Diagonals and Walks

Alin Bostan\textsuperscript{1}  Louis Dumont\textsuperscript{1}  Bruno Salvy\textsuperscript{2}

\textsuperscript{1}INRIA, SpecFun
\textsuperscript{2}INRIA, AriC

November 4, 2014
Introduction
In combinatorics:
Lattice walks

\( w_{i,j} \): number of walks that end at \((i,j)\)

\( w_{n,n} \): number of walks that end on the diagonal at \((n,n)\)

\[ W(X, Y) = \sum_{i,j \geq 0} w_{i,j} X^i Y^j \]

Diagonal series:
\[ \text{diag} W(T) = \sum_{n \geq 0} w_{n,n} T^n \]

Diagonals also appear in statistical physics, number theory,...
Motivations

In combinatorics:
Lattice walks

$w_{i,j}$: number of walks that end at $(i,j)$

$w_{n,n}$: number of walks that end on the diagonal at $(n,n)$

Generating power series:
$$W(X, Y) = \sum_{i,j \geq 0} w_{i,j} X^i Y^j$$

Diagonal series:
$$\text{diag} W(T) = \sum_{n \geq 0} w_{n,n} T^n$$

Diagonals also appear in statistical physics, number theory,...
Diagonals

\[ F(X, Y) = \frac{1}{1 - X - Y} = \sum_{i,j \geq 0} \binom{i+j}{i} X^i Y^j \]

\[ \text{diag}(F) = \sum_{n \geq 0} \binom{2n}{n} T^n = \frac{1}{\sqrt{1 - 4T}} \]

Abel (≈ 1830): \( \text{Alg} \subset \text{D-Finite} \)

Furstenberg (1967): \( \text{Diag} = \text{Alg} \)
1-dimensional walks

Set of steps of the form \((1, a), \ a \in \mathbb{Z}\)

Example: Dyck Paths

\[ B(T) = \sum_{n \geq 0} b_n T^n, \ b_n : \text{number of bridges of length } n \]

\(B(T)\) can be expressed as a diagonal.
Questions

Question 1
How hard is it to compute a polynomial equation satisfied by the diagonal of a bivariate rational function?

\[ E(T) = \sum_{n \geq 0} e_n T^n, \quad e_n : \text{number of excursions of length } n \]

Question 2
How to compute \( E(T) \mod T^N \) for a given \( N \)?
**Theorem 1 (B., D., S.)**

*Generically:*

- The minimal polynomial equation can be computed in time that is quasi-linear with respect to its size.
- The minimal polynomial equation is exponentially bigger than the initial rational function ($16^d$ vs $d^2$, where $d$ is the degree)

<table>
<thead>
<tr>
<th>$d$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\deg_T P, \deg_Y P$</td>
<td>(2, 2)</td>
<td>(16, 6)</td>
<td>(108, 20)</td>
<td>(640, 70)</td>
</tr>
</tbody>
</table>

**Theorem 2 (B., D., S.)**

$E(T) \mod T^N$ can be computed in $\tilde{O}(N)$ arithmetic operations, with a fairly inexpensive pre-computation.
Diagonals

How to compute a polynomial equation satisfied by the diagonal of a bivariate rational function?
Algebraic equation for the diagonal

Ingredients of the algorithm:

- Partial fraction decomposition: the diagonal is a sum of residues of a rational function
- Rothstein-Trager resultant (1976): algebraic equation that cancels all residues $\alpha_1(T), \alpha_2(T), \ldots, \alpha_d(T)$
- Main difficulty: if $P = \prod_{i=1}^{d} (Y - \alpha_i)$ is known, efficiently compute the polynomial

$$P_k = \prod_{\{i_1, i_2, \ldots, i_k\}} (Y - (\alpha_{i_1} + \alpha_{i_2} + \ldots + \alpha_{i_k}))$$

for a given $k \leq d$
Newton sums

Most important part of our algorithm: a way around the main difficulty using **Newton sums**.

\[ P = \prod_{i=1}^{d} (Y - \alpha_i) \leftrightarrow \mathcal{N}(P) = \sum_{n \geq 0} (\alpha_1^n + \alpha_2^n + \ldots + \alpha_d^n) \frac{Y^n}{n!} \]

- \( P \leftrightarrow \mathcal{N}(P) \) is well-known and effective
- \( \mathcal{N}(P_k) \) can be computed from \( \mathcal{N}(P) \)
- \( P_k \) is then recovered from \( \mathcal{N}(P_k) \)
Walks

How to compute $E(T) \mod T^N$ for a given $N$?
Excursions: naive method

- $w_{n,k}$: number of walks that stay in the upper half plane, of length $n$ and ending at height $k$
- $e_n = w_{n,0}$ (i.e., number of excursions of length $n$)
- $w_{n,k}$ satisfies a linear recurrence relation with constant coefficients:

$$w_{n,k} = \sum_{(1,a) \in S} w_{n-1,k-a},$$

where $S$ is the set of available steps.

- Algorithm with $O(N^2)$ arithmetic operations and no pre-computation
Excursions: more efficient algorithm

Idea

(Banderier, Flajolet, 2002) Use the fact that $E(T)$ is algebraic to find a better recurrence relation.

- recurrence for $e_n$ (1 index) instead of $w_{n,k}$ (2 indices) $\rightarrow O(N)$ operations
- **But** linear complexity at the cost of pre-computations:
  - the algebraic equation, which is exponentially big in $d$ (Bousquet-Mélou, 2008)
  - algebraic eqn $\rightarrow$ differential eqn *(also exponentially big)* (Bostan, Chyzak, Lecerf, Salvy, Schost, 2007)
  - initial conditions of the recurrence *(exponentially many)*

Algorithm with $O(N)$ operations and pre-computation of an exponential size in $d$ equation
Excursions: new algorithm

Ideas

- \( E(T) = \exp \int_0^T \frac{B(t)-1}{t} \, dt \) (Banderier, Flajolet 2002). Recover \( E(T) \) from \( B(T) \) using this formula.
- Diagonals (including \( B(T) \)) satisfy small (polynomial size) differential equations (Bostan, Chen, Chyzak, Li, 2010)).

Method:

- Fast computation of a differential equation for \( B \) using Hermite reduction (loc. cit.)
- Calculate \( B \mod T^N \) with this equation \( O(N) \)
- Apply the formula using Newton iteration \( \tilde{O}(N) \)

Algorithm with \( \tilde{O}(N) \) operations and pre-computation of a polynomial size in differential equation
Conclusion

D-finite

Alg

Diag

Bridges

Excursons

\[ \text{exponential growth of the size} \]