$$

By $[z^m]\, f(z) + g(z) = [z^m]\, f(z) + [z^m]\, g(z)$, this gives a closed-form expression:

$$\left|(\Phi, A \text{ NPS})\right| = m! \, [z^m] \, e^{z4\binom{n}{3}} \left(2e^{z\binom{n-1}{2}} - 1\right)^n. \tag{7}$$

The same remark as in the previous section, Stirling formula, and the change of variable $z = \delta\binom{n-1}{2}$ provide that for any $\delta > 0$ with $m = rn$:

$$\mathbf{P}(\Phi \text{ satisfiable}) \leq \left(\left(\frac{3r}{8e}\right)^r \frac{e^{\frac{4}{3}\delta}\left(2e^{\delta} - 1\right)}{\delta^r} \right)^n. \tag{8}$$

This expression is minimized by $\delta\left(\frac{4}{3} + \frac{2e^{\delta}}{2e^{\delta}-1}\right) = r$ and, with such a δ, is strictly smaller than 1 as soon as $r > 4.643$. Hence, the probability of satisfiability tends to 0 for every r greater than this value. This bound was first obtained by Dubois and can be extended to k-SAT for any k. It is so far the best *general* upper bound known for k-SAT.

5. **Typical Formulæ**

In the previous section, we have enumerated all pairs of formulæ and NPS. However, there may be a negligible proportion of formulæ with a huge number of such solutions. In this case, when we enumerate the NPS for these formulæ, the contribution to the whole sum may be non negligible. The idea here is to throw away some formulæ and then, enumerate the NPS only for the retained formulæ, which are called *typical* formulæ. The whole calculation will not be given here, only the idea that led to the proof.

In this section, we introduce for convenience a variation of the model used so far (this does not affect the threshold value). A formula consists in a sequence of $3m$ literals among the $2n$ possible ones, where 3 consecutive literals form a clause (thus literals within clauses are allowed to repeat). Let $\omega_{p,l}$ be the random variable giving the fraction of variables which appear in the formula p times where l of the occurrences are positive. Then, when $m = rn$, the variable quantity $\omega_{p,l}$ follows a Poisson limit law in the following sense: let $\kappa_{p,l} = \frac{1}{2^l}\binom{p}{l}\frac{\lambda^k}{k!}e^{-\lambda}$ with $\lambda = 3r$, then

$$\forall l, p \quad \forall \epsilon > 0 \qquad \mathbf{P}\left(|\omega_{l,p} - \kappa_{l,p}| > \epsilon\right) \underset{n\to\infty}{\longrightarrow} 0. \tag{9}$$

Let $x_{\max}$ be an integer and $\epsilon > 0$. A formula will be called typical if and only if

$$\forall 0 \leq p \leq l \leq x_{\max} \qquad \left|\omega_{l,p}(\Phi) - \kappa_{l,p}\right| \leq \epsilon.$$

For any fixed $x_{\max}$ and ϵ, as a consequence of (9), the set of non typical formulæ is negligible. Hence:

$$\mathbf{P}(\Phi \text{ satisfiable}) \leq \frac{\left|(\Phi \text{ typical}, S \text{ NPS})\right|}{|\Phi|} + o(1). \tag{10}$$

With $x_{max} = 56$ and $\epsilon = 10^{-15}$, for $r = 4.506$, the expectation of the number of NPS among typical formulæ was proven to be $o(1)$. This value, obtained by Dubois, is the best currently known upper bound for the 3-SAT threshold.

Remark. In fact, one last refinement is needed in order to achieve the upper bound 4.506. In a formula, if one switches all variables appearing more often under positive form than under negative form, in the sense that all positive occurrences (resp. all negative) are replaced by the negated literal (resp. the positive), the satisfiability of the formula remains unchanged, as does the number of solutions. However, the number of NPS is lowered. The last idea in the proof, is to enumerate, for typical formulæ, not their own number of NPS but the one of their so called *totally unbalanced* form.

Bibliography

[1] Achlioptas (D.). – Setting two variables at a time yields a new lower bound for random 3-sat. In *Proceeding of the 32nd ACM Symposium on Theory of Computing, Association for Computing Machinery*, pp. 28–37. – 2000.

[2] Achlioptas (Dimitris). – Lower bounds for random 3-SAT via differential equations. *Theoret. Comput. Sci.*, vol. 265, n° 1-2, 2001, pp. 159–185. – Phase transitions in combinatorial problems (Trieste, 1999).

[3] Broder (Andrei Z.), Frieze (Alan M.), and Upfal (Eli). – On the satisfiability and maximum satisfiability of random 3-CNF formulas. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (Austin, TX, 1993)*. pp. 322–330. – ACM, New York, 1993.

[4] Chao (Ming-Te) and Franco (John). – Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem. *Information Sciences*, vol. 51, n° 3, 1990, pp. 289–314.

[5] Dubois (O.) and Boufkhad (Y.). – A general upper bound for the satisfiability threshold of random r-SAT formulae. *Journal of Algorithms*, vol. 24, n° 2, 1997, pp. 395–420.

[6] El Maftouhi (A.) and Fernandez de la Vega (W.). – On random 3-sat. *Combinatorics, Probability and Computing*, vol. 4, n° 3, 1995, pp. 189–195.

[7] Franco (John). – Results related to threshold phenomena research in satisfiability: lower bounds. *Theoretical Computer Science*, vol. 265, n° 1-2, 2001, pp. 147–157. – Phase transitions in combinatorial problems (Trieste, 1999).

[8] Franco (John) and Paull (Marvin). – Probabilistic analysis of the Davis-Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics*, vol. 5, n° 1, 1983, pp. 77–87.

[9] Friedgut (Ehud). – Sharp thresholds of graph properties, and the k-sat problem. *J. Amer. Math. Soc.*, vol. 12, n° 4, 1999, pp. 1017–1054. – With an appendix by Jean Bourgain.

[10] Frieze (Alan) and Suen (Stephen). – Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, vol. 20, n° 2, 1996, pp. 312–355.

[11] Janson (Svante), Stamatiou (Yannis C.), and Vamvakari (Malvina). – Erratum to: "Bounding the unsatisfiability threshold of random 3-SAT" [Random Structures Algorithms **17** (2000), no. 2, 103–116; MR 2001c:68065]. *Random Structures and Algorithms*, vol. 18, n° 1, 2001, pp. 99–100.

[12] Kamath (Anil), Motwani (Rajeev), Palem (Krishna), and Spirakis (Paul). – Tail bounds for occupancy and the satisfiability threshold conjecture. *Random Structures & Algorithms*, vol. 7, n° 1, 1995, pp. 59–80.

[13] Kirousis (Lefteris M.), Kranakis (Evangelos), Krizanc (Danny), and Stamatiou (Yannis C.). – Approximating the unsatisfiability threshold of random formulas. *Random Structures & Algorithms*, vol. 12, n° 3, 1998, pp. 253–269.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 173–176.

Available online at the URL
http://algo.inria.fr/seminars/.

Génération aléatoire[†]

Alain Denise
LRI, Orsay (France)

March 19, 2002

Summary by Sylvie Corteel

Abstract

Le but de ce résumé est de présenter brièvement les techniques de génération aléatoire. Nous nous concentrerons sur deux aspects : l'approche récursive et les chaines de Markov. Pour une vue plus générale et détaillée, nous conseillons la lecture du rapport d'habilitation d'Alain Denise [1] dont est inspiré ce résumé.

1. L'approche récursive

1.1. **Premiers formalismes.** En 1977, Wilf considère une famille de structures combinatoires dont la construction peut être représentée par un chemin dans un graphe orienté acyclique. L'exemple des sous-ensembles de cardinal k d'un ensemble $E = \{e_1, e_2, \ldots, e_n\}$ illustre le principe. Les étapes successives de la construction d'un tel sous-ensemble peuvent être représentées par un chemin dans le plan discret de longueur n et de hauteur k qui n'emprunte que des pas $s_1 = (+1, +1)$ et $s_0 = (+1, 0)$. Lorsqu'on est au point (i, j), le choix du pas s_1 détermine l'appartenance de l'élément e_i au sous-ensemble ; le choix de s_0 détermine sa non-appartenance. Notons $C_{n,k}$ l'ensemble de ces chemins et convenons d'appeler respectivement longueur et hauteur les entiers n et k. Tout chemin de $C_{i,j}$ est soit un pas s_0 suivi d'un chemin de $C_{i-1,j}$, soit un pas s_1 suivi d'un chemin de $C_{i-1,j-1}$. À chaque étape, on va donc choisir d'engendrer un pas s_0 avec probabilité $|C_{i-1,j}|/|C_{i,j}| = (i-j)/i$ et un pas s_1 avec probabilité $|C_{i-1,j-1}|/|C_{i,j}| = j/i$.

1.2. **Spécifications pour les structures décomposables.** En 1994, Flajolet, Zimmermann et Van Cutsem publient un schéma général de composition de structures combinatoires qui repose sur la notion de *spécification combinatoire*. Dans ce formalisme, les objets primitifs sont l'objet « vide » (de taille 0) noté 1, et un ensemble fini d'*atomes* de taille 1. Le symbole Z désigne un atome générique. Cinq opérateurs permettent de définir récursivement des ensembles d'objets combinatoires à partir d'autres ensembles et des deux types d'objets primitifs :

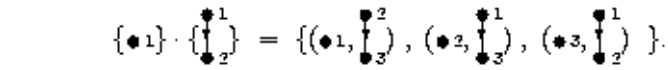
- L'union disjointe : $A \cdot B = \{\, i \mid i \in A \text{ ou } i \in B \,\}$.
- Le produit (non commutatif) : $A \cdot B = \{\, (a, b) \mid a \in A \text{ et } b \in B \,\}$.
- La séquence, l'ensemble et le cycle : $\texttt{sequence}(A)$ (resp. $\texttt{set}(A)$, $\texttt{cycle}(A)$) désigne l'ensemble des suites (resp. des ensembles, des cycles) finies d'éléments de A. Ces opérateurs peuvent être accompagnés d'un argument qui fixe une condition sur la cardinalité des suites (resp. ensembles, cycles).

[†]Notes de cours pour le cours donné pendant le groupe de travail ALÉA'02 au Cirm à Luminy (France).

Structures étiquetées	
Permutations	$P = \texttt{sequence}(Z)$ ou $P = \texttt{set}(\texttt{cycle}(Z))$
Partitions d'ensembles	$P = \texttt{set}(\texttt{set}(Z,\ \text{card} \geq 1))$
Surjections	$S = \texttt{sequence}(\texttt{set}(Z,\ \text{card} \geq 1))$
Structures non étiquetées	
Partitions d'entiers	$F = \texttt{set}(\texttt{sequence}(Z,\ \text{card} \geq 1))$
Compositions d'entiers	$F = \texttt{set}(\texttt{set}(Z,\ \text{card} \geq 1))$
Chemins de Dyck	$D = 1 \ + \ Z \cdot D \cdot \bar{Z} \cdot D$

FIGURE 1. Quelques spécifications combinatoires.

Dans l'article initial, les objets sont *étiquetés* : à chaque atome d'un objet est associé un entier. Deux atomes différents ont toujours des étiquettes différentes. Cela implique que, lors des quatre dernières opérations ci-dessus, un réétiquetage des objets est effectué. Ainsi, par exemple, le produit n'est pas un produit cartésien, comme illustré ci-après :



Dans un travail ultérieur, les mêmes auteurs adaptent leur formalisme aux objets non étiquetés. Dans ce cas, les opérateurs restent le même et le produit est un produit cartésien. L'opérateur `set` pour les objets non étiquetés ne construit pas des ensembles, mais des multi-ensembles.

Soit $T = \{T_0, T_1, \ldots, T_m\}$ une famille de $m+1$ ensembles d'objets combinatoires (étiquetés ou non). Une *spécification combinatoire* de T est un ensemble de $m+1$ équations telles que la $i$ème (pour tout $0 \leq i \leq m$) s'écrit $T_i = \Psi_i(T_0, T_1, \ldots, T_m)$, où Ψ_i est une combinaison des cinq opérateurs définis ci-dessus appliquée aux T_i, à l'objet vide et aux atomes.

La Figure 1 présente quelques exemples de spécifications pour des objets combinatoires classiques.

Pour aboutir à des algorithmes de génération aléatoire efficace, il est nécessaire de transformer la spécification en *spécification standard*.

Soit $T = \{T_0, T_1, \ldots, T_m\}$ une famille de $m+1$ ensembles d'objets combinatoires étiquetés. Une *spécification standard* de T est un ensemble de $m+1$ équations telles que la $i$ème (pour tout $0 \leq i \leq m$) s'écrit $T_i = 1$ ou $T_i = Z$ ou $T_i = U_j + U_k$ ou $T_i = U_j \cdot U_k$ ou $\Theta T_i = U_j \cdot U_k$. Chaque U_j appartient à $\{1, Z, T_0, \ldots, T_m, \Theta T_0, \ldots, \Theta T_m\}$. Le symbole Θ désigne l'opérateur de *pointage* :

$$\Theta A \ = \ \bigcup_{n=1}^{\infty} (A_n \times \{1, 2, \ldots, n\})$$

où A_n est l'ensemble des objets de A de taille n.

1.3. **Algorithmes de génération.** L'algorithme de génération aléatoire se déduit de la spécification standard. Soit n la taille des objets à engendrer. La première étape est une étape de dénombrement : il s'agit de calculer, pour tout ensemble C intervenant dans la spécification standard et pour tout $0 \leq i \leq n$, le nombre c_i d'objets de taille i de C. On utilise les fonctions génératrices exponentielles (resp. ordinaires) pour les objets étiquetés (resp. non étiquetés). On déduit directement de la spécification standard les relations de récurrence pour les séries. Par exemple :

$$C = 1 \ \Rightarrow \ C(z) = 1\,; \qquad C = Z \ \Rightarrow \ C(z) = z\,;$$
$$C = A + B \ \Rightarrow \ C(z) = A(z) + B(z); \qquad C = A \cdot B \ \Rightarrow \ C(z) = A(z)B(z)\,.$$

	Dénombrement	Génération	Mémoire
Structures décomposables			
Cas général	$O(n(\log n)^2 \operatorname{loglog} n)$	$O(n \log n)$	$O(n)$
Cas holonome	$O(n)$	$O(n \log n)$	$O(1)$
Cas itératif	$O(n(\log n)^2 \operatorname{loglog} n)$	$O(n)$	$O(n)$
Langages algébriques	$O(n)$	$O(n \log n)$	$O(1)$
Langages rationnels	$O(n)$	$O(n)$	$O(1)$

TABLE 1. Quelques complexités.

Le nombre d'opérations arithmétiques effectuées au cours de l'étape de dénombrement est clairement en $O(n^2)$. Certaines classes d'objets possèdent une série génératrice *holonome* : il existe (et on sait calculer) une formule de récurrence linéaire à coefficients polynomiaux qui donne les termes de la série génératrice. Ceci implique que les termes jusqu'à l'ordre n peuvent être calculés en $O(n)$ opérations arithmétiques. Dans le cas général comme dans le cas holonome, $O(n)$ entiers doivent être stockés.

La phase de dénombrement n'est effectuée qu'une fois. Le processus de génération s'effectue de façon récursive. À chaque ensemble C d'objets représenté dans la spécification standard, on associe une procédure. Par exemple :

```
Cas : C = 1.
   gC := procedure(n: integer)
      if n = 0 then return(1)
   end

Cas : C = A + B.
   gC := procedure(n: integer)
      U:=Uniform([0,1]);
      if U < a_n/c_n
         then return(gA(n))
         else return(gB(n))
   end

Cas : C = Z.
   gC := procedure(n: integer)
      if n = 1 then return(Z)
   end

Cas : C = A · B.
   gC := procedure(n: integer)
      U:=Uniform([0,1]);
      k := 0; S := a_0 b_n/c_n;
      while U > S do
         k := k + 1; S := S + a_k b_{n-k}/c_n;
      return([gA(k),gB(n − k)])
   end
```

Décrivons des moyens d'améliorer la complexité. Pour la complexité en temps, dans le cas $C = A \cdot B$, soit K la variable aléatoire qui représente la taille de l'élément de A, et $\pi_{n,k} = \mathbf{P}(K = k) = a_k b_{n-k}/c_n$. L'algorithme présenté ci-dessus ajoute successivement ces probabilités à S dans l'ordre $\pi_{n,0}, \pi_{n,1}, \pi_{n,2}, \ldots$ Considérons maintenant un algorithme qui effectue le même traitement, mais dans l'ordre suivant : $\pi_{n,0}$, $\pi_{n,n}$, $\pi_{n,1}$, $\pi_{n,n-1}$, ... Cette variante est appelée « boustrophédon ». Cette seule modification donne une complexité arithmétique en $O(n \log n)$. Le principe de la preuve est extrêmement simple : soit $f(n)$ la complexité au pire. Elle satisfait une récurrence de type $f(n) = \max_{0 \le k \le n}\big(f(k) + f(n-k) + 2\min(k, n-k)\big)$ dont la solution est en $O(n \log n)$.

Les *structures itératives* sont celles dont le graphe de dépendance (il existe un arc de l'ensemble A vers l'ensemble B s'ils sont respectivement dans le membre gauche et le membre droit de la même règle) de la spécification combinatoire est acyclique. Dans ce cas, la phase de génération est linéaire. Pour le cas holonome, Goldwurm a prouvé en 1995 que la génération d'un mot de longueur n peut s'effectuer en espace arithmétique $O(1)$, tout en conservant les complexités arithmétiques en $O(n)$ et $O(n \log n)$ respectivement pour les phases de dénombrement et de génération. Dans le cas général,

Van der Hoeven a proposé en 1999 une méthode pour calculer les coefficients jusqu'à l'ordre n de toute série génératrice de structures décomposables en temps $0(M(n)\log n)$, où $M(n)$ désigne la complexité arithmétique de multiplication de deux polynômes de degré $n-1$. Le meilleur algorithme de multiplication présente une complexité arithmétique $M(n) = O(n \log n \log\log n)$. Les résultats sont regroupés en Table 1. Il existe de nombreuses autres techniques basées sur la récursivité : méthodes pour les langages algébriques, méthode paresseuse, méthodes à rejet, génération non uniforme controlée.

2. Chaînes de Markov

2.1. **Génération presque uniforme.** Si on peut déterminer en temps polynomial le nombre d'objets de taille n, au moins asymptotiquement, Jerrum, Valiant et Vazirani ont montré qu'il est toujours possible de concevoir un générateur aléatoire uniforme de complexité polynomiale. Il existe un grand nombre de structures combinatoires que l'on ne sait pas compter aussi facilement. Pour certaines, le problème de leur dénombrement est $\#P$-complet.

Pour ces cas difficiles, la génération aléatoire *presque* uniforme peut être envisagée. De plus, sous certaines conditions, un algorithme de génération presque uniforme peut mener à un algorithme probabiliste polynomial de dénombrement approximatif. Pour un ensemble E d'objets combinatoires, il s'agit de faire en sorte que la différence relative entre la probabilité $p(e)$ d'engendrer un objet $e \in E$ et la probabilité uniforme $p_u = 1/|E|$ soit inférieure à un réel ε fixé.

On considère une chaîne de Markov dont les éléments de E sont les états, et dont chaque transition est déterminée par une modification d'un objet. Si la chaîne est irréductible et apériodique et si la probabilité de transition de e à e' est égale à celle de la transition inverse pour tout e et tout e' de E, alors la loi du processus tend vers une unique distribution stationnaire uniforme. Donc, l'algorithme de génération consiste à partir d'un état quelconque et suivre les transitions avec les probabilités correspondantes. Le problème est de déterminer le temps suffisant pour que la génération soit uniforme à ε prés. S'il est polynomial en la taille n des objets à engendrer et en $\log(1/\varepsilon)$, on dit que la chaîne *se mélange rapidement*. Mais prouver ceci est loin d'être facile. Le premier résultat positif est dû à Jerrum et Sinclair, qui ont présenté en 1989 un algorithme polynomial pour la génération presque uniforme de couplages parfaits dans un graphe.

2.2. **Génération exactement uniforme.** Dans certains cas, l'approche par chaîne de Markov aboutit à une distribution exactement uniforme ; par exemple pour la génération d'arbres couvrants d'un graphe. Un arbre couvrant d'un graphe est sous-graphe connexe sans cycle qui contient tous les sommets du graphe. On sait compter les arbres couvrants d'un graphe, et plusieurs algorithmes de génération existent. Toutefois, l'algorithme présenté ici est à la fois extrêmement simple et plus performant que les précédents. Il a été découvert indépendamment par Aldous et Broder puis amélioré par Wilson. Le procédé consiste à construire l'arbre arête par arête comme suit : partir d'un sommet quelconque du graphe ; à chaque étape, choisir uniformément une arête adjacente au sommet courant et la traverser ; si elle n'appartient pas déjà à l'arbre et si elle n'y occasionne pas de circuit, alors l'ajouter à l'arbre ; stopper dès que l'arbre couvre tous les sommets. Le processus peut être vu comme une chaîne de Markov dont les états sont les sous-arbres du graphe enracinés au sommet courant. On montre qu'on obtient un arbre couvrant avec probabilité uniforme en complexité moyenne en $O(n \log n)$ pour presque tous les graphes, et $O(n^3)$ pour les pires d'entre eux.

Bibliography

[1] Denise (Alain). – *Structures aléatoires : Modèles et analyse des génomes.* – Habilitation à diriger des recherches, LRI, Université Paris-Sud, dec 2001. 76 pages.

Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 177–182.

Available online at the URL
http://algo.inria.fr/seminars/.

Combinatorics and Random Generation

Dominique Gouyou-Beauchamps
LRI, Université Paris-Sud (France)

March 18, 2002

Summary by Nicolas Bonichon

Abstract

We present an overview of different techniques to randomly and uniformly generate combinatorial objects.

1. Motivations and Hypotheses

One of the goals of a combinatorist is to recognize, to enumerate, and to generate objects of different combinatorial classes. Here we present several methods to randomly and uniformly generate objects of a given class. This has applications in simulation in general: image syntheses, statistical physics, genomic, program testing, algorithms analyses, etc. In general, we want to generate an object of size n such that the probability that an object appears is the same for all objects of size n.

There are several ways to measure the complexity of a generating algorithm. The first one is to count the number of calls to the *RANDOM*() function. This function returns a floating-point number between 0 and 1. Another way to measure the complexity is to count the number of arithmetic operations on floating-point numbers or on integers (a call of the *RANDOM*() function is considered as an arithmetic operation). This measure is called the *arithmetic complexity*. Since the generated objects can be huge (up to 10^7 or 10^8) and the manipulated numbers are as large as a^n or $n!$ (hence coded with $O(n)$ or $O(n \log n)$ bits), it also makes sense to count the number of operations on single bits. This is the *bit complexity*. In order to compute some objects of large size, the *time complexity* of efficient algorithms is usually $O(n)$ or $O(n \log n)$.

2. The Predecessors

Nijenhuis and Wilf [7] were the first ones to propose two types of generation algorithms:

- *NEXT*: with a total order on objects of size n and a given object of the family, compute the next one in the order. Generally, these algorithms have a constant average time complexity;
- *RANDOM*: we select randomly and uniformly objects of size n of the family.

Here are two examples of these types of algorithms. Here and throughout the remainder of the text, $[n]$ denotes the set $\{1, \ldots, n\}$.

Example (Permutations of $[n]$, algorithms *NEXTPER* and *RANPER* [7]). *NEXTPER* uses the notion of *sub-exceeding* function: a function f from $[n]$ to $[n]$ is sub-exceeding is and only if for each $i \in [n]$, $1 \leq f(i) \leq i$. It is obvious that the number of sub-exceeding functions from $[n]$ to $[n]$ is $n!$ (one possibility for $f(1)$, two for $f(2)$, and so on).

A clever order on sub-exceeding functions allows us to transform a permutation into the next one with few operations. The average cost of a transformation is $O(1)$ steps.

Input: $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, a permutation of S_n and its signature s (i.e., the number of couples (i, j) such that $\sigma_i > \sigma_j$ and $i < j$).
Output: next permutation.

```
if s = 1 then
  s ← −1; switch σ1 and σ2 and exit
else
  s ← 1; i ← 0; t ← 0
  loop
    d ← 0; i ← i + 1
    for j from 1 to i do
      if σj > σi+1 then d ← d + 1 end if
    end for
    t ← t + d
    if (t is odd) and (d < i) then
      find in σ = (σ1, σ1, ..., σi) the largest number less than σi+1;
      switch this number with σi+1 and exit
    end if
    if (t is even) and (d > 0) then
      find in σ = (σ1, σ1, ..., σi) the smallest number greater than σi+1;
      switch this number with σi+1 and exit
    end if
  end loop
end if
```

FIGURE 1. Algorithm *NEXTPER*.

To compute a random permutation of $[n]$, the algorithm is quite simple (see Algorithm 2). This algorithm is *incremental*. This means that after m steps, for each $m \leq n$, the algorithm generates a random permutation of $[m]$. Considering this algorithm, we can see that the number of calls of the function *RANDOM* is n. The arithmetic complexity is also linear. Since this algorithm works with integers less than n, the bit complexity is $O(n \log n)$.

```
σ1 ← 1
for i from 2 to n do
  σi ← i
  k ← ⌈RANDOM() ∗ i⌉
  switch σk and σi
end for
```

FIGURE 2. Algorithm *RANPER*.

Example (Subsets of size k of a set of size n, algorithms *NEXTKSB* and *RANKSB*). In this case, *RANKSB* is more difficult than *NEXKSB*. For *NEXTKSB*, it is possible to use the lexicographic order. If $k < n/2$, less than 2 operations are necessary to obtain the next subset with k elements.

If $k > n/2$, we apply the algorithm on subsets of $n-k$ elements. For *RANDKSB*, it is more complicated because k memory cells are needed to store the subsets with k elements. For this purpose, there exists a rejection algorithm with an $O(k)$ average complexity [7].

3. Ad Hoc Algorithms

For some classes of objects, general methods do not work or are not efficient. Hence, it is necessary to develop ad hoc algorithms. In this section we present several algorithms that generate random complete binary trees with $2n$ edges.

3.1. **Rémy's Algorithm.** Rémy's Algorithm [8] uses the fact that complete binary trees are in bijection with well-formed parentheses words (or Dyck words on the alphabet $A = \{x, \bar{x}\}$). The equation of the non-commutative generating series of this language is $D = \epsilon + xD\bar{x}D$. The Dyck words of length $2n$ are enumerated by the Catalan numbers:

$$\left|D \cap A^{2n}\right| = \frac{1}{n+1}\binom{2n}{n} =: C_n.$$

Hence C_n enumerates the complete binary trees with $(n+1)$ leaves, n inner vertices, and $2n$ edges.

For a complete binary tree T with $2n$ edges, we have $(2n+1)$ ways to choose one edge (if we admit that there is a virtual edge that goes to the root). Then, we can choose an orientation left or right (2 choices). We place a new vertex in the middle of the chosen edge and we add a new edge to this vertex on the left or on the right depending on the chosen orientation. If it is the virtual edge, we place above the root a "reversed chevron" ($\wedge$), so two edges, linked to the root by the right leaf or the left leaf depending of the chosen orientation (see Figure 3). We obtain a complete binary tree T' with $(2n+2)$ edges with a pointed leaf (the new added leaf). This tree T' has $n+2$ leaves. So, there are $n+2$ ways to point it, in other words, there are $n+2$ ways to obtain it from a tree with $n+1$ leaves with the described process.

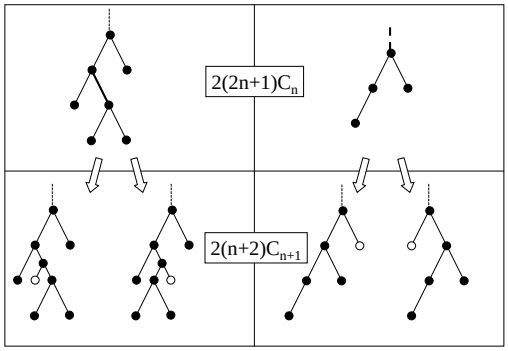


FIGURE 3. Rémy's construction.

We just proved bijectively and gave a combinatorial interpretation of the (obvious) recurrence relation $2(2n+1)C_n = (n+2)C_{n+1}$. We also proved that this process generates after m steps a random tree of size m in the class of trees of size m. By recursion, the probability of T is $1/C_n$. The probability of a pointed tree T' is $1/\big(2(2n+1)C_n\big)$. If we call T'' the tree obtained from T' while forgetting the pointing, then the probability of the tree T'' to be generated is $(n+2)/\big(2(2n+1)C_n\big)$ and so $1/C_{n+1}$. Note that this algorithm is incremental. Another advantage of this algorithm is that it manipulates numbers of order $O(n)$. Moreover it computes in linear time and memory (for fixed-size arithmetic operations).

3.2. **Algorithm based on the cyclic lemma.** There are other algorithms that can be built from a combinatorial interpretation of the identity $(2n+1)C_n = \binom{2n+1}{n}$ satisfied by Catalan numbers. For this identity we use the cyclic lemma (or Raney's lemma) [6, p. 213–227]:

Lemma 1. *A word f on the alphabet $A = \{x, \bar{x}\}$ composed of n letters x and $n+1$ letters $\bar{x}$ has only one factorization $f = f'f''$ with $f' \neq \epsilon$ (in the $2n+1$ possibilities) such that $f''f'$ represents a complete binary tree with $2n$ edges (i.e., a Dyck words followed with a letter $\bar{x}$).*

In this case, we start from a random word composed of n letters x and $n+1$ letters $\bar{x}$ (it is easy to build such a word since it corresponds with a subset of n elements of a set of $2n+1$ elements). Then we look for the unique possible factorization. One can remark that this algorithm is not incremental.

Another identity we can use is $(n+1)C_n = \binom{2n}{n}$, proved by the Catalan factorization [3].

3.3. **Step-by-step random generation of Dyck words.** Let L be a language on an alphabet A. Let L_n be the set of words of L of length n. For a word w in L, a letter a in A, and an integer n, let us define $p(w, a, n)$ as the ratio of the number of words in L_n beginning with wa over the number of words in L_n beginning with w. Using this function, it is possible to generate a word uniformly:

$w \leftarrow \epsilon$
while $|w| < n$ **do**
  a $\leftarrow$ a random letter with probability $p(w, a, n)$
  $w \leftarrow wa$
end while

FIGURE 4. Algorithm to compute a uniform random word.

This method can be efficiently applied to generate Dyck words, and therefore complete binary trees. For this purpose, let us assume that we have generated a left factor w of a Dyck word on the alphabet $A = \{x, \bar{x}\}$. This word is composed of p letters x and q letters $\bar{x}$ with $p+q = 2n-m \leq 2n$, $p-q = h \geq 0$ and such that m and h have the same parity. The number of Dyck words beginning with p letters x and q letters $\bar{x}$ is equal to the number of left factors of Dyck words with p letters x and q letters $\bar{x}$ times the number of left factors of Dyck words of length m with h more x than $\bar{x}$:

$$F_{h,m} = \frac{h+1}{m+1}\binom{m+1}{(m-h)/2}.$$

By induction we suppose that w is selected such that all Dyck words ww' have probability $1/C_n$ to appear. The probability of w is $F_{n,m}/C_n$. Now a letter x is selected with probability $\frac{h+2}{h+1}\frac{m-h}{2m}$ and a letter $\bar{x}$ is selected with probability $\frac{h}{h+1}\frac{m+h+2}{2m}$. With such probabilities, the probability of the left factor wx is equal to the probability of the left factor w to appear times the probability of the letter x:

$$\frac{\frac{h+2}{h+1}\frac{m-h}{2m}\frac{h+1}{m+1}\binom{m+1}{(m-h)/2}}{C_n} = \frac{\frac{h+2}{m}\binom{m}{(m-h-2)/2}}{C_n} = \frac{F_{h+1,m-1}}{C_n}$$

Similarly, the probability for the left factor $w\bar{x}$ is equal to the probability of w to be selected times the probability of the letter $\bar{x}$ to be selected:

$$\frac{\frac{h}{h+1}\frac{m+h+2}{2m}\frac{h+1}{m+1}\binom{m+1}{(m-h)/2}}{C_n} = \frac{\frac{h}{m}\binom{m}{(m-h)/2}}{C_n} = \frac{F_{h-1,m-1}}{C_n}.$$

We set the expected probabilities. This proves the uniformity of the distribution.

4. **Rejection Algorithms**

Assume we want to uniformly generate an object of a set S_1. The idea is to uniformly generate an object e in a set S_2 such that $S_1 \subset S_2$. If $e \in S_1$ then we keep e, otherwise we reject e and we try to select another one. This method assumes than we know how to select efficiently objects in S_2, that the ratio $|S_2|/|S_1|$ is not too big, and that we can test membership to S_1 efficiently.

4.1. **Left factor of Motzkin words.** Here is an example of rejection algorithm that computes left factor of Motzkin words [2]. Motzkin's language M is composed of words f on the alphabet $A = \{x, \bar{x}, a\}$ such that the subset of f composed only of letters x and $\bar{x}$ is a Dyck word. The idea is to generate a word, letter by letter, with probability 1/3 for each letter until it reaches the length n or until there is more letters $\bar{x}$ than letters x. In this last case, the partially built word is rejected and the algorithm is started again.

To evaluate the complexity of this algorithm, we enumerate the average number of letters generated before a word of length n in F is obtained. The language M satisfies the equation $M = \epsilon + aM + xM\bar{x}M$. So, the generating function $M(t)$ of M satisfies the equation $M(t) = 1 + tM(t) + t^2M(t)^2$ and is equal to $\left(1 - t - \sqrt{(1+t)(1-3t)}\right)/(2t^2)$. The language F of left factors of Motzkin words satisfies the equation $F = M + MxF$. So, the generating function $F(t)$ of F satisfies the equation $F(t) = M(t) + tM(t)F(t)$ and is equal to $\left(-1 + \sqrt{\frac{1+t}{1-3t}}\right)/(2t)$.

Let R_n be the language of rejected words by the algorithm and let $F_{\leq p}$ be the language of words of F of length less or equal to p. One can remark that $R_n = F_{\leq n-1}A \setminus F_{\leq n}$ and that $\lim_{n\to\infty} R_n = M\bar{x}$. The algorithm generates words of the language $G = F_n + R_nG$. The generating function $G(t)$ of G is equal to $G(t) = \frac{f_n t^n}{1 - Rn(t)}$ where $R_n(t)$ is the generating function of R_n and where $F(t)$ is the generating function of F.

Let $P_G(t)$ the probability generating function of G, i.e., the generating function where each word is weighed by its probability; the average length $\gamma(G)$ of words of G is $P'_G(1)$. In formulas:

$$P_G(t) = G(t/3) = \frac{f_n t^n}{3^n\big(1 - R_n(t/3)\big)} \quad \text{and} \quad P'_G(t) = \frac{n f_n t^{n-1}}{3^n\big(1 - R_n(t/3)\big)} + \frac{f_n t^n R'_n(t/3)}{3^{n+1}\big(1 - R_n(t/3)\big)^2}.$$

One can remark that $A^n = F_n \cup \bigcup_{i=1}^n R_n^{(i)} A^{n-i}$ where $R_n^{(i)} = R_n \cap A^i$. If we note $r_n^{(i)}$ the cardinality of $R_n^{(i)}$, then $3^n = f_n + \sum_{i=1}^n r_n^{(i)} 3^{n-i}$ and so $f_n/3^n = 1 - R_n(1/3)$ and we get $P'_G(1) = n + \frac{3^{n-1}}{f_n} R'_n(1/3)$. But, as $R'_n(1/3) = \sum_{i=1}^n i r_n^{(i)} 3^{-i+1} = k\lambda(R_n)$, we get $P'_G(1) = n + \frac{3^n}{f_n}\lambda(R_n)$ where $\lambda(R_n) = \lambda(F_{\leq n-1} \setminus F_{\leq n}) = \sum_{i=0}^n i \frac{k f_{i-1} - f_i}{3^i} = \sum_{i=0}^n i \frac{f_i}{3^i} - n\frac{f_n}{3^n}$. Using both equations above, we obtain:

$$P'_G(1) = \frac{[t^n]\left(\frac{1}{1-t}F(t/3)\right)}{[t^n]\,F(t/3)}, \quad [t^n]\,F(t/3) = \frac{\sqrt{3}}{\sqrt{\pi n}} + O\left(\frac{1}{n}\right), \quad [t^n]\left(\frac{1}{1-t}F(t/3)\right) = \frac{2\sqrt{3n}}{\sqrt{\pi}} + O(1).$$

Therefore when n goes to infinity, the average number of selected letters converges to $2n$. Alain Denise has extended this rejection method to the fg-languages [4].

4.2. **Motzkin words.** Let us consider a Motzkin word of length n. This is a word of parentheses on $\{x, \bar{x}\}$ of length $2i \leq n$ with $n - 2i$ letters a intertwined. The Motzkin words of length n are enumerated by Motzkin numbers $m_n = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} C_i$ where C_i is a Catalan number. To select a Motzkin word of length n uniformly, the problem is to decide the number i of letters x in the word. Then the problem is easy to solve, as we know how to select a Dyck word length $2i$ and we know how to select $2i$ positions in n possible ones where the letters of this word will be inserted.

The probability that a word has i letters x and i letters $\bar{x}$ is $\binom{n}{2i}C_i/m_n$. To generate i with the appropriate distribution, using the formula, it is necessary to manipulate huge numbers. The idea of Laurent Alonso [1] is to approximate this distribution by a larger distribution, easy to simulate. This idea can also be found in the Luc Devroye's book [5]. Assume that we have $v+1$ boxes numbered from 0 to v. Box i contains N_i black balls ($N = \sum_{i=0}^{v} N_i$). This is the initial distribution. For each $i = 0, 1, \ldots, v$, we add B_i white balls into box i. Globally, there are D balls ($D = \sum_{i=0}^{v} D_i$, $D_i = N_i + B_i$). This is an easy distribution to compute. We select box i with probability D_i/D. Then we consider that this choice is correct with probability N_i/D_i (the probability to select a black ball in the box i). If this choice is not correct, we choose another box. Otherwise the integer i is definitively selected. The probability to select the box i with such process is N_i/N and the average number of trials before a box is definitively selected is D/N.

For Motzkin's language, L. Alonso [1] takes $v = n + 1 - \lfloor (n+1)/3 \rfloor$,

$$N_i = \begin{cases} \frac{n!}{(i-1)!\, i!\, (n-2i+2)!} & \text{for } i \in [\,1, 1+n/2\,], \\ 0 & \text{otherwise,} \end{cases}$$

and

$$D_i = \frac{n!}{\lfloor (n+1)/3 \rfloor !\, i!\, \big(n+1-i-\lfloor (n+1)/3 \rfloor\big)!}.$$

Note that D_i is mainly a binomial coefficient times a constant.

We can show that when $i \in [\,1, 1+n/2\,]$, we have $N_i/D_i = \binom{a}{c}/\binom{b}{c} \leq 1$ where the values of a, b, and c depend only of the position of i from $\lfloor (n+1)/3 \rfloor + 1$. The choice of box i with probability D_i/D can be done by generating a sequence of $n+1-\lfloor (n+1)/3 \rfloor$ bits and considering the sum of generated bits. The validity test of the choice of a box with a probability $N_i/D_i = \binom{a}{c}/\binom{b}{c}$ can be done by choosing c integers in the interval $[\,1, b\,]$ and verifying that they are less than a. We can compute that

$$D \sim \frac{3^{n+2}}{2n^{3/2}\sqrt{\pi}}, \qquad M \sim \frac{3^{n+1}\sqrt{3}}{2n^{3/2}\sqrt{\pi}}.$$

This implies the result $D/M \sim \sqrt{3}$.

Theorem 1. [1] *The average complexity of the random generating algorithm of Motzkin words is linear.*

Bibliography

[1] Alonso (Laurent) and Schott (René). – *Random generation of trees*. – Kluwer Academic Publishers, Boston, MA, 1995, x+208p. Random generators in computer science.

[2] Barcucci (E.), Pinzani (R.), and Sprugnoli (R.). – The random generation of directed animals. *Theoretical Computer Science*, vol. 127, n° 2, 1994, pp. 333–350.

[3] Chottin (Laurent) and Cori (Robert). – Une preuve combinatoire de la rationalité d'une série génératrice associée aux arbres. *RAIRO Informatique Théorique*, vol. 16, n° 2, 1982, pp. 113–128.

[4] Denise (Alain). – *Méthodes de génération aléatoire d'objets combinatoires de grande taille et problèmes d'énumération*. – Thèse, Université Bordeaux I, 1994.

[5] Devroye (Luc). – *Nonuniform random variate generation*. – Springer-Verlag, New York, 1986, xvi+843p.

[6] Lothaire (M.). – *Combinatorics on words*. – Addison-Wesley, Reading, Mass., 1983, *Encyclopedia of Mathematics and its Applications*, vol. 17, xix+238p. Collective work under a pseudonym.

[7] Nijenhuis (Albert) and Wilf (Herbert S.). – *Combinatorial algorithms*. – Academic Press, New York, 1978, second edition, *Computer Science and Applied Mathematics*, xv+302p. For computers and calculators.

[8] Rémy (Jean-Luc). – Un procédé itératif de dénombrement d'arbres binaires et son application à leur génération aléatoire. *RAIRO Informatique Théorique*, vol. 19, n° 2, 1985, pp. 179–195.

Contents

Part I. Combinatorics

The Site Perimeter of Bargraphs.
Talk by M. Bousquet-Mélou, summary by S. Corteel 3

Animals, Domino Tilings, Functional Equations.
Talk by M. Bousquet-Mélou, summary by C. Banderier 7

Counting Domino Tilings of Rectangles via Resultants.
Talk by V. Strehl, summary by S. Corteel 13

Random Generation from Boltzmann Principles.
Talk by Ph. Flajolet, summary by M. Pelletier and M. Soria 17

A Relaxed Approach to Tree Generation.
Talk by Ph. Duchon, summary by M. Mishna 19

Symmetric Functions and P-Recursiveness.
Talk by M. Mishna, summary by H. Crapo 23

Part II. Symbolic Computation

Computation of the Inverse and Determinant of a Matrix.
Talk by G. Villard, summary by E. Thomé 29

Fast Algorithms for Polynomial Systems Solving.
Talk by A. Bostan, summary by F. Chyzak 33

Transseries Solutions of Algebraic Differential Equations.
Talk by J. van der Hoeven, summary by A. Fredet 37

Recent Algorithms for Solving Second-Order Differential Equations.
Talk by J.-A. Weil, summary by M. Loday-Richaud 43

The Structure of Multivariate Hypergeometric Terms.
Talk by M. Petkovšek, summary by B. Salvy 47

Numerical Elimination, Newton Method and Multiple Roots.
Talk by J.-C. Yakoubsohn, summary by B. Salvy 49

Part III. Analysis of Algorithms, Data Structures, and Network Protocols

Everything You Always Wanted to Know about Quicksort, but Were Afraid to Ask.
Talk by M. Durand, summary by M. Nguyễn-Thế 57

Traveling Waves and the Height of Binary Search Trees.
Talk by M. Drmota, summary by B. Chauvin 63

Microscopic Behavior of TCP.
Talk by Ph. Robert, summary by Ch. Fricker 69

Interaction Between Sources Controlled by TCP.
Talk by F. Baccelli 73

Asymptotic Analysis of TCP Performances Under Mean-field Approximation.
Talk by Ph. Jacquet, summary by Ch. Fricker 75

Part IV. Asymptotics and Analysis

A Hyperasymptotic Approach of the Multi-Dimensional Saddle-Point Method.
Talk by É. Delabaere, summary by M. Durand 79

Ramanujan's Summation.
Talk by É. Delabaere, summary by V. Puyhaubert 83

Multi-Variable sinc Integrals and the Volumes of Polyhedra.
Talk by J. Borwein, summary by L. Meunier 89

Part V. Number Theory

L-Series of Squares of Squares.
Talk by J. Borwein 95

Irrationality of the ζ Function on Odd Integers.
Talk by T. Rivoal, summary by M. Durand 97

Irrationality Measures of $\log 2$ and $\pi/\sqrt{3}$.
Talk by N. Brisebarre, summary by B. Salvy 101

Part VI. Miscellany

Approximate Matching of Secondary Structures.
Talk by M. Raffinot, summary by P. Nicodème 107

Les algorithmes évolutionnaires : état de l'art et enjeux *(Evolutionary Algorithms: State of the Art and Stakes).*
Talk by M. Schoenauer, summary by Ph. Dumas 113

Part VII. ALEA'2002 Lecture Notes

Systèmes dynamiques et algorithmique *(Dynamical Systems and Algorithms).*
Talk by V. Baladi and B. Vallée, summary by F. Chazal, V. Maume-Deschamps, and B. Vallée . 121

Martingales discrètes et applications à l'analyse d'algorithmes *(Discrete Martingales Applied to Algorithms Analysis).*
Talk by B. Chauvin, summary by B. Chauvin 151

Phase Transitions and Satisfiability Threshold.
Talk by O. Dubois and V. Puyhaubert, summary by V. Puyhaubert 167

Génération aléatoire *(Random Generation).*
Talk by A. Denise, summary by S. Corteel 173

Combinatorics and Random Generation.
Talk by D. Gouyou-Beauchamps, summary by N. Bonichon 177

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS
Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
(France)
http://www.inria.fr
ISSN 0249-6399