Guessing with Little Data

Manuel Kauers · Institute for Algebra · JKU

Joint work with Christoph Koutschan

What's the next term?

$$1, 1, 2, 5, 14, 42$$

What's the next term?

$$1, 1, 2, 5, 14, 42, 135$$

What's the exact value?

0.571429

What's the exact value?

$$0.571429 \approx \frac{4}{7}$$

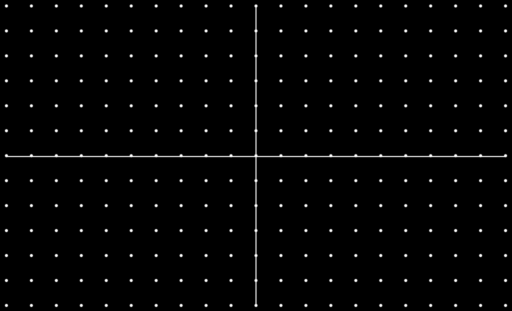
The LLL algorithm can do this trick

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")

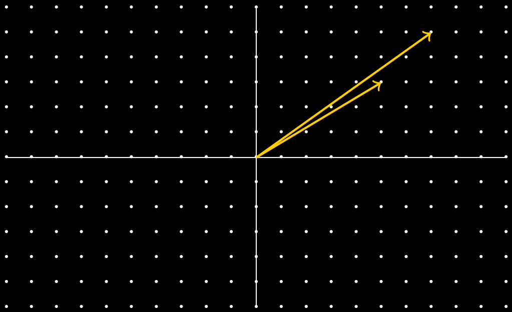
The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")



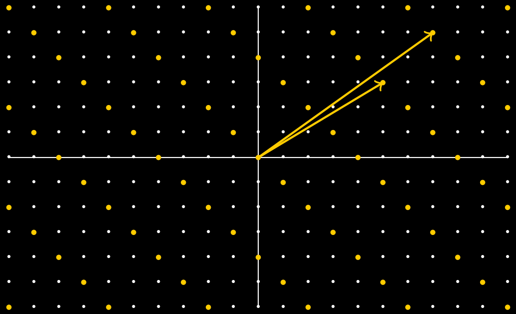
The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")



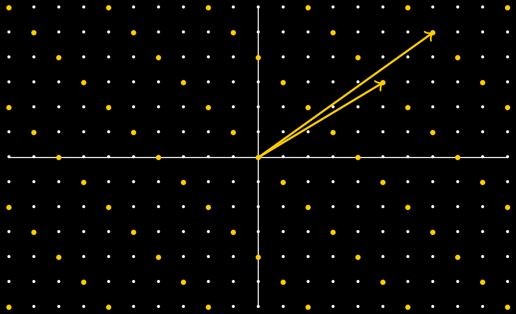
The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")



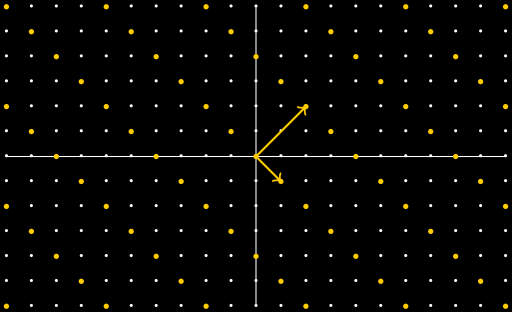
The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors



The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors



The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka “lattice”)
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $|\frac{p}{q} - 0.5714286|$ is small.

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $p - 0.5714286q = \epsilon$.

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $p - 0.5714286q - \epsilon = 0$.

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $10^7 p - 5714286 q - 10^7 \epsilon = 0$.

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $10^7 p - 5714286q - r = 0$.

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $10^7 p - 5714286q - r = 0$.

$$\begin{pmatrix} 4 \\ 7 \\ -2 \end{pmatrix} \in \left\langle \begin{pmatrix} 1 \\ 0 \\ 10000000 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 5714286 \end{pmatrix} \right\rangle \subseteq \mathbb{Z}^3$$

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $10^7 p - 5714286q - r = 0$.

$$\underbrace{\begin{pmatrix} 4 \\ 7 \\ -2 \end{pmatrix}}_{\text{short}} \in \langle \underbrace{\begin{pmatrix} 1 \\ 0 \\ 10000000 \end{pmatrix}}_{\text{long}}, \underbrace{\begin{pmatrix} 0 \\ -1 \\ 5714286 \end{pmatrix}}_{\text{long}} \rangle \subseteq \mathbb{Z}^3$$

The LLL algorithm can do this trick

Input: A basis of a submodule L of $\mathbb{Z}^n$ (aka "lattice")
Output: A basis of L consisting of short vectors

Want: $p, q \in \mathbb{Z}$ such that $10^7 p - 5714286q - r = 0$.

$$\underbrace{\begin{pmatrix} 4 \\ 7 \\ -2 \end{pmatrix}}_{\text{short}} \in \langle \underbrace{\begin{pmatrix} 1 \\ 0 \\ 10000000 \end{pmatrix}}_{\text{long}}, \underbrace{\begin{pmatrix} 0 \\ -1 \\ 5714286 \end{pmatrix}}_{\text{long}} \rangle \subseteq \mathbb{Z}^3$$

An exact rational number can be recovered in this way from a numerical approximation provided that we have enough digits of accuracy.

Guessing Linear Recurrence Equations

Guessing Linear Recurrence Equations

Instantiate the ansatz

$$(c_{10} + c_{11}\, n)a_{n+1} + (c_{00} + c_{01}\, n)a_n \overset{!}{=} 0$$

for $n = 0, \ldots, 3$ to get a linear system for the four undetermined coefficients $c_{00}, c_{01}, c_{10}, c_{11}$.

Guessing Linear Recurrence Equations

Instantiate the ansatz

$$(c_{10} + c_{11}\, n)a_{n+1} + (c_{00} + c_{01}\, n)a_n \overset{!}{=} 0$$

for $n = 0, \ldots, 3$ to get a linear system for the four undetermined coefficients $c_{00}, c_{01}, c_{10}, c_{11}$.

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 2 \\ 2 & 4 & 5 & 10 \\ 5 & 15 & 14 & 42 \end{pmatrix} \begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} = 0$$

Guessing Linear Recurrence Equations

If we have $\#$equations $\geq$ $\#$variables, we do not expect a solution.

Guessing Linear Recurrence Equations

If we have $\#$equations $\geq$ $\#$variables, we do not expect a solution.

If there does exist a solution, it probably means something.

Guessing Linear Recurrence Equations

If we have #equations $\geq$ #variables, we do not expect a solution.

If there does exist a solution, it probably means something.

Assume we know the first N terms of a certain sequence.

Guessing Linear Recurrence Equations

If we have #equations $\geq$ #variables, we do not expect a solution.

If there does exist a solution, it probably means something.

Assume we know the first N terms of a certain sequence.

Assume we want to find a recurrence of order r and degree d.

Guessing Linear Recurrence Equations

If we have #equations $\geq$ #variables, we do not expect a solution.

If there does exist a solution, it probably means something.

Assume we know the first N terms of a certain sequence.

Assume we want to find a recurrence of order r and degree d.

Then there are $(r+1)(d+1)$ variables and $N-r$ equations.

Guessing Linear Recurrence Equations

If we have #equations $\geq$ #variables, we do not expect a solution.

If there does exist a solution, it probably means something.

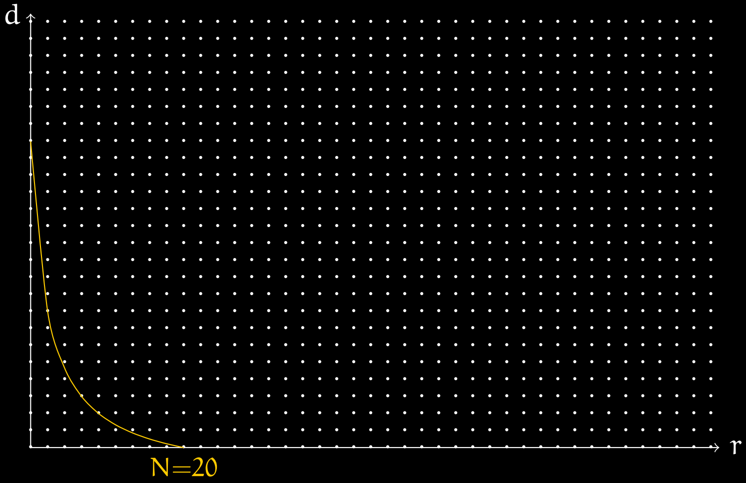
Assume we know the first N terms of a certain sequence.

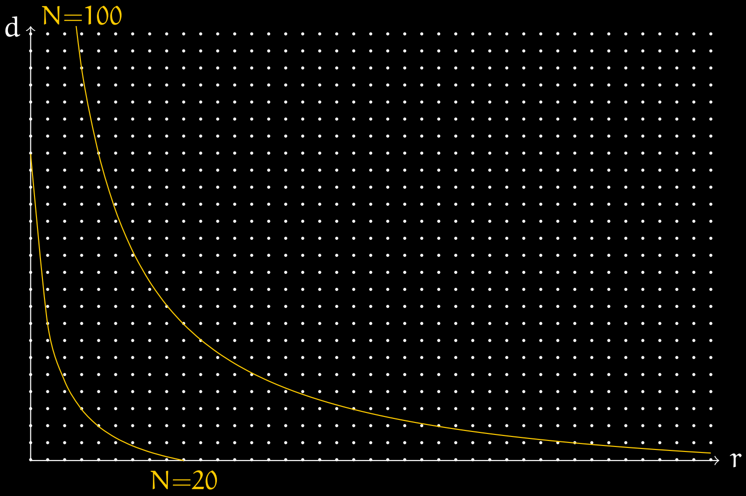
Assume we want to find a recurrence of order r and degree d.

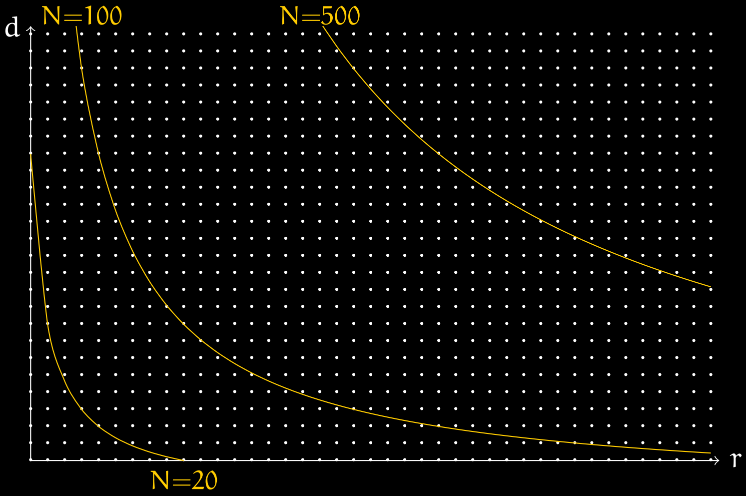
Then there are $(r+1)(d+1)$ variables and $N-r$ equations.

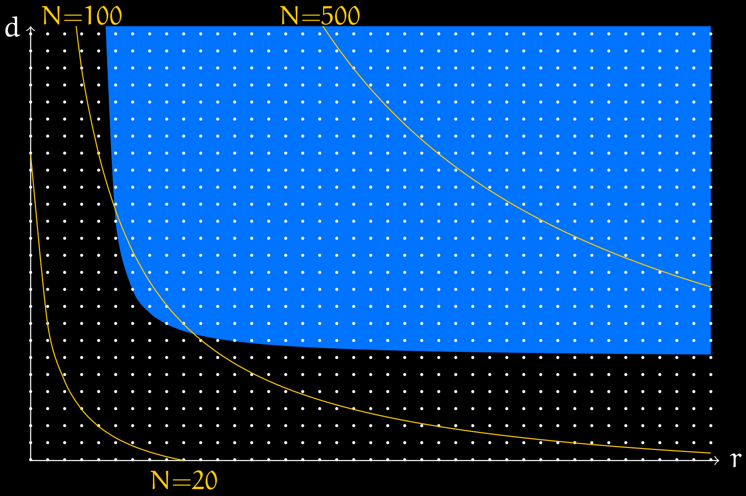
The linear system is overdetermined iff $(r+1)(d+2) \leq N$.


d
r


d
r
N=20


N=100
d
r
N=20


N=100
N=500
d
r
N=20



N=100
N=500
d
r
N=20

What if we don't have enough data?

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers
- There are no irregular singularities

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers
- There are no irregular singularities
- The so-called p-curvature is nilpotent

What if we don’t have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers
- There are no irregular singularities
- The so-called p-curvature is nilpotent
- The coefficients only involve small integers

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers
- There are no irregular singularities
- The so-called p-curvature is nilpotent
- The coefficients only involve small integers

Enforcing the first three leads to nonlinear equations 🙁

What if we don't have enough data?

Correct equations are typically much nicer than generic solutions.

Idea: search specifically for nice equations.

Some common features of nice equations:

- Unrolling the recurrence produces integers
- There are no irregular singularities
- The so-called p-curvature is nilpotent
- The coefficients only involve small integers

Enforcing the first three leads to nonlinear equations 🙁

But the last one is accessible via LLL 🙂

Example

The numbers $a_n = \sum_k \binom{n}{k}^3$ satisfy a recurrence of order 2 and degree 2.

Example

The numbers $a_n = \sum_k \binom{n}{k}^3$ satisfy a recurrence of order 2 and degree 2.

Classical guessing needs at least 12 terms to find this equation.

Example

The numbers $a_n = \sum_k \binom{n}{k}^3$ satisfy a recurrence of order 2 and degree 2.

Classical guessing needs at least 12 terms to find this equation.

Suppose we only know $a_0, \ldots, a_6$.

Example

The numbers $a_n = \sum_k \binom{n}{k}^3$ satisfy a recurrence of order 2 and degree 2.

Classical guessing needs at least 12 terms to find this equation.

Suppose we only know $a_0, \ldots, a_6$.

$$\begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 0 & 10 & 0 & 0 \\ 2 & 2 & 2 & 10 & 10 & 10 & 56 & 56 & 56 \\ 10 & 20 & 40 & 56 & 112 & 224 & 346 & 692 & 1384 \\ 56 & 168 & 504 & 346 & 1038 & 3114 & 2252 & 6756 & 20268 \\ 346 & 1384 & 5536 & 2252 & 9008 & 36032 & 15184 & 60736 & 242944 \end{pmatrix} \begin{pmatrix} c_{00} \\ c_{01} \\ c_{02} \\ c_{10} \\ c_{11} \\ c_{12} \\ c_{20} \\ c_{21} \\ c_{22} \end{pmatrix} = 0$$

Example

A basis for the $\mathbb{Z}$-module of all solutions in $\mathbb{Z}^9$ is

$$\begin{pmatrix} 2 \\ 0 \\ 0 \\ 67069 \\ -52693 \\ -45994 \\ -13414 \\ 13424 \\ 5636 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 231310 \\ -181747 \\ -158629 \\ -46262 \\ 46300 \\ 19438 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 2 \\ 232560 \\ -182728 \\ -159486 \\ -46512 \\ 46550 \\ 19543 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 434140 \\ -341119 \\ -297729 \\ -86828 \\ 86900 \\ 36483 \end{pmatrix}$$

Example

A basis for the $\mathbb{Z}$-module of all solutions in $\mathbb{Z}^9$ is

$$\begin{pmatrix} -8 \\ -16 \\ -8 \\ -16 \\ -21 \\ -7 \\ 4 \\ 4 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} -8 \\ 21 \\ -11 \\ -6 \\ -29 \\ 1 \\ 2 \\ 4 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 12 \\ 12 \\ -34 \\ -46 \\ 27 \\ 21 \\ 8 \\ -6 \\ -2 \end{pmatrix}, \quad \begin{pmatrix} -42 \\ 14 \\ 58 \\ -29 \\ -17 \\ 40 \\ 10 \\ -4 \\ -6 \end{pmatrix}$$

Example

A basis for the $\mathbb{Z}$-module of all solutions in $\mathbb{Z}^9$ is

$$\begin{pmatrix} -8 \\ -16 \\ -8 \\ -16 \\ -21 \\ -7 \\ 4 \\ 4 \\ 1 \end{pmatrix}, \begin{pmatrix} -8 \\ 21 \\ -11 \\ -6 \\ -29 \\ 1 \\ 2 \\ 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 12 \\ 12 \\ -34 \\ -46 \\ 27 \\ 21 \\ 8 \\ -6 \\ -2 \end{pmatrix}, \begin{pmatrix} -42 \\ 14 \\ 58 \\ -29 \\ -17 \\ 40 \\ 10 \\ -4 \\ -6 \end{pmatrix}$$

Example

A basis for the $\mathbb{Z}$-module of all solutions in $\mathbb{Z}^9$ is

$$\begin{pmatrix}-8\\-16\\-8\\-16\\-21\\-7\\4\\4\\1\end{pmatrix}, \begin{pmatrix}-8\\21\\-11\\-6\\-29\\1\\2\\4\\0\end{pmatrix}, \begin{pmatrix}12\\12\\-34\\-46\\27\\21\\8\\-6\\-2\end{pmatrix}, \begin{pmatrix}-42\\14\\58\\-29\\-17\\40\\10\\-4\\-6\end{pmatrix}$$

Indeed,

$$(-8-16n-8n^2)a_n+(-16-21n-7n^2)a_{n+1}+(4+4n+n^2)a_{n+2}=0$$

is a correct recurrence.

How far can we go with this idea?

How far can we go with this idea?

Classically, at least $(r+1)(d+2)$ terms are needed to find a recurrence of order r and degree d.

How far can we go with this idea?

Classically, at least $(r+1)(d+2)$ terms are needed to find a recurrence of order r and degree d.

It seems that with LLL we get along with fewer terms.

How far can we go with this idea?

Classically, at least $(r+1)(d+2)$ terms are needed to find a recurrence of order r and degree d.

It seems that with LLL we get along with fewer terms.

How many terms are needed now?

How far can we go with this idea?

Classically, at least $(r+1)(d+2)$ terms are needed to find a recurrence of order r and degree d.

It seems that with LLL we get along with fewer terms.

How many terms are needed now?

It depends.

How far can we go with this idea?

The linear system must be such that the “true” coefficient vector is significantly shorter than a “generic” solution.

How far can we go with this idea?

The linear system must be such that the "true" coefficient vector is significantly shorter than a "generic" solution.

Theorem (Siegel's lemma). For every $A \in \mathbb{Z}^{n \times m}$ with $m > n$ there exists $x \in \ker_{\mathbb{Z}} A \setminus \{0\}$ with $\|x\|_\infty \leq (m\,\|A\|_\infty)^{m/(m-n)}$.

How far can we go with this idea?

The linear system must be such that the "true" coefficient vector is significantly shorter than a "generic" solution.

Theorem (Siegel's lemma). For every $A \in \mathbb{Z}^{n \times m}$ with $m > n$ there exists $x \in \ker_{\mathbb{Z}} A \setminus \{0\}$ with $\|x\|_\infty \le (m\,\|A\|_\infty)^{m/(m-n)}$.

For generic recurrences of order r and degree d, a somewhat informal estimation suggests that $O(\sqrt{rd})$ terms will suffice.

How far can we go with this idea?

The linear system must be such that the “true” coefficient vector is significantly shorter than a “generic” solution.

Theorem (Siegel's lemma). For every $A \in \mathbb{Z}^{n\times m}$ with $m > n$ there exists $x \in \ker_{\mathbb{Z}} A \setminus \{0\}$ with $\|x\|_\infty \leq (m\,\|A\|_\infty)^{m/(m-n)}$.

For generic recurrences of order r and degree d, a somewhat informal estimation suggests that $O(\sqrt{rd})$ terms will suffice.

Recurrences arising in meaningful applications are not generic. What about these?

How far can we go with this idea?

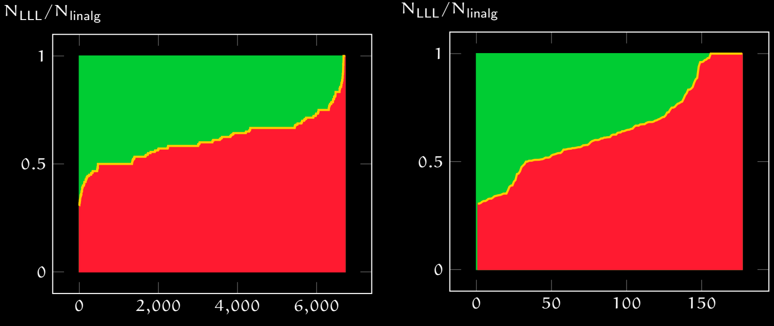
For some 6700 D-finite entries from the OEIS, we compared

- the number N_{linalg} of terms needed by linear algebra guessing.
- the number N_{LLL} of terms needed by LLL-based guessing.

How far can we go with this idea?

For some 6700 D-finite entries from the OEIS, we compared

- the number N_{linalg} of terms needed by linear algebra guessing.
- the number N_{LLL} of terms needed by LLL-based guessing.



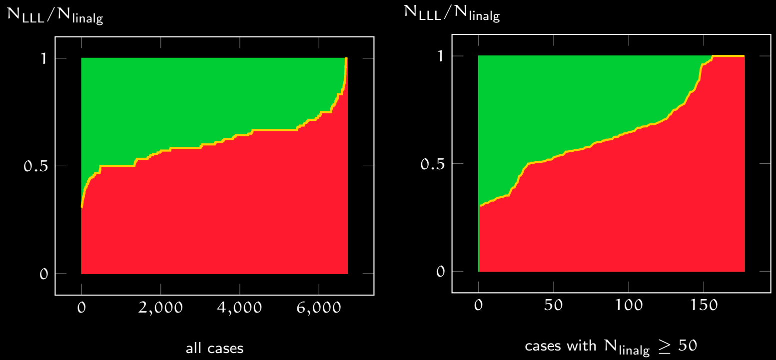
all cases

cases with $N_{\text{linalg}} \geq 50$

How far can we go with this idea?

For some 6700 D-finite entries from the OEIS, we compared

- the number N_{linalg} of terms needed by linear algebra guessing.
- the number N_{LLL} of terms needed by LLL-based guessing.

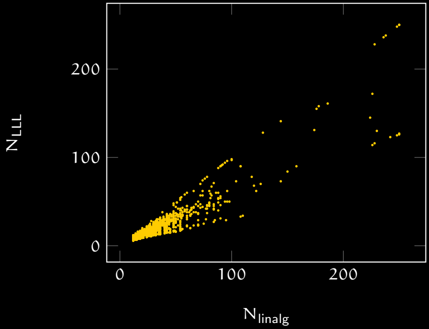


Observation: in some cases we only need a third of the data.

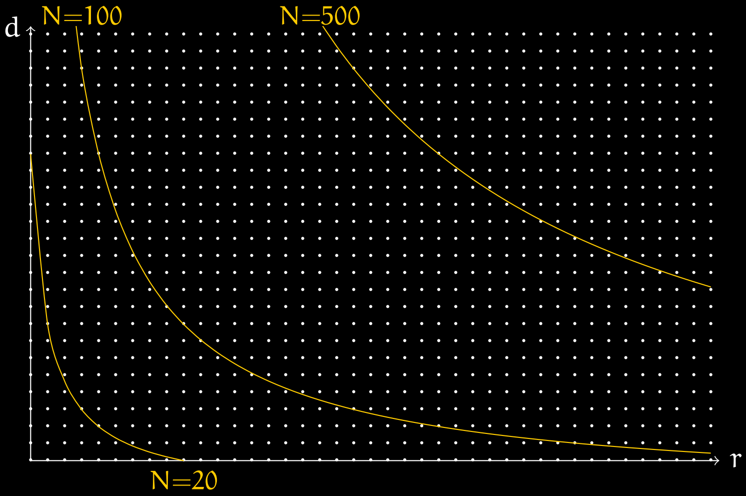
How far can we go with this idea?

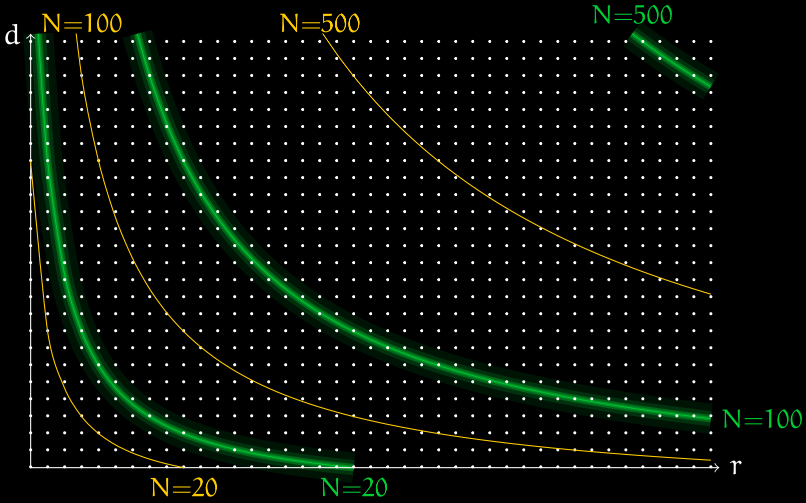
For some 6700 D-finite entries from the OEIS, we compared

- the number N_{linalg} of terms needed by linear algebra guessing.
- the number N_{LLL} of terms needed by LLL-based guessing.



Observation: in some cases we only need a third of the data.


N=100
N=500
d
r
N=20


d
N=100
N=500
N=500
N=100
r
N=20
N=20

Why not just compute more terms?

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

For example, if a_n is the number of permutations in S_n which avoid the pattern 1324, then a_n is unknown for all $n > 50$.

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

For example, if a_n is the number of permutations in S_n which avoid the pattern 1324, then a_n is unknown for all $n > 50$.

"Not even God knows the 1000th term."

—Doron Zeilberger

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

For example, if a_n is the number of permutations in S_n which avoid the pattern 1324, then a_n is unknown for all $n > 50$.

"Not even God knows the 1000th term."

—Doron Zeilberger

Linear algebra doesn't find a recurrence with 50 terms.

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

For example, if a_n is the number of permutations in S_n which avoid the pattern 1324, then a_n is unknown for all $n > 50$.

"Not even God knows the 1000th term."

—Doron Zeilberger

Linear algebra doesn't find a recurrence with 50 terms.

LLL-based guessing doesn't succeed either.

Why not just compute more terms?

The computation of further terms can be prohibitively expensive.

For example, if a_n is the number of permutations in S_n which avoid the pattern 1324, then a_n is unknown for all $n > 50$.

"Not even God knows the 1000th term."
—Doron Zeilberger

Linear algebra doesn't find a recurrence with 50 terms.

LLL-based guessing doesn't succeed either.

Let's try some less prominent examples.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

We checked r, d such that $(r+1)(d+2) \leq 3N$ and accepted a solution if the next 10 terms produced from the given data by the corresponding recurrence are all integers.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

We checked r, d such that $(r+1)(d+2) \leq 3N$ and accepted a solution if the next 10 terms produced from the given data by the corresponding recurrence are all integers.

This led to about 1300 hits.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

We checked r, d such that $(r+1)(d+2) \leq 3N$ and accepted a solution if the next 10 terms produced from the given data by the corresponding recurrence are all integers.

This led to about 1300 hits.

Many of these were obviously wrong.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

We checked r, d such that $(r+1)(d+2) \leq 3N$ and accepted a solution if the next 10 terms produced from the given data by the corresponding recurrence are all integers.

This led to about 1300 hits.

Many of these were obviously wrong.

Many others were obviously correct.

Why not just compute more terms?

There are 350000 sequences in the OEIS. We checked all entries for which between 25 and 150 terms were known and where linear algebra doesn't suffice to find a recurrence.

We checked r, d such that $(r+1)(d+2) \leq 3N$ and accepted a solution if the next 10 terms produced from the given data by the corresponding recurrence are all integers.

This led to about 1300 hits.

Many of these were obviously wrong.

Many others were obviously correct.

In about 20 cases, we may have found something interesting.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

A189281

$$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$$

Only $a_0, \ldots, a_{35}$ are known:

1,
1,
2,
5,
18,
75,
410,
2729,
20906,
181499,
1763490,
18943701,
222822578,
2847624899,
39282739034,
581701775369,
9202313110506,
154873904848803,
2762800622799362,
52071171437696453,
1033855049655584786,
21567640717569135515,
471630531427793184474,
10787660036599729160073,
257590656485400508526570,
6409633590481106885238443,
165928838963556686281573922,
4462073606461933066205164757,
124470791290376112779747519538,
3597058248632667485834774744787,
107559658152025736992729145688602,
3324154021716716493547315823808809,
106067493846954075776733869818571690,
3490771207487802026912252686947947027,
118383998479651470880820236769742970626,
4133478159186775319059453592629838113797

A189281

$$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$$

Only $a_0, \ldots, a_{35}$ are known:

1,	222822578,	25759065648540050852 6570,
1,	2847624899,	640963359048110688523 8443,
2,	39282739034,	1659288389635566862815 73922,
5,	581701775369,	44620736064619330662051 64757,
18,	9202313110506,	124470791290376112779747 519538,
75,	154873904848803,	3597058248632667485834774 744787,
410,	2762800622799362,	10755965815202573699272914 5688602,
2729,	52071171437696453,	332415402171671649354731582 3808809,
20906,	1033855049655584786,	1060674938469540757767338698 18571690,
181499,	21567640717569135515,	3490771207487802026912252686 947947027,
1763490,	471630531427793184474,	11838399847965147088082023676 9742970626,
18943701,	10787660036599729160073,	41334781591867753190594535926 29838113797

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

Example: $a_3 = 5$ because only one $\pi \in S_3$ is excluded.

$$\begin{pmatrix} 123 \\ 123 \end{pmatrix}, \begin{pmatrix} 123 \\ 132 \end{pmatrix}, \begin{pmatrix} 123 \\ 213 \end{pmatrix}, \begin{pmatrix} 123 \\ 231 \end{pmatrix}, \begin{pmatrix} 123 \\ 312 \end{pmatrix}, \begin{pmatrix} 123 \\ 321 \end{pmatrix}.$$

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

Example: $a_3 = 5$ because only one $\pi \in S_3$ is excluded.

$$\begin{pmatrix} 123 \\ 123 \end{pmatrix}, \begin{pmatrix} 123 \\ 132 \end{pmatrix}, \begin{pmatrix} 123 \\ 213 \end{pmatrix}, \begin{pmatrix} 123 \\ 231 \end{pmatrix}, \begin{pmatrix} 123 \\ 312 \end{pmatrix}, \xcancel{\begin{pmatrix} 123 \\ 321 \end{pmatrix}}.$$

A189281

$$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$$

Example: $a_4 = 18$ because six $\pi \in S_4$ are excluded.

$$\begin{pmatrix}1234\\1234\end{pmatrix}, \begin{pmatrix}1234\\1243\end{pmatrix}, \begin{pmatrix}1234\\1324\end{pmatrix}, \begin{pmatrix}1234\\1342\end{pmatrix}, \begin{pmatrix}1234\\1423\end{pmatrix}, \begin{pmatrix}1234\\1432\end{pmatrix},$$

$$\begin{pmatrix}1234\\2134\end{pmatrix}, \begin{pmatrix}1234\\2143\end{pmatrix}, \begin{pmatrix}1234\\2314\end{pmatrix}, \begin{pmatrix}1234\\2341\end{pmatrix}, \begin{pmatrix}1234\\2413\end{pmatrix}, \begin{pmatrix}1234\\2431\end{pmatrix},$$

$$\begin{pmatrix}1234\\3124\end{pmatrix}, \begin{pmatrix}1234\\3142\end{pmatrix}, \begin{pmatrix}1234\\3214\end{pmatrix}, \begin{pmatrix}1234\\3241\end{pmatrix}, \begin{pmatrix}1234\\3412\end{pmatrix}, \begin{pmatrix}1234\\3421\end{pmatrix},$$

$$\begin{pmatrix}1234\\4123\end{pmatrix}, \begin{pmatrix}1234\\4132\end{pmatrix}, \begin{pmatrix}1234\\4213\end{pmatrix}, \begin{pmatrix}1234\\4231\end{pmatrix}, \begin{pmatrix}1234\\4312\end{pmatrix}, \begin{pmatrix}1234\\4321\end{pmatrix}.$$

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

Example: $a_4 = 18$ because six $\pi \in S_4$ are excluded.

$$
\begin{array}{llllll}
\begin{pmatrix} 1234 \\ 1234 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 1243 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 1324 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 1342 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 1423 \end{pmatrix}, & \cancel{\begin{pmatrix} 1234 \\ 1432 \end{pmatrix}}, \\
\begin{pmatrix} 1234 \\ 2134 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 2143 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 2314 \end{pmatrix}, & \cancel{\begin{pmatrix} 1234 \\ 2341 \end{pmatrix}}, & \begin{pmatrix} 1234 \\ 2413 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 2431 \end{pmatrix}, \\
\begin{pmatrix} 1234 \\ 3124 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 3142 \end{pmatrix}, & \cancel{\begin{pmatrix} 1234 \\ 3214 \end{pmatrix}}, & \begin{pmatrix} 1234 \\ 3241 \end{pmatrix}, & \cancel{\begin{pmatrix} 1234 \\ 3412 \end{pmatrix}}, & \begin{pmatrix} 1234 \\ 3421 \end{pmatrix}, \\
\cancel{\begin{pmatrix} 1234 \\ 4123 \end{pmatrix}}, & \begin{pmatrix} 1234 \\ 4132 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 4213 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 4231 \end{pmatrix}, & \begin{pmatrix} 1234 \\ 4312 \end{pmatrix}, & \cancel{\begin{pmatrix} 1234 \\ 4321 \end{pmatrix}}.
\end{array}
$$

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.
- The first vector in the LLL-basis is much shorter than the other basis vectors.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.
- The first vector in the LLL-basis is much shorter than the other basis vectors.
- Even if we just use 29 of the 36 known terms the recurrence can be detected.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.
- The first vector in the LLL-basis is much shorter than the other basis vectors.
- Even if we just use 29 of the 36 known terms the recurrence can be detected.
- The guessed operator has a right factor of order 8 and degree 11.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.
- The first vector in the LLL-basis is much shorter than the other basis vectors.
- Even if we just use 29 of the 36 known terms the recurrence can be detected.
- The guessed operator has a right factor of order 8 and degree 11.
- This right factor has only "nice" asymptotic solutions.

A189281

$a_n := \#\{\, \pi \in S_n : \pi(k+2) - \pi(k) \neq 2 \text{ for all } k \,\}$

We found a trustworthy recurrence of order 10 and degree 6.

- At least the next 10000 sequence terms predicted by the guessed recurrence are integers.
- The first vector in the LLL-basis is much shorter than the other basis vectors.
- Even if we just use 29 of the 36 known terms the recurrence can be detected.
- The guessed operator has a right factor of order 8 and degree 11.
- This right factor has only "nice" asymptotic solutions.
- We computed $a_{36}, a_{37}, a_{38}, a_{39}$ (!) and found that these terms were correctly predicted by the guessed recurrence.

A189281

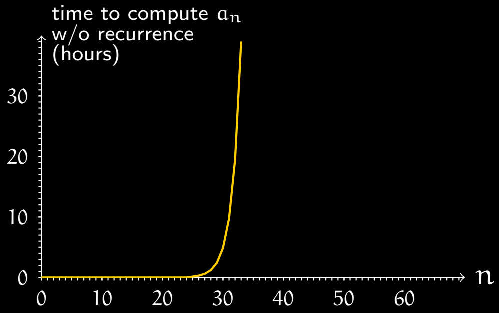
With the recurrence, it's easy to compute more terms of the sequence.

A189281

With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.

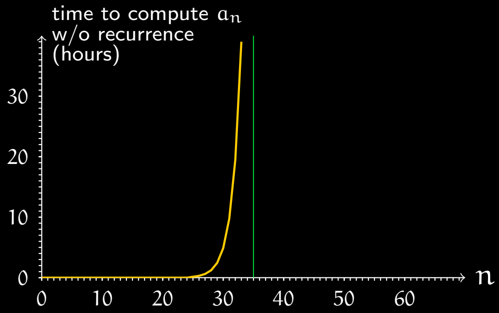
A189281

With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.



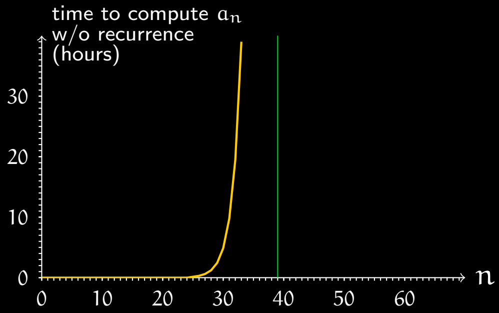
A189281

With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.



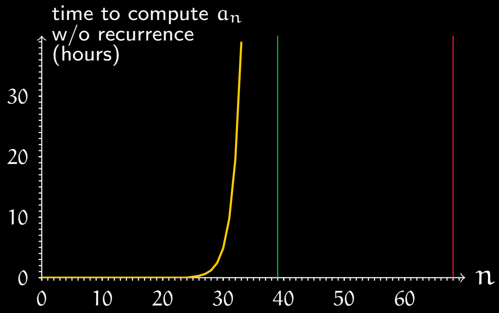
A189281

With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.



A189281

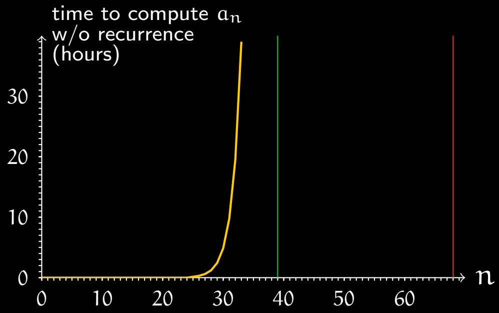
With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.



The classical approach would need 68 terms to find the recurrence.

A189281

With the recurrence, it's easy to compute more terms of the sequence. Without the recurrence, it's much more difficult.



The classical approach would need 68 terms to find the recurrence.

Their computation would take $1.5 \cdot 10^8$ years.

Conclusion

Conclusion

- Guessing is a classical tool in experimental mathematics

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations
- We propose a way to reduce the amount of data needed

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations
- We propose a way to reduce the amount of data needed
- The idea is to use features that linear algebra doesn't see

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations
- We propose a way to reduce the amount of data needed
- The idea is to use features that linear algebra doesn't see
- We search for equations with short coefficients using LLL

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations
- We propose a way to reduce the amount of data needed
- The idea is to use features that linear algebra doesn't see
- We search for equations with short coefficients using LLL
- It seems that this idea works at least on some examples

Conclusion

- Guessing is a classical tool in experimental mathematics
- The more terms you have, the better it works
- Sometimes it is hard to generate much data
- With too few terms, classical guessers may overlook equations
- We propose a way to reduce the amount of data needed
- The idea is to use features that linear algebra doesn't see
- We search for equations with short coefficients using LLL
- It seems that this idea works at least on some examples
- Are there other features that can be used efficiently?