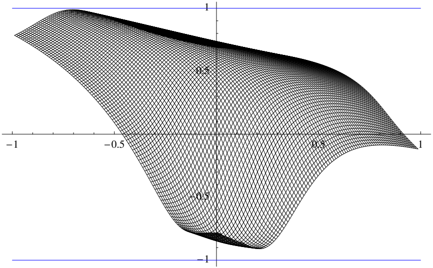
SPQR Method:
a new linear-time exact sampler of combinatorial structures

Andrea Sportiello
CNRS | LIPN, Université Paris Nord, Villetaneuse
work in collaboration with Frédérique Bassino

ALÉA 2014
CIRM, Luminy, 21st March 2014



Algorithms is `coding`

Complex analysis is only for the analysis of algorithms

(and, in fact, only the very fine structure of it)

If I'm happy with rough estimates

and heuristic performance analysis,

I can just live without

Algorithms is `coding`

Complex analysis is only for the analysis of algorithms

(and, in fact, only the very fine structure of it)

If I'm happy with rough estimates

and heuristic performance analysis,

I can just live without

FALSE!

Algorithms is `coding`

Complex analysis is only for the analysis of algorithms

(and, in fact, only the very fine structure of it)

If I'm happy with rough estimates

and heuristic performance analysis,

I can just live without

FALSE!

A first example: in the Boltzmann method you need complex analysis in the preprocessing, for finding "the value of the oracle"

Algorithms is coding
Complex analysis is only for the analysis of algorithms
(and, in fact, only the very fine structure of it)
If I'm happy with rough estimates
and heuristic performance analysis,
I can just live without

FALSE!

A first example: in the Boltzmann method you need complex analysis in the preprocessing, for finding "the value of the oracle"

In this talk you see a more striking example: complex analysis is used all the time along the core part of the algorithm

Algorithms is `coding`

Complex analysis is only for the analysis of algorithms
(and, in fact, only the very fine structure of it)
If I'm happy with rough estimates
and heuristic performance analysis,
I can just live without

FALSE!

A first example: in the Boltzmann method you need complex analysis in the preprocessing, for finding "the value of the oracle"

In this talk you see a more striking example: complex analysis is used all the time along the core part of the algorithm

Moral: if complex analysis "knows the truth" on the asymptotics of your random structures, (and it's the only one who knows), no surprise that algorithms not using it have worse performances. . .

Part 1

An introduction to Exact Sampling

(with a zest of Statistical Mechanics)

Part 1

An introduction to Exact Sampling

(with a zest of Statistical Mechanics)

Exact sampling

Our goal today is the exact sampling
of large random combinatorial structures.

Large: "size n". We want to do that fast.

In many cases, it is obvious that you can do that
in $T(n) \sim \exp(\alpha n)$ or $T(n) \sim \exp(\alpha n \ln n)$.
And you are much happier with a polynomial algorithm, $T(n) \sim n^{\gamma}$.

This is what happens, for example, with Coupling From The Past
of Propp and Wilson (used e.g. for the Potts Model), or with
Wilson's cycle-popping algorithm for Uniform Spanning Trees.

However, if the problem is easy, we want to do that really fast:
in quasi-linear time $T(n) \sim n \cdot (\ln n)^{\gamma}$.

A prototype of easy problem

What do we mean by "if the problem is easy" ?

A typical example of an easy problem is directed walks.
Let's do that in $D = 2$, just to be definite.
You have some "nice" functions $h_{x,y}, v_{x,y} : \mathbb{N}^2 \to \mathbb{R}^+$,
and you want to sample paths $\omega : (0,0) \to (n-m, m)$,
according to the unnormalised measure

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet_{(x,y)}}} v_{x,y} \prod_{(x,y)\bullet\!\!\to} h_{x,y}$$

Examples:

- directed walks (binomials): $h_{x,y} = v_{x,y} = 1$.
- directed walks weighted with their area (q-binomials):
  $h_{x,y} = q^y;\ v_{x,y} = 1$.
- $\mathcal{P}(n,m) \equiv$ partitions of $[n]$ into m parts (Stirling of 2nd kind):
  $h_{x,y} = y;\ v_{x,y} = 1$.

A prototype of easy problem

What do we mean by "if the problem is easy" ?

A typical example of an easy problem is directed walks.
Let's do that in $D = 2$, just to be definite.
You have some "nice" functions $h_{x,y}, v_{x,y} : \mathbb{N}^2 \to \mathbb{R}^+$,
and you want to sample paths $\omega : (0,0) \to (n-m, m)$,
according to the unnormalised measure

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet_{(x,y)}}} v_{x,y} \prod_{(x,y)\bullet\!\!\longrightarrow} h_{x,y}$$

Examples:

- directed walks (binomials): $h_{x,y} = v_{x,y} = 1$.
- directed walks weighted with their area (q-binomials):
  $h_{x,y} = q^y$; $v_{x,y} = 1$.
- $\mathcal{P}(n,m) \equiv$ partitions of $[n]$ into m parts (Stirling of 2nd kind):
  $h_{x,y} = y$; $v_{x,y} = 1$.

A prototype of easy problem

What do we mean by "if the problem is easy" ?

A typical example of an easy problem is directed walks.
Let's do that in $D = 2$, just to be definite.
You have some "nice" functions $h_{x,y}, v_{x,y} : \mathbb{N}^2 \to \mathbb{R}^+$,
and you want to sample paths $\omega : (0,0) \to (n-m, m)$,
according to the unnormalised measure

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet_{(x,y)}}} v_{x,y} \prod_{(x,y) \bullet\!\!\longrightarrow} h_{x,y}$$

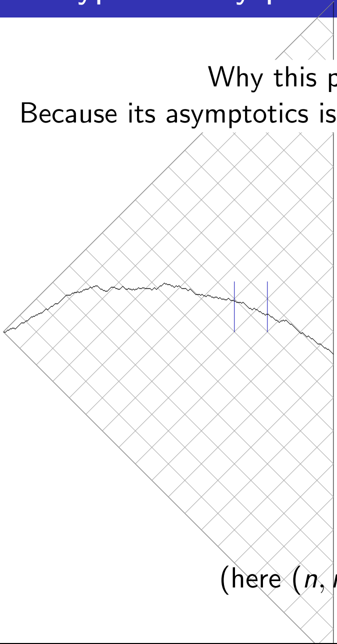
Examples:

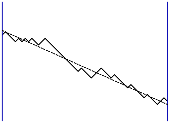
- directed walks (binomials): $h_{x,y} = v_{x,y} = 1$. ✔
- directed walks weighted with their area (q-binomials): ✔
  $h_{x,y} = q^y$; $v_{x,y} = 1$.
- $\mathcal{P}(n, m) \equiv$ partitions of $[n]$ into m parts (Stirling of 2nd kind): ✘
  $h_{x,y} = y$; $v_{x,y} = 1$.

A prototype of easy problem

Why this problem *must* be easy?

Because its asymptotics is given by calculus of variations in 1D:





$1 \ll \ell \ll n$

At scales $1 \ll \ell \ll n$, a typical path looks like a random walk, with some drift, diffusion constant and vertical offset.

(here $(n, m) = (500, 267)$, $h_{x,y} = (1.03)^y$, $v_{x,y} = x$)

A prototype of easy problem

Why this problem *must* be easy?

Because its asymptotics is given by calculus of variations in 1D:

$$\text{Call } U\left(\tfrac{x+y}{2}, \tfrac{y-x}{2}\right) = \tfrac{1}{2} \ln\left(v_{x,y-1}/h_{x-1,y}\right)$$

$$\text{Call } V\left(\tfrac{x+y}{2}, \tfrac{y-x}{2}\right) = \tfrac{1}{2} \ln\left(v_{x,y-1} \cdot h_{x-1,y}\right)$$

$$\text{Call } s(x) = -\tfrac{1+x}{2} \ln \tfrac{1+x}{2} - \tfrac{1-x}{2} \ln \tfrac{1-x}{2}$$

(Shannon entropy of a binary stream with probabilities $\frac{1\pm x}{2}$)

Then the limit profile $\phi(t)$ maximizes the functional

$$S_\lambda[\phi] = \int_0^1 \mathrm{d}t \left[s(\phi'(t)) + \lambda\phi'(t) + \phi'(t) U(t, \phi(t)) + V(t, \phi(t)) \right]$$

with λ determined by the constraint $\mathbb{E}(\phi(1)) = \frac{2m-n}{n}$.

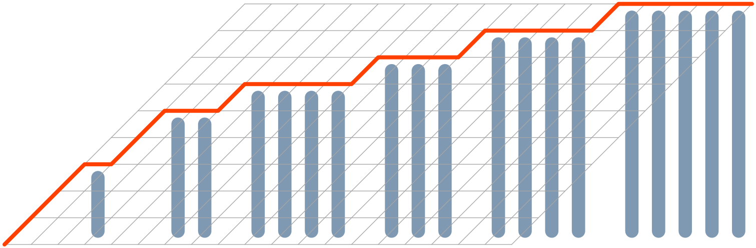
Finally, $Z_{n,m} = \exp\left((n-2m)\lambda + nS[\phi^*] + o(n)\right)$.

A digression on Random Minimal Automata

Why shall we care of (inhomogeneous) directed random walks?
Because sometimes they are in bijection with more interesting objects

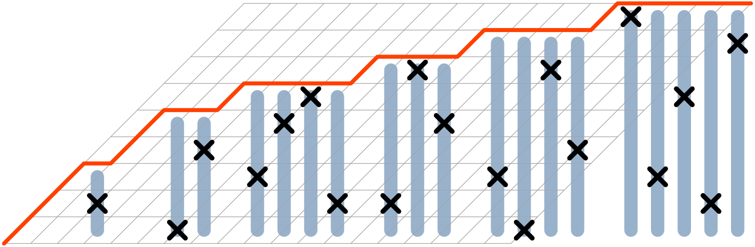
A digression on Random Minimal Automata

In our case, directed paths with $h_{x,y} = y$ can be interpreted as paths ω, $\times$ a choice $Y(x) \in \{1, \dots, y(x)\}$ per horizontal step. Pairs (ω, Y) are in bijection with $\pi \in \mathcal{P}(n, m)$, i.e. partitions of $[n]$ into m non-empty blocks.



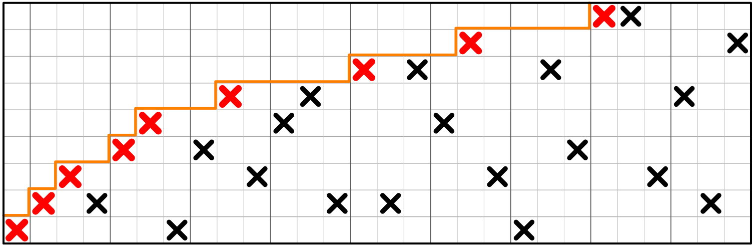
A digression on Random Minimal Automata

In our case, directed paths with $h_{x,y} = y$ can be interpreted as paths ω, $\times$ a choice $Y(x) \in \{1, \dots, y(x)\}$ per horizontal step. Pairs (ω, Y) are in bijection with $\pi \in \mathcal{P}(n, m)$, i.e. partitions of $[n]$ into m non-empty blocks.



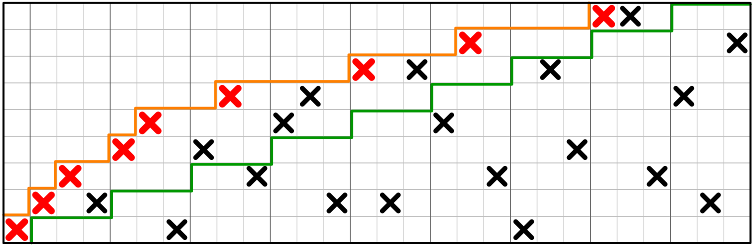
A digression on Random Minimal Automata

In our case, directed paths with $h_{x,y} = y$ can be interpreted as paths ω, $\times$ a choice $Y(x) \in \{1, \dots, y(x)\}$ per horizontal step. Pairs (ω, Y) are in bijection with $\pi \in \mathcal{P}(n, m)$, i.e. partitions of $[n]$ into m non-empty blocks.



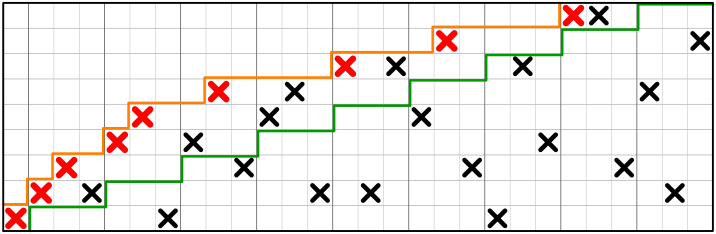
A digression on Random Minimal Automata

In our case, directed paths with $h_{x,y} = y$ can be interpreted as paths ω, $\times$ a choice $Y(x) \in \{1, \dots, y(x)\}$ per horizontal step. Pairs (ω, Y) are in bijection with $\pi \in \mathcal{P}(n, m)$, i.e. partitions of $[n]$ into m non-empty blocks.



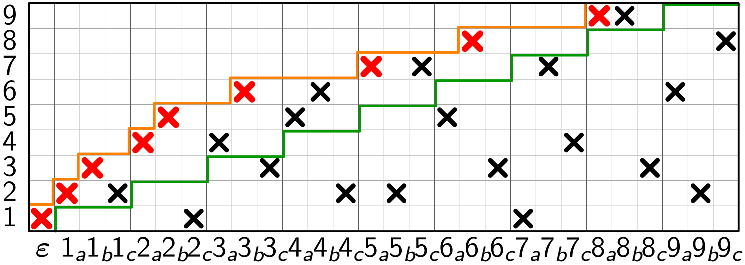
For $n = km + 1$, a $\mathcal{O}(1)$ subset of this set (those which are "k-Dyck") is in bijection with accessible deterministic complete automata (ADCA), with m states and alphabet of size k.

A digression on Random Minimal Automata

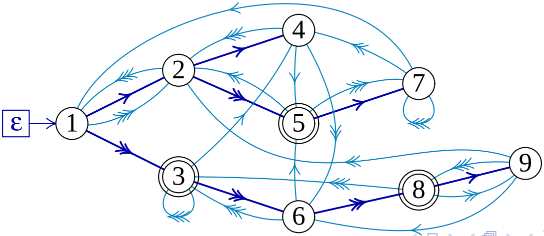


A $\mathcal{O}(1)$ fraction of $\mathcal{P}(km+1, m)$ (k-Dyck partitions)
is in bijection with ADCA's, on m states and alphabet of size k.

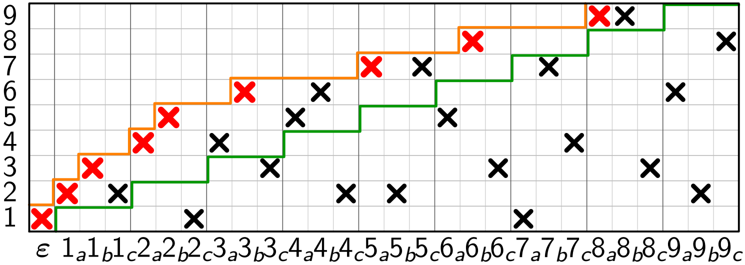
A digression on Random Minimal Automata



A $\mathcal{O}(1)$ fraction of $\mathcal{P}(km+1, m)$ (k-Dyck partitions)
is in bijection with ADCA's, on m states and alphabet of size k.



A digression on Random Minimal Automata



A $\mathcal{O}(1)$ fraction of $\mathcal{P}(km+1, m)$ (k-Dyck partitions) is in bijection with ADCA's, on m states and alphabet of size k.

Those which are not k-Dyck, with probability $1 - \exp(-cs)/\sqrt{s}$ cross the diagonal within the last s steps.

Thus, in a sampling algorithm "starting from the top-left corner" (as will be our own), are sampled quite efficiently through anticipated rejection.

A digression on Random Minimal Automata

A $\mathcal{O}(1)$ fraction of ADCA's (in fact, $1 - o(1)$ if $k \geq 3$) are minimal.

Thus, if we have quasi-linear exact sampling algorithm for ω,
we have a quasi-linear exact sampling algorithm
for random uniform minimal automata.

A digression on Random Minimal Automata

A $\mathcal{O}(1)$ fraction of ADCA's (in fact, $1 - o(1)$ if $k \geq 3$) are minimal.

At $k = 2$, if we use a modified walk, taking columns in pairs, and steps with weights $w(\nearrow, \rightarrow, \searrow) = (1, 2y, y^2 - \beta x)$ (no pairs of columns have the same marks, and boolean status), we get a fraction $1 - o(1)$ of minimal automata within ADCA's

$$|\mathcal{P}(km+1, m)| = \left\{ \begin{matrix} km+1 \\ m \end{matrix} \right\} \sim \frac{(km+1)!}{m!} \exp(m \cdot a(k))$$
$$\sim 2^{(k-1)m \log_2 m} \exp(m \cdot a'(k))$$

The complexity for the sole Buffon procedure for sampling the "black marks" is $\sim 2^{(k-1)m \langle \log_2 y \rangle} \sim 2^{(k-1)m \log_2 m} \exp(m \cdot a''(k))$

➠ if we have an exact sampling algorithm for $\omega : (0,0) \rightarrow (\alpha m, m)$ with complexity $o(m \ln m)$, we have an **optimal** exact sampling algorithm for random uniform minimal automata on any alphabet size.

Directed walks: recursion for Z

Let us call $Z_{n,m}$ the normalisation factor

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet_{(x,y)}}} v_{x,y} \prod_{(x,y)\bullet\!\!\longrightarrow} h_{x,y} \qquad Z_{n,m} = \sum_{\omega:(0,0)\to(n-m,m)} \mu(\omega)$$

(Z stands for "Zustandssumme", as first introduced by our Austrian friend Ludwig Boltzmann. . .)

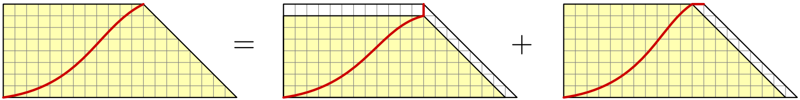
Directed walks: recursion for Z

Let us call $Z_{n,m}$ the normalisation factor

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet\,(x,y)}} v_{x,y} \prod_{(x,y)\,\bullet\!\!\longrightarrow} h_{x,y} \qquad Z_{n,m} = \sum_{\omega:(0,0)\to(n-m,m)} \mu(\omega)$$

Of course we have

$$Z_{n,m} = v_{n-m,m-1}\, Z_{n-1,m-1} + h_{n-m-1,m}\, Z_{n-1,m}$$



Directed walks: recursion for Z

Let us call $Z_{n,m}$ the normalisation factor

$$\mu(\omega) = \prod_{\substack{\uparrow \\ \bullet (x,y)}} v_{x,y} \prod_{(x,y) \bullet\!\!\longrightarrow} h_{x,y} \qquad Z_{n,m} = \sum_{\omega:(0,0)\to(n-m,m)} \mu(\omega)$$

Of course we have

$$Z_{n,m} = v_{n-m,m-1}\, Z_{n-1,m-1} + h_{n-m-1,m}\, Z_{n-1,m}$$

And in fact, for our three examples

$$\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m};$$

$$\begin{bmatrix} n \\ m \end{bmatrix}_q = \begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q + q^m \begin{bmatrix} n-1 \\ m \end{bmatrix}_q;$$

$$\left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \left\{ \begin{matrix} n-1 \\ m-1 \end{matrix} \right\} + m \left\{ \begin{matrix} n-1 \\ m \end{matrix} \right\};$$

Directed walks: recursion for Z

Let us call $Z_{n,m}$ the normalisation factor

$$\mu(\omega) = \prod_{\overset{\uparrow}{\bullet}_{(x,y)}} v_{x,y} \prod_{(x,y)\,\bullet\!\!\longrightarrow} h_{x,y} \qquad Z_{n,m} = \sum_{\omega:(0,0)\to(n-m,m)} \mu(\omega)$$

Of course we have

$$Z_{n,m} = v_{n-m,m-1}\, Z_{n-1,m-1} + h_{n-m-1,m}\, Z_{n-1,m}$$

What's the difference, then?

Directed walks: recursion for Z

Let us call $Z_{n,m}$ the normalisation factor

$$\mu(\omega) = \prod_{\stackrel{\uparrow}{\bullet}\,(x,y)} v_{x,y} \prod_{(x,y)\,\bullet\!\!\longrightarrow} h_{x,y} \qquad Z_{n,m} = \sum_{\omega:(0,0)\to(n-m,m)} \mu(\omega)$$

Of course we have

$$Z_{n,m} = v_{n-m,m-1}\, Z_{n-1,m-1} + h_{n-m-1,m}\, Z_{n-1,m}$$

…let's try to solve the recursion …

$$\binom{n}{m} = \frac{n(n-1)\cdots(n-m+1)}{m(m-1)\cdots 1} \, ;$$

$$\begin{bmatrix} n \\ m \end{bmatrix}_q = \frac{(1-q^n)(1-q^{n-1})\cdots(1-q^{n-m+1})}{(1-q^m)(1-q^{m-1})\cdots(1-q)} \, ;$$

$\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$: **no factorisation!**

An application of the recursive method

What can we do with a factorised formula?
We can perform an exact sampling through a Recursive Method...

A walk ω is a string $(s_1, \dots, s_n)$ in $\{\uparrow, \rightarrow\}^n$.

Define $p_{n,m} = Z_{n-1,m-1}/Z_{n,m}$. Suppose you can evaluate $p_{n,m}$ (exactly) through an oracle that requires a time τ_n.

This trivial algorithm then has complexity

$$T(n) = \Theta(n) + \sum_{i=1}^{n} \tau_i \lesssim n\,\tau_n$$

```
n' = n; m' = m;
while n' > 0 do
    if Bern_{p_{n',m'}} then
        s_{n'} = ↑; m' = m' − 1;
    else
        s_{n'} = →;
    end
end
```

An application of the recursive method

The role of the factorised formulas
is in the production of the oracle $\mathrm{Bern}_{p_{n,m}}$, with
$p_{n,m} = Z_{n-1,m-1}/Z_{n,m}$.

reminder:

$$\binom{n}{m} = \frac{n(n-1)\cdots(n-m+1)}{m(m-1)\cdots 1};$$

$$\begin{bmatrix} n \\ m \end{bmatrix}_q = \frac{(1-q^n)(1-q^{n-1})\cdots(1-q^{n-m+1})}{(1-q^m)(1-q^{m-1})\cdots(1-q)};$$

$\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$: **no factorisation!**

Thus, for binomials you need a Buffon machine for rationals
for q-binomials you need a Buffon machine for ratios of q-numbers,
but for Stirling of 2nd kind you are stuck!

An application of the recursive method

The role of the factorised formulas
is in the production of the oracle $\mathrm{Bern}_{p_{n,m}}$, with
$p_{n,m} = Z_{n-1,m-1}/Z_{n,m}$.

thus

$$\binom{n-1}{m-1} \Bigg/ \binom{n}{m} = \frac{m}{n}\,;$$

$$\begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q \Bigg/ \begin{bmatrix} n \\ m \end{bmatrix}_q = \frac{1-q^m}{1-q^n}\,;$$

$$\left\{ \begin{matrix} n-1 \\ m-1 \end{matrix} \right\} \Bigg/ \left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \mathbf{???}$$

Thus, for binomials you need a Buffon machine for rationals
for q-binomials you need a Buffon machine for ratios of q-numbers,
but for Stirling of 2nd kind you are stuck!

An application of the recursive method

The role of the factorised formulas
is in the production of the oracle $\mathrm{Bern}_{p_{n,m}}$, with
$p_{n,m} = Z_{n-1,m-1}/Z_{n,m}$.

thus

$$\binom{n-1}{m-1} \Big/ \binom{n}{m} = \frac{m}{n}\,;$$
$$\begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q \Big/ \begin{bmatrix} n \\ m \end{bmatrix}_q = \frac{1-q^m}{1-q^n}\,;$$
$$\left\{ \begin{matrix} n-1 \\ m-1 \end{matrix} \right\} \Big/ \left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \textbf{???}$$

Thus, for binomials you need a Buffon machine for rationals
for q-binomials you need a Buffon machine for ratios of q-numbers,
but for Stirling of 2nd kind you are stuck!

StatMech in a nutshell

Let us recall some basic "statistical physics" (in combinatorial terms...).

In our walks ω, we have fixed the arrival point m, within the range $m \in \{0, 1, \dots, n\}$ of possible values. Thus we are in the canonical ensemble, with unnormalised measure $\mu_m(\omega)$.

If we had walks ω with non-prescribed arrival point, by setting in the grand-canonical ensemble. Doing this, we are naturally induced to introduce a Lagrange multiplier, and have unnormalised measure $\mu^\lambda(\omega) = \sum_m e^{\lambda m} \mu_m(\omega)$.

In large n, marginals of finitely many extensive statistics are dominated by saddle points: $Z(x, y, \dots) \sim \exp\left[nF(\xi, \eta, \dots)\right]$, with $\xi = x/n$, $\eta = y/n$, ...

StatMech in a nutshell

In large n, marginals are dominated by saddle points:
$$Z(x, y, \dots) \sim \exp\left[nF(\xi, \eta, \dots)\right]$$
and in particular, (call $\alpha = m/n$)
$$Z_\lambda(x, y, \dots) = \sum_m e^{\lambda m} Z_m(x, y, \dots) \sim \int \mathrm{d}\alpha \exp\left[n(\lambda\alpha + F(\xi, \eta, \dots, \alpha)\right]$$

Laplace transform on Z is "tropicalised" into Legendre transform on F (the $(+, \times)$ ring is transformed into $(\max, +)$)

The Legendre transform is involutive, (cf. Fourier analogous property) but only if f is convex, otherwise you get the convexified of f.

BTW, this is the origin of phase coexistence in Nature (liquid water and vapour coexisting at 100°, instead of a mixture with density $0.5\, g/cm^3$).

The Boltzmann method in a nutshell

The Boltzmann method for exact sampling works when:

- you have a sampler in the grand-canonical ensemble (but not in the canonical one);
- your goal value α is in a region where the convexified of $F(\alpha)$ coincides with $F(\alpha)$.
- you can find the appropriate "Legendre-dual" parameter $\lambda^* = \lambda(\alpha)$ (the "problem of the oracle").

The law of large numbers is underlying here. Even at the optimal value λ^*, you only get the random variable $m' = m + \mathcal{O}(\sqrt{m})$. The complexity is, in the best of cases, $T(n) \sim n^{\frac{3}{2}}$.

If you ask more extensive statistics to have a certain given value, you get a $\frac{1}{2}$ in the exponent per parameter.

Still, you may have an algorithm. And it may be your best choice so far.

An application of the Boltzmann method

Let's see how this works for our directed walks...

Assume for simplicity that $h_{x,y} = h_y$, and $v_{x,y} = 1$
Consider the grandcanonical ensemble of walks that have exactly $m \uparrow$, and a variable number of $\rightarrow$, with a Lagrange multiplier λ.

$$\omega = (\underbrace{\rightarrow \cdots \rightarrow}_{c_0} \uparrow \underbrace{\rightarrow \cdots \rightarrow}_{c_1} \uparrow \ldots \uparrow \underbrace{\rightarrow \cdots \rightarrow}_{c_m})$$

thus we have $\mu_\lambda(c_0, c_1, \ldots, c_m) = e^{\lambda \sum_y c_y} \prod_{y=0}^m h_y^{c_y}$,
and $n = m + \sum_y c_y$.

Each c_y is an independent geometric variable,
with average $\frac{e^\lambda h_y}{1-e^\lambda h_y}$ and variance $\frac{e^\lambda h_y(1+e^\lambda h_y)}{(1-e^\lambda h_y)^2}$.

Thus, $\lambda(\alpha)$ is the solution (if any) of the equation

$$\alpha^{-1} := \frac{n}{m} = 1 + \left\langle \frac{e^\lambda h_y}{1-e^\lambda h_y} \right\rangle_{0 \leq y \leq m} = \left\langle \frac{1}{1-e^\lambda h_y} \right\rangle_{0 \leq y \leq m}$$

in the range $\lambda \in (-\infty, -\ln \max h_y)$.

An application of the Boltzmann method

As $\alpha(\lambda) : (-\infty, -\ln \max h_y) \to (0,1)$ is clearly smooth and monotone, from LNN we get easily the existence and unicity of a solution, and concentration.

This trivial algorithm then has complexity $T(n) \sim n^{\frac{3}{2}}$ (even neglecting the time for finding λ^*, and the Buffon complexity for the geometric laws)

Find a decent approx. of λ^*,
(e.g. by Newton, or by bisection);
repeat
  $n' = 0$;
  for $y = 0$ **to** m **do**
    $c_y = \mathrm{Geom}_{e^{\lambda^*} h_y}$;
    $n' = n' + c_y$;
  end
until $n' = n$;

The role of complex analysis

The Boltzmann method has a much broader range of applications than the simple case of directe walks. In its generality, the determination of λ^* is based on the formulation of $Z(\lambda)$ as a Cauchy integral, amenable for a saddle-point analysis.

We could avoid this, as we just had independent geometric variables Nonetheless, it's instructive to see what does the "standard approach" give:

$$Z_{n,m} = \sum_{\substack{(c_0,\dots,c_m) \\ \sum_y c_y = n-m}} \prod_y h_y^{c_y}$$

The role of complex analysis

The Boltzmann method has a much broader range of applications than the simple case of directe walks. In its generality, the determination of λ^* is based on the formulation of $Z(\lambda)$ as a Cauchy integral, amenable for a saddle-point analysis.

We could avoid this, as we just had independent geometric variables Nonetheless, it's instructive to see what does the "standard approach" give:

$$Z_{n,m} = \sum_{\substack{(c_0,\ldots,c_m) \\ \sum_y c_y = n-m}} \prod_y h_y^{c_y} = \oint \frac{\mathrm{d}z}{2\pi i\, z} \sum_{(c_0,\ldots,c_m)} z^{-(n-m)+\sum_y c_y} \prod_y h_y^{c_y}$$

The role of complex analysis

The Boltzmann method has a much broader range of applications than the simple case of directe walks. In its generality, the determination of λ^* is based on the formulation of $Z(\lambda)$ as a Cauchy integral, amenable for a saddle-point analysis.

We could avoid this, as we just had independent geometric variables Nonetheless, it's instructive to see what does the "standard approach" give:

$$Z_{n,m} = \sum_{\substack{(c_0,\dots,c_m) \\ \sum_y c_y = n-m}} \prod_y h_y^{c_y} = \oint \frac{\mathrm{d}z}{2\pi i\, z} \sum_{(c_0,\dots,c_m)} z^{-(n-m)+\sum_y c_y} \prod_y h_y^{c_y}$$

$$= \oint \frac{\mathrm{d}z}{2\pi i\, z} z^{-(n-m)} \prod_{y=0}^{m} \sum_{c_y} (z h_y)^{c_y}$$

The role of complex analysis

The Boltzmann method has a much broader range of applications than the simple case of directe walks. In its generality, the determination of λ^* is based on the formulation of $Z(\lambda)$ as a Cauchy integral, amenable for a saddle-point analysis.

We could avoid this, as we just had independent geometric variables Nonetheless, it's instructive to see what does the "standard approach" give:

$$
\begin{aligned}
Z_{n,m} &= \sum_{\substack{(c_0,\ldots,c_m) \\ \sum_y c_y = n-m}} \prod_y h_y^{c_y} = \oint \frac{\mathrm{d}z}{2\pi i\, z} \sum_{(c_0,\ldots,c_m)} z^{-(n-m)+\sum_y c_y} \prod_y h_y^{c_y} \\
&= \oint \frac{\mathrm{d}z}{2\pi i\, z} z^{-(n-m)} \prod_{y=0}^{m} \sum_{c_y} (z h_y)^{c_y} \\
&= \oint \frac{\mathrm{d}z}{2\pi i\, z} \exp\left[-(n-m) \ln z - \sum_{y=0}^{m} \ln\left(1 - z h_y\right) \right]
\end{aligned}
$$

The role of complex analysis

$$Z_{n,m} = \oint \frac{\mathrm{d}z}{2\pi i\, z} \exp\Big[-(n-m)\ln z - m \left\langle \ln\left(1 - z h_y\right)\right\rangle_{0 \le y \le m} \Big]$$
$$=: \oint \frac{\mathrm{d}z}{2\pi i\, z} \exp\left[m S_\alpha(z)\right]$$

Saddle point equation: $\frac{\mathrm{d}}{\mathrm{d}z} S_\alpha(z) = 0.$

Remark: z^* satisfies the same equation as the Lagrange multiplier e^{λ^*}
This is no accident, and is a well-known 'duality' between Cauchy integrals and Lagrange multipliers, occurring in combinatorial constructions with positive coefficients.

Note that, as a corollary, we get $\lambda^* \in \mathbb{R} \Rightarrow z^* \in \mathbb{R}^+$

Recursive and Boltzmann methods, in summary

For problems of memoryless strings (i.e. directed random walks) the generating functions satisfy simple linear recursion relations, with positive coefficients, amenable to exact sampling through the Recursive Method, provided that $p_{n,m}$ can be evaluated efficiently. Complexity $T(n) \lesssim n\tau_n$ if evaluating $p_{n,m}$ costs τ_n.

This is the case, e.g., when $Z_{n,m}$ has a factorised formula, so that $p_{n,m}$ is a rational function of constant size.

For problems of "almost-independent" random variables, subject to a finite number k of global linear constraints (e.g. $\sum_y c_y = n - m$), the Boltzmann strategy, of passing to the grand-canonical ensemble with suitably-tuned Lagrange multipliers λ^*, would work with complexity $T \sim n^{1+\frac{k}{2}}$ (neglecting some solvable details).

Finding λ^* is often based on a saddle-point equation.

Part 2

The SPQR Method
(i.e. *Saddle-Point–Query Recursive Method*)

The SPQR Method

We want to take the good of both Boltzmann and Recursive Methods, and devise a new "SPQR Method"

The name stands for "Saddle-Point–Query Recursive Method". That says it all.

When $Z_{n,m}$ does not have a factorised formula, it may still have a saddle-point formulation, so that $p_{n,m}$ is a ratio of almost identical saddle-point integrals

$$p_{n,m} = \frac{\oint \frac{\mathrm{d}z}{2\pi i\, z} A_1(z;\alpha) \exp\left[nB(z;\alpha)\right]}{\oint \frac{\mathrm{d}z}{2\pi i\, z} A_2(z;\alpha) \exp\left[nB(z;\alpha)\right]}$$

As well known, if the dominant saddle point structure is simple,

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1+\mathcal{O}(n^{-1})\right), \qquad \text{where } z^* \text{ solves } \frac{\mathrm{d}}{\mathrm{d}z} B(z;\alpha) = 0.$$

Roadmap to an algorithm

What do we need to turn the SPQR idea into an algorithm?

Roadmap to an algorithm

What do we need to turn the SPQR idea into an algorithm?

- Transform the saddle-point integral perturbative series into a hierarchy of rigorous bounds.

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots\right)$$

$$p_{n,m} \lesseqgtr \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots + \frac{1}{n^k} G_k^{\pm}(z^*)\right)$$

Roadmap to an algorithm

What do we need to turn the SPQR idea into an algorithm?

- Transform the saddle-point integral perturbative series into a hierarchy of rigorous bounds.
- Re-use the information on (a decent approximation of) $z^*(n,m)$ for evaluating (a d. a. of) $z^*(n-1,m')$, where $m' = m$ or $m-1$. E.g. by one step of Newton algorithm.

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots\right)$$

$$p_{n,m} \lesseqgtr \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots + \frac{1}{n^k} G_k^{\pm}(z^*)\right)$$

Roadmap to an algorithm

What do we need to turn the SPQR idea into an algorithm?

- Transform the saddle-point integral perturbative series into a hierarchy of rigorous bounds.
- Re-use the information on (a decent approximation of) $z^*(n,m)$ for evaluating (a d. a. of) $z^*(n-1,m')$, where $m' = m$ or $m-1$. E.g. by one step of Newton algorithm.
- When n is too small (say $n \sim N^{\frac{1}{\gamma}}$) our bounds become large. At that point you finish with another algorithm (say, with complexity $T(n) \sim n^\gamma \sim N$).

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n}G_1(z^*) + \frac{1}{n^2}G_2(z^*) + \cdots\right)$$

$$p_{n,m} \lessgtr \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\left(1 + \frac{1}{n}G_1(z^*) + \frac{1}{n^2}G_2(z^*) + \cdots + \frac{1}{n^k}G_k^{\pm}(z^*)\right)$$

Recursive Method with approx. branching probabilities

This was the Recursive Method when $p_{n,m}$ can be calculated.

Complexity

$$T(n) = \Theta(n) + \sum_{i=1}^{n} \tau_i \lesssim n\, \tau_n$$

```
n' = n; m' = m;
while n' > 0 do
    if Bern_{p_{n',m'}} then
        s_{n'} =↑; m' = m' − 1;
    else
        s_{n'} =→;
    end
end
```

Recursive Method with approx. branching probabilities

This was the Recursive Method when $p_{n,m}$ can be calculated.
Complexity

$$T(n) = \Theta(n) + \sum_{i=1}^{n} \tau_i \lesssim n\,\tau_n$$

$n' = n$; $m' = m$;
while $n' > 0$ **do**
  if $\mathrm{Bern}_{p_{n',m'}}$ **then**
    $s_{n'} = \uparrow$; $m' = m' - 1$;
  else
    $s_{n'} = \rightarrow$;
  end
end

The SPQR Method variant enters in "**if** $\mathrm{Bern}_{p_{n',m'}}$"...

We have a "hierarchy of bounds" $p^{\pm,k}_{n,m}$, with $p^{+,k}_{n,m} - p^{-,k}_{n,m} = \mathcal{O}(n^{-k})$

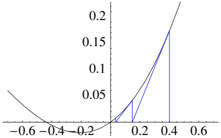
Calculating $p^{\pm,k}_{n,m}$ costs $\tau_n^{(k)}$

"New" complexity:

$$n\tau_n \longrightarrow n \sum_k n^{-k+1} \tau_n^{(k)} = n\tau_n^{(1)} + \tau_n^{(2)} + n^{-1}\tau_n^{(3)} + \cdots$$

The Newton step

Unless things go bad,
a Newton step doubles
the number of
“good” digits



$$f(z) = z + a_2 z^2 + a_3 z^3 + \cdots$$

$$\begin{aligned}\mathcal{N}(z) &= z - \frac{f(z)}{f'(z)} \\ &= z^2\big(a_2 + 2(a_3 - a_2^2)z + \cdots\big)\end{aligned}$$

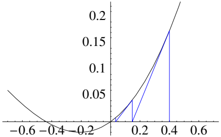
Again, you need to transform a perturbative expansion
into a rigorous (bilateral) bound

In general, you revert to *Blum, Cucker, Shub and Smale: Complexity and Real Computation* (ch. 8 and 9)

For a specific function, you might have a shortcut…

The Newton step

Unless things go bad, a Newton step doubles the number of "good" digits



$$f(z) = z + a_2 z^2 + a_3 z^3 + \cdots$$

$$\begin{aligned}\mathcal{N}(z) &= z - \tfrac{f(z)}{f'(z)} \\ &= z^2\big(a_2 + 2(a_3 - a_2^2)z + \cdots\big)\end{aligned}$$

Again, you need to transform a perturbative expansion into a rigorous (bilateral) bound

In general, you revert to *Blum, Cucker, Shub and Smale: Complexity and Real Computation* (ch. 8 and 9)

For a specific function, you might have a shortcut...

Quick certifications of Newton Method

Being "near enough" to a zero the precision doubles at each step. The theorems in [BCSS] give complicated sufficient conditions.

If your function $f(z)$ is gentle enough, this property may hold globally, for all problems $f(z) = h$ and starting position z_0: let $f(z_\infty) = h$ and $z_1 = \mathcal{N}_h(z_0) = z_0 - (f(z_0) - h)/f'(z_0)$, for all pairs (z_0, h) you may have $|z_\infty - z_1| \leq |z_\infty - z_0|^2$.

Say a function $y(x)$ is fast if both $y(x)$ and $x(y)$ can be computed efficiently. E.g., $y(x) = \frac{ax+b}{cx+d}$ (Möbius transformation).

Normally, you are not so lucky, and f is not gentle...
But you have the right of playing with monotone transformations $f \to g = \phi_1 \circ f \circ \phi_2$, with both ϕ_α's fast, so that the resulting function g is gentle.

The saddle point equation for Stirling 2nd kind is gentle

The saddle point equation for our $\left\{ {n \atop \alpha n} \right\}$ is $\alpha = \frac{1-e^{-z}}{z} =: f(z)$

$f(z) : \mathbb{R}^+ \to (0,1]$. It is not gentle (and has not compact support).
Choose $f \to g = \phi_1 \circ f \circ \phi_2$ with $\phi_1(z) = \frac{3z}{4-z}$ and $\phi_2(z) = \frac{3z}{2-2z}$
The new function $g : [0,1] \to [0,1]$ is gentle

$$\frac{z_\infty - z_1}{(z_\infty - z_0)^2}$$

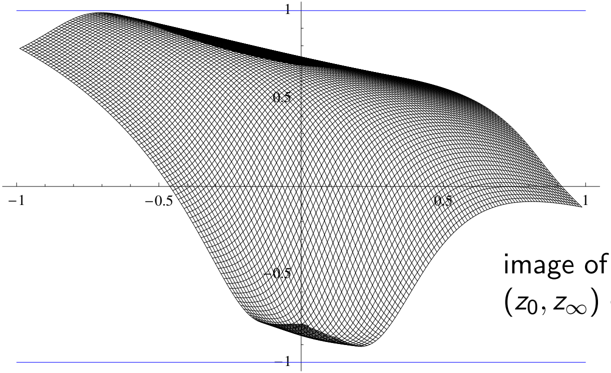


image of
$(z_0, z_\infty) \in [0,1]^2$

A number of preliminary results...

Now we have to produce our claimed
hierarchy of saddle-point rigorous bounds

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\Big(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots\Big)$$

$$p_{n,m} \lesseqgtr \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\Big(1 + \frac{1}{n} G_1(z^*) + \frac{1}{n^2} G_2(z^*) + \cdots + \frac{1}{n^k} G_k^{\pm}(z^*)\Big)$$

For this we need some preliminary definitions and results:

- a notation for propagation of errors in $\mathbb{C}$;
- a notion of "sign decomposition" of a function;
- a result on the formal inversion of $S(x(y)) = y^2$;

A number of preliminary results...

Now we have to produce our claimed
hierarchy of saddle-point rigorous bounds

$$p_{n,m} = \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\Big(1 + \frac{1}{n}G_1(z^*) + \frac{1}{n^2}G_2(z^*) + \cdots\Big)$$
$$p_{n,m} \lessgtr \frac{A_1(z^*;\alpha)}{A_2(z^*;\alpha)}\Big(1 + \frac{1}{n}G_1(z^*) + \frac{1}{n^2}G_2(z^*) + \cdots + \frac{1}{n^k}G_k^{\pm}(z^*)\Big)$$

For this we need some preliminary definitions and results:

- a notation for propagation of errors in $\mathbb{C}$;
- a notion of "sign decomposition" of a function;
- a result on the formal inversion of $S(x(y)) = y^2$;

The $\pm$ notation for propagation of errors in $\mathbb{C}$

For $x, a \in \mathbb{C}$, $b \in \mathbb{R}^+$, let $x = a \pm b$ mean $|x - a| \leq b$.
Let $(a \pm b) \stackrel{\triangleleft}{=} (c \pm d)$ mean $(x = a \pm b) \Rightarrow (x = c \pm d)$.

Nice properties:

$$
\begin{aligned}
(a \pm b) + (c \pm d) &= (a + c) \pm (b + d)\,; \\
c(a \pm b) &= ca \pm |c|\,b\,;
\end{aligned}
$$

and, when $f(z)$ is analytic,

$$
f(a \pm b) \stackrel{\triangleleft}{=} f(a) \pm b' \qquad b' = \max_\theta \left| f(a + be^{i\theta}) - f(a) \right|
$$

Among the corollaries of this fact, we have for $b \in \mathbb{R}^+$

$$
\begin{aligned}
\exp(\pm b) &\stackrel{\triangleleft}{=} 1 \pm (e^b - 1) && b \in \mathbb{R}^+ \\
\frac{a \pm b}{c \pm d} &\stackrel{\triangleleft}{=} \frac{1}{c^2 - d^2} \big((ac + bd) \pm (ad + bc) \big) && \begin{array}{c} a > b \\ c > d \end{array}
\end{aligned}
$$

The $\pm\!\!\!\!\sim$ notation for propagation of errors in $\mathbb{C}$

Other special case: $P(z) = p_1 z + p_2 z^2 + \ldots + p_d z^d$ on D_η

$$e^{P(z)} = e^{p_1 z} e^{\,\pm\!\!\!\!\sim(|p_2 z^2| + \cdots + |p_d z^d|)} \overset{\trianglelefteq}{=} e^{p_1 z} \left(1 \,\pm\!\!\!\!\sim |z|^2 \frac{e^{|p_2|\eta^2 + \cdots + |p_d|\eta^d} - 1}{\eta^2} \right)$$

We need a similar result for generic functions.
Let $f(z) = f_0 + f_1 z + f_2 z^2 + \cdots$, analytic and with radius of conv. ρ
Call $f^{[k]}(z) = f_0 + f_1 z + f_2 z^2 + \cdots + f_{k-1} z^{k-1}$.
For $\eta < \rho$, we want $r(\eta)$ such that $f(z) \in f^{[k]}(z) \,\pm\!\!\!\!\sim r(\eta)|z|^k$.

If, for $j \geq k$, all coefficients f_j are real positive, then

$$f(z) = f^{[k]}(z) \,\pm\!\!\!\!\sim |z|^k \frac{f(\eta) - f^{[k]}(\eta)}{\eta^k} \,.$$

Analogous tricks if f_j's are all negative, or have alternating sign.

Sign-decomposition of a function

Let $\mathcal{F}_{\sigma_e,\sigma_o}$ the space of analytic functions $f(z) = \sum_j f_j z^j$ with $\mathrm{sign}(f_{2j}) = \sigma_e$ and $\mathrm{sign}(f_{2j+1}) = \sigma_o$.

We just said that, for $f \in \mathcal{F}_{\sigma_e,\sigma_o}$, we can find good bounds $f(z) = f^{[k]}(z) \pm r(\eta)|z|^k$ (in the disk of radius η).

You want to do that for your functions $A(z)$ and $B(z)$ in the saddle-point integral $I = \oint \frac{\mathrm{d}z}{2\pi i z} A(z) \exp(nB(z))$...
...but most of functions are not this way!

Still, you can decompose
$f(z) = f_{++}(z) + f_{+-}(z) + f_{-+}(z) + f_{--}(z)$,
with $f_{\sigma_e,\sigma_o} \in \mathcal{F}_{\sigma_e,\sigma_o}$, and all "computable efficiently".

Sign-decomposition of a function

In our example for Stirling's 2nd kind $\left\{ {n \atop m} \right\}$, for $\frac{n+m+1}{n} = \frac{\zeta}{1-e^{-\zeta}}$,

$$B_\zeta(x) = (1 - e^{-\zeta}) \ln(e^{\zeta+x} - 1) - \zeta \ln(\zeta + x)$$

for all $\zeta \in \mathbb{R}^+$ we find

- $-\zeta \ln(\zeta + x) \in \mathcal{F}_{-+}$ (obvious)
- $(1 - e^{-\zeta}) \ln(e^{\zeta+x} - 1) \in \mathcal{F}_{+-}$ (smart)

Call $y = e^{-\zeta}$. Write $(1 - e^{-\zeta}) \ln(e^{\zeta+x} - 1) = a(\zeta) + b(\zeta) \ln \frac{e^x - y}{1-y}$ (with $a(\zeta), b(\zeta) > 0$). Then

$$\ln \frac{e^x - y}{1 - y} = \frac{x}{1 - y} + y \sum_{n,k} \frac{(-1)^{n-1}}{n!(1 - y)^n} T_{n,k} x^n y^k$$

where the coefficients $T_{n,k}$ are the *Eulerian numbers* (number of permutations of $n + 1$ objects with k rises)
`http://oeis.org/A008292` $\Rightarrow T_{n,k} \in \mathbb{N}$.

Formal inversion of $S(x(y)) = y^2$

We want to "rectify" the function $S(z)$ of the Cauchy integral $\oint A(z) \exp(nS(z)) \mathrm{d}z$, at the (simple) saddle point $z = z^*$, into an exact parabola, through a change of variables.

At this aim we want to solve the equation $S(x(y)) = y^2$, given that $x(y) = y + a_2 y^2 + a_3 y^3 + \ldots$ and $S(x) = x^2 + b_3 x^3 + b_4 x^4 + \ldots$ ($x(y)$ and $S(x)$ are formal power series)

The problem exists in two versions: ① find a, given b; ② find b, given a.
We need to solve, for all $k \geq 3$, $C_k(a, b) := [y^k] S(x(y)) = 0$
Remark: $C_k(a, b) = 2a_{k-1} + b_k + C'_k(\{a_h\}_{h \leq k-2}, \{b_h\}_{h \leq k-1})$,
Thus the system of equations is triangular and linear,
for both versions of the problem.
The solution is unique, and is find quite efficiently.

The first few terms for $b(a)$ read

$$
\begin{aligned}
b_3 &= -2\,a_2 \\
b_4 &= 5\,a_2^2 - 2\,a_3 \\
b_5 &= -14\,a_2^3 + 12\,a_2 a_3 - 2\,a_4 \\
b_6 &= 42\,a_2^4 - 56\,a_2^2 a_3 + 7\,a_3^2 + 14\,a_2 a_4 - 2\,a_5 \\
b_7 &= -132\,a_2^5 + 240\,a_2^3 a_3 - 72\,a_2 a_3^2 - 72\,a_2^2 a_4 \\
&\quad + 16\,a_3 a_4 + 16\,a_2 a_5 - 2\,a_6 \\
b_8 &= 429\,a_2^6 - 990\,a_2^4 a_3 + 495\,a_2^2 a_3^2 - 30\,a_3^3 + 330\,a_2^3 a_4 \\
&\quad - 180\,a_2 a_3 a_4 + 9\,a_4^2 - 90\,a_2^2 a_5 + 18\,a_3 a_5 + 18\,a_2 a_6 - 2\,a_7 \\
&\quad \vdots
\end{aligned}
$$

Formal inversion of $S(x(y)) = y^2$

For $a(b)$, we better visualise even and odd coefficients separately

$$2a_2 = -b_3\,; \qquad 2a_4 = -2\,b_3^3 + 3\,b_3 b_4 - b_5\,;$$
$$2a_6 = -7\,b_3^5 + 20\,b_3^3 b_4 - 10\,b_3 b_4^2 - 10\,b_3^2 b_5 + 4\,b_4 b_5 + 4\,b_3 b_6 - b_7\,;$$

and

$$\begin{aligned}
2^3 a_3 &= 5\,b_3^2 - 4\,b_4 \\
2^7 a_5 &= 231\,b_3^4 - 504\,b_3^2 b_4 + 112\,b_4^2 + 224\,b_3 b_5 - 64\,b_6\,; \\
2^{11} a_7 &= 14586\,b_3^6 - 51480\,b_3^4 b_4 + 41184\,b_3^2 b_4^2 - 4224\,b_4^3 + 27456\,b_3^3 b_5 \\
&\quad - 25344\,b_3 b_4 b_5 + 2304\,b_5^2 - 12672\,b_3^2 b_6 + 4608\,b_4 b_6 \\
&\quad + 4608\,b_3 b_7 - 1024\,b_8\,.
\end{aligned}$$

In our algorithm we only need the solution $a(b)$ up to a given order. Thus this is a fixed $\mathcal{O}(1)$ preprocessing.

The hierarchy of saddle-point bounds

How do we produce our bounds,
in the case of Stirling numbers of second kind?

- Recall the decomposition $B(x;\zeta) = B_{-+}(x;\zeta) + B_{+-}(x;\zeta)$;
- For each summand, near to $x = x^*$ and within a radius η, write $B_{\sigma\tau}(x;\zeta) = B^{[k]}_{\sigma\tau}(x;\zeta) \overset{+}{\sim} r(\eta)|x - x^*|^k$;
- Use the solution to $S(x(y)) = y^2$ to bring higher orders out of the exponential;
- Perform the corresponding integrals, which are moments of the Gaussian (deal with the tail terms $|x - x^*| > \eta$ as usual);
- The moments associated to $\overset{+}{\sim} |x - x^*|^k$ factors cause the gap between lower and upper bound. They get a factor $m^{-\frac{k-2}{2}}$;
- Use the formula for $\frac{a \overset{+}{\sim} b}{c \overset{+}{\sim} d}$ to finally get $p_{n,m} = \frac{\oint A_1 \cdots}{\oint A_2 \cdots}$.

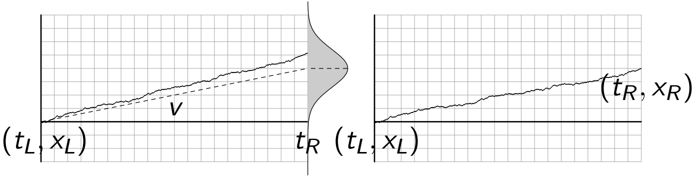

BRADPITT
ACHILLES

Part 3

The Zeno–Boltzmann Method
(a.k.a. *Achilles and the tortoise*)

Boltzmann Method: a metaphore from the continuum

Brownian Motion $\mathrm{BM}(t_L, x_L; t_R; v)$ | Brownian Bridge $\mathrm{BB}(t_L, x_L; t_R, x_R)$

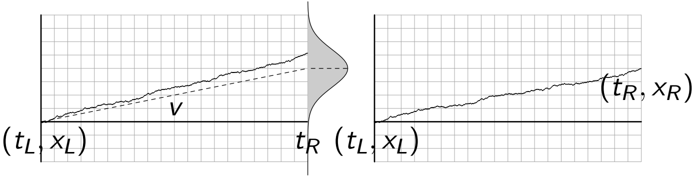


Let's play the following *exercice de style*:
Suppose you have a perfect sampler for $\mathrm{BM}(t_L, x_L; t_R; v)$.
How can you produce a sampler for $\mathrm{BB}(0, 0; t, x)$?

Claim: the original idea of Boltzmann samplers is to sample $\mathrm{BM}(0, 0; t; x/t)$, and reject if you don't arrive at the good point.
Acceptance rate in the discrete: $\sim n^{-\frac{1}{2}}$.
Goes to zero in the continuum limit!

Boltzmann Method: a metaphore from the continuum

Brownian Motion $\mathrm{BM}(t_L, x_L; t_R; v)$ | Brownian Bridge $\mathrm{BB}(t_L, x_L; t_R, x_R)$



Let's play the following *exercice de style*:

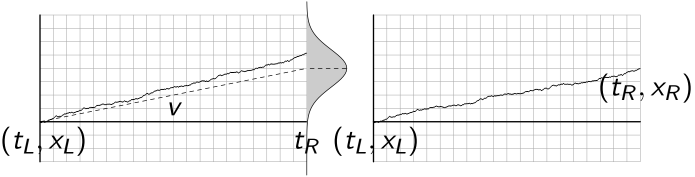
Suppose you have a perfect sampler for $\mathrm{BM}(t_L, x_L; t_R; v)$.
How can you produce a sampler for $\mathrm{BB}(0, 0; t, x)$?

the new strategy is to sample $\mathrm{BM}(0, 0; \alpha t; x/t)$ ($\alpha \in (0, 1)$, e.g. $\alpha = 1/2$), and reject with a probability related to BM and BB marginals. Then, restart with the remaining interval.

Claim: acceptance rate is $\sqrt{1-\alpha}$. Finite, both in the discrete and continuum, and in fact arbitrarily near to optimal ($\alpha \to 0$).

Boltzmann Method: a metaphore from the continuum

Brownian Motion $\mathrm{BM}(t_L, x_L; t_R; v)$ | Brownian Bridge $\mathrm{BB}(t_L, x_L; t_R, x_R)$



$$\left(E(x,s) := \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{x^2}{2s}\right);\ \text{in BB, } v := \frac{x_R - x_L}{t_R - t_L};\ s_i^j := s_j - s_i;\ \right)$$

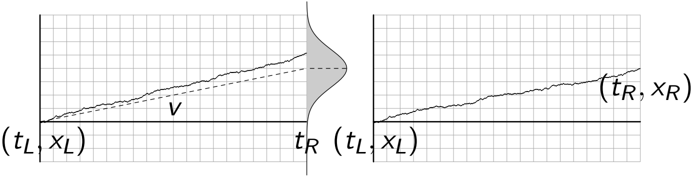
$$\mathbb{P}_{\mathrm{BM}}(\{t_i, x_i\}_{i=1}^k)\,\mathrm{d}\mathbf{x} = \mathrm{d}\mathbf{x} \prod_{i=1}^{k} E(x_{i-1}^i - v t_{i-1}^i, t_{i-1}^i)$$

$$\mathbb{P}_{\mathrm{BB}}(\{t_i, x_i\}_{i=1}^k)\mathrm{d}\mathbf{x} = \mathrm{d}\mathbf{x} E(x_L^R - v t_L^R, t_L^R)^{-1} \prod_{i=1}^{k+1} E(x_{i-1}^i - v t_{i-1}^i, t_{i-1}^i)$$

$$= \mathbb{P}_{\mathrm{BM}}(\{t_i, x_i\}_{i=1}^k)\,\mathrm{d}\mathbf{x}\, \frac{E(x_k^R - v t_k^R, t_k^R)}{E(x_L^R - v t_L^R, t_L^R)}$$

Boltzmann Method: a metaphore from the continuum

Brownian Motion $\mathrm{BM}(t_L, x_L; t_R; v)$ | Brownian Bridge $\mathrm{BB}(t_L, x_L; t_R, x_R)$



$$\mathrm{Acc.Rate}_\alpha(x) = \frac{\mathbb{P}_{\mathrm{BB}}(\alpha t, x)}{\mathbb{P}_{\mathrm{BM}}(\alpha t, x)} = C\, e^{-\frac{(x-\mathbb{E}(x))^2}{2(1-\alpha)t}}$$

Can choose $C = 1$ (instead of Boltzmann's $C \sim 1/\sqrt{n}$)

$$\mathbb{E}(\mathrm{Acc.Rate}_\alpha) = \int \mathrm{d}x \frac{\mathbb{P}_{\mathrm{BB}}(\alpha t, x)}{\mathbb{P}_{\mathrm{BM}}(\alpha t, x)}\, e^{-\frac{(x-\mathbb{E}(x))^2}{2(1-\alpha)t}}$$

$$= \int \mathrm{d}x \frac{1}{\sqrt{2\pi\alpha t}}\, e^{-\frac{(x-\mathbb{E}(x))^2}{2\alpha t}}\; e^{-\frac{(x-\mathbb{E}(x))^2}{2(1-\alpha)t}} = \sqrt{1-\alpha}$$

Directed walks under the new paradigm

Let us come back to our problem on the lattice:
$\omega : (0,0) \to (n-m, m)$, in the case $h_{x,y} = h_y$ and $v_{x,y} = 1$.
We already determined

$$\omega = (\underbrace{\to \cdots \to}_{c_0} \uparrow \underbrace{\to \cdots \to}_{c_1} \uparrow \dots \uparrow \underbrace{\to \cdots \to}_{c_m})$$

$$\mu_\lambda^{\mathrm{BM}}(c_0, c_1, \dots, c_m) = \Big(\prod_{y=0}^{m} h_y^{c_y} \Big) e^{\lambda \sum_y c_y}$$

$$\mu_n^{\mathrm{BB}}(c_0, c_1, \dots, c_m) = \Big(\prod_{y=0}^{m} h_y^{c_y} \Big) \chi\big[n = m + \sum_y c_y \big]$$

$$\frac{n}{m} = \left\langle \frac{1}{1 - e^\lambda h_y} \right\rangle_{0 \le y \le m}$$

Directed walks under the new paradigm

How would you make an algorithm?

- select $Y \subseteq \{0, 1, \ldots, m\}$, $|Y| \sim \alpha m$, through i.i.d. Bern_α.
- Sample the geometric variables $c_y = \mathrm{Geom}_{e^\lambda h_y}$.
- Calculate their sum $n_{\mathrm{samp}}(Y) = \sum_{y \in Y} c_y$.
- Calculate the acceptance rate $\rho_Y(n_{\mathrm{samp}})$.
- If $\mathrm{Rand}_{[0,1]} < \rho_Y(n_{\mathrm{samp}})$, accept the partial configuration and repeat on Y^c, otherwise repeat on $[m]$.

How would you make an algorithm?

- select $Y \subseteq \{0, 1, \dots, m\}$, $|Y| \sim \alpha m$, through i.i.d. Bern_α.
- Sample the geometric variables $c_y = \mathrm{Geom}_{e^\lambda h_y}$.
- Calculate their sum $n_{\mathrm{samp}}(Y) = \sum_{y \in Y} c_y$.
- Calculate the acceptance rate $\rho_Y(n_{\mathrm{samp}})$.
- If $\mathrm{Rand}_{[0,1]} < \rho_Y(n_{\mathrm{samp}})$, accept the partial configuration and repeat on Y^c, otherwise repeat on $[m]$.

$\rho_Y(n)$ is approximatively a Gaussian with 'good' properties, as we know from the analogy with BM/BB process. So $\mathbb{E}(\rho_Y) = \mathcal{O}(1)$.

Directed walks under the new paradigm

How would you make an algorithm?

- select $Y \subseteq \{0, 1, \ldots, m\}$, $|Y| \sim \alpha m$, through i.i.d. Bern_α.
- Sample the geometric variables $c_y = \mathrm{Geom}_{e^{\lambda} h_y}$.
- Calculate their sum $n_{\mathrm{samp}}(Y) = \sum_{y \in Y} c_y$.
- Calculate the acceptance rate $\rho_Y(n_{\mathrm{samp}})$.
- If $\mathrm{Rand}_{[0,1]} < \rho_Y(n_{\mathrm{samp}})$, accept the partial configuration and repeat on Y^c, otherwise repeat on $[m]$.

$\rho_Y(n)$ is approximatively a Gaussian with 'good' properties, as we know from the analogy with BM/BB process. So $\mathbb{E}(\rho_Y) = \mathcal{O}(1)$.

But you don't know its value analytically
(just as in the Recursive Method "bad cases")

Directed walks under the new paradigm

How would you make an algorithm?

- select $Y \subseteq \{0, 1, \dots, m\}$, $|Y| \sim \alpha m$, through i.i.d. Bern_α.
- Sample the geometric variables $c_y = \mathrm{Geom}_{e^{\lambda} h_y}$.
- Calculate their sum $n_{\mathrm{samp}}(Y) = \sum_{y \in Y} c_y$.
- Calculate the acceptance rate $\rho_Y(n_{\mathrm{samp}})$.
- If $\mathrm{Rand}_{[0,1]} < \rho_Y(n_{\mathrm{samp}})$, accept the partial configuration and repeat on Y^c, otherwise repeat on $[m]$.

$\rho_Y(n)$ is approximatively a Gaussian with 'good' properties, as we know from the analogy with BM/BB process. So $\mathbb{E}(\rho_Y) = \mathcal{O}(1)$.

But you don't know its value analytically
(just as in the Recursive Method "bad cases")

So you must again use a saddle-point–query!

Directed walks under the new paradigm: complexity

Let us show that optimality can be reached, within a simplified complexity paradigm: you have a unit cost per sampling of a geometric random variable, and a cost s per saddle-point–query.

Optimality would be $T_{\rm opt} = n$

The complexity satisfies

$$T(n) = \min_{\alpha \in [0,1]} \left(\frac{1}{\sqrt{1-\alpha}} \alpha n + T((1-\alpha)n) + s \right)$$

Make the ansatz $T(n) = n + B\sqrt{n}$. Plug in the equation above, take the leading term for $\alpha \ll 1$, and derive $B = \sqrt{8s}$, $\alpha^* = \sqrt{\frac{s}{2n}}$

Thus an asymptotically optimal strategy is to sample the square root of the number of remaining variables at each round.

Directed walks under the new paradigm: complexity

Even in a less restrictive setting, with unit cost per geometric random variable, and a cost $s(n) \sim s\, n^{\gamma}$ per saddle-point–query at size n (with $\gamma < 1$), we still have optimality

We now have

$$T(n) = \min_{\alpha \in [0,1]} \left(\frac{1}{\sqrt{1-\alpha}} \alpha n + T((1-\alpha)n) + s\, n^{\gamma} \right)$$

Make the ansatz $T(n) = n + Bn^{\beta}$. Substitute above, take $\alpha \ll 1 \dots$

$$n + Bn^{\beta} = n + Bn^{\beta} + \min_{\alpha \in [0,1]} \left(\frac{\alpha^2}{2} n - \alpha\, \beta B n^{\beta} + s\, n^{\gamma} \right)$$

…that gives $\alpha^* = \beta B n^{\beta-1}$ and leaves with $s\, n^{\gamma} = \frac{(\beta B)^2}{2} n^{2\beta-1}$ …
…that gives $\beta = \frac{1+\gamma}{2}$, $B = \frac{\sqrt{8s}}{1+\gamma}$ and $\alpha^* = \sqrt{2s}\, n^{\frac{\gamma-1}{2}}$.

Here an asymptotically optimal strategy is to sample a fraction $n^{\frac{\gamma-1}{2}}$ of the remaining variables at each round.