Who wants (a few) “guessing” data points for free?

Manuel Kauers
Thibaut Verron

Institute for Algebra, Johannes Kepler University, Linz, Austria

SFB Statusseminar, Strobl, 05 December 2018

Context: the “Guess & Prove” technique

Problem: given the definition of a sequence $(u_n)_{n\in\mathbb{N}}$, decide whether...

- ... the power series $\sum_{i=0}^{\infty} u_n t^n$ is algebraic?
- ... the sequence $(u_n)_{n\in\mathbb{N}}$ satisfies a D-finite recurrence formula?
- ... the power series $\sum_{i=0}^{\infty} u_n t^n$ satisfies a D-finite differential equation?

The “Guess & Prove” approach:

1. Compute a lot of initial terms $(u_0, \ldots, u_N)$, $N \in \mathbb{N}$
2. Guess a polynomial/recurrence/differential relation satisfied by this data
3. Prove that this relation is satisfied by the sequence as a whole

Guessing equations

The guessing machinery:

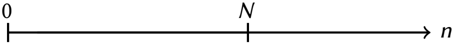
- Several algorithms: linear algebra, Hermite-Padé approximation
- We want to build and solve an overdetermined system to keep bad solutions away
- The size of the equations that we can guess depends on:
  - the type of equation that we want
  - how much data we have

Key question: if I want to guess an equation of a given type, with a given order and degree...

How many data points do I need?

Number of data points required for guessing an algebraic relation

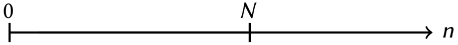
$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^{d}\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^{d}\right) F(t)^{r} = 0$$



1 1 0 - - - - - - - - - - - - - 0

Number of data points required for guessing an algebraic relation

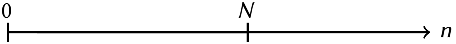
$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$



1	1	0 - - - - - - - - - - - - - 0		
t	0	1	0 - - - - - - - - - - - 0	

Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$



$$\begin{array}{c|cccccc}
1 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\
t & 0 & 1 & 0 & \cdots & \cdots & 0 \\
\vdots & \vdots & & \ddots & & & \\
t^d & 0 & \cdots & 0 & 1 & 0 \cdots & 0
\end{array}$$

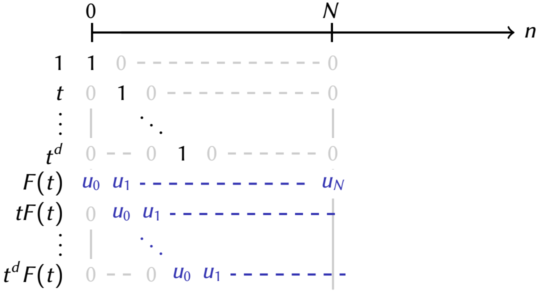
Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$

$$\begin{array}{r|llllllll}
 & 0 & & & & N & & & \xrightarrow{} n \\
1 & 1 & 0 & \cdots & \cdots & 0 & & & \\
t & 0 & 1 & 0 & \cdots & 0 & & & \\
\vdots & | & & \ddots & & & & & \\
t^d & 0 & \cdots & 0 \;\; 1 & 0 \cdots & 0 & & & \\
F(t) & u_0 & u_1 & \cdots & \cdots & u_N & & & \\
tF(t) & 0 & u_0 & u_1 & \cdots & \cdots & u_N & & \\
\vdots & | & & \ddots & & & & \ddots & \\
t^d F(t) & 0 & \cdots & 0 & u_0 & u_1 & \cdots & \cdots & u_N
\end{array}$$

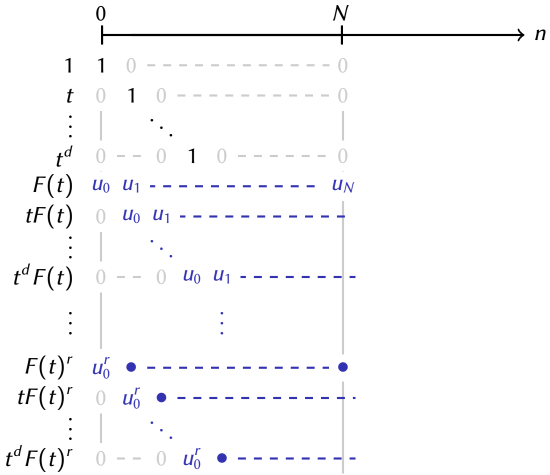
Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^{d}\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^{d}\right) F(t)^{r} = 0$$



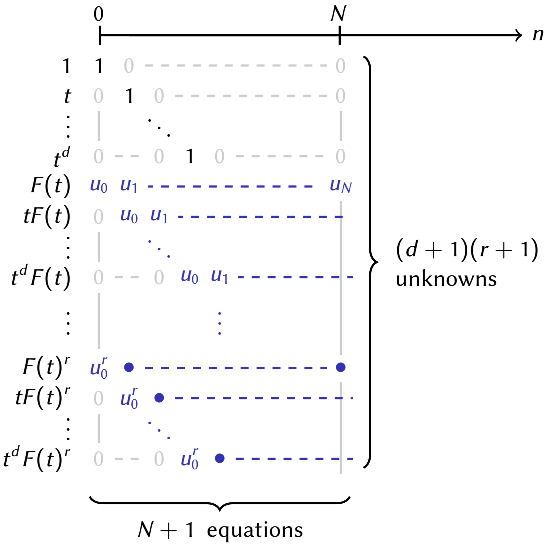
Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$



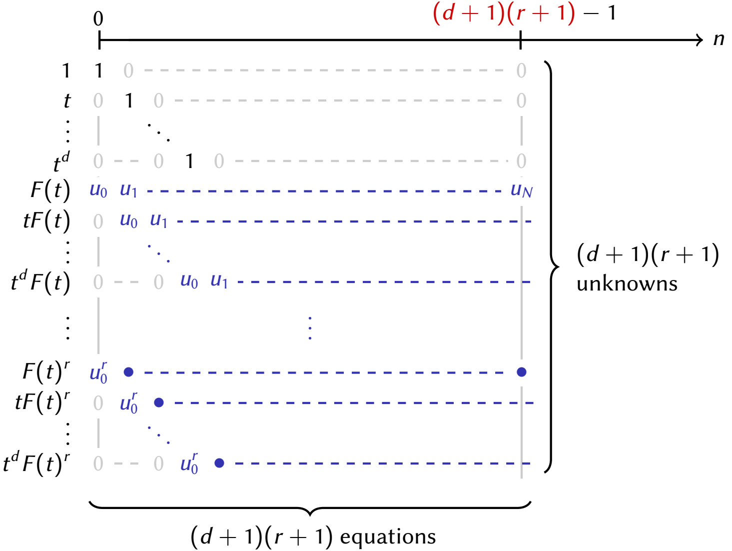
Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$



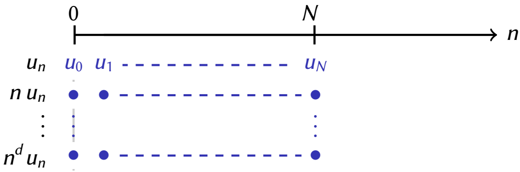
Number of data points required for guessing an algebraic relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F(t)^r = 0$$



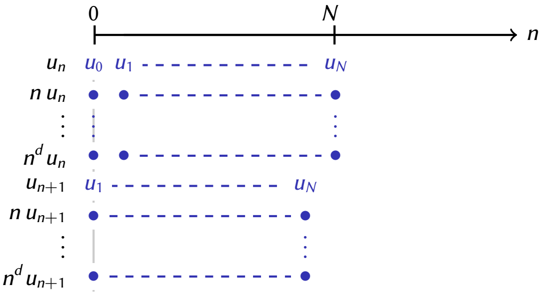
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^{d}\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^{d}\right) u_{n+r} = 0$$



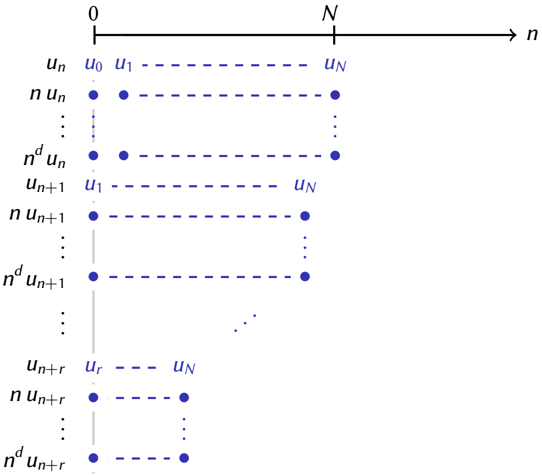
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^{d}\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^{d}\right) u_{n+r} = 0$$



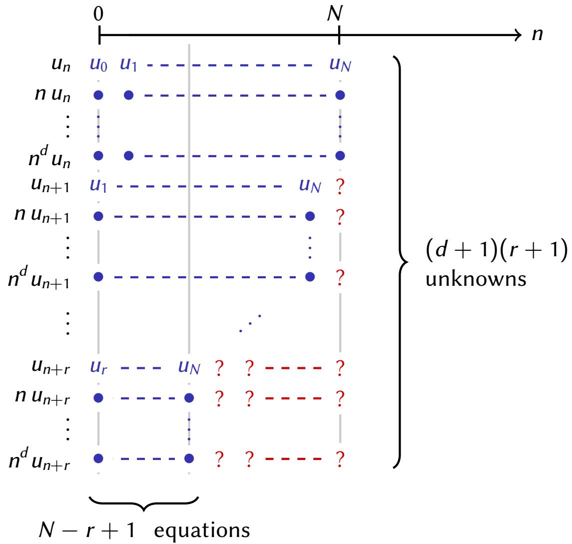
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^{d}\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^{d}\right) u_{n+r} = 0$$



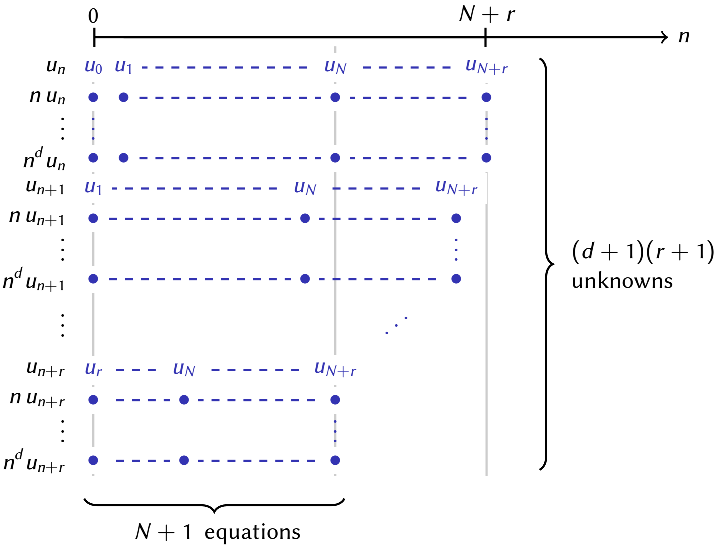
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^d\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^d\right) u_{n+r} = 0$$



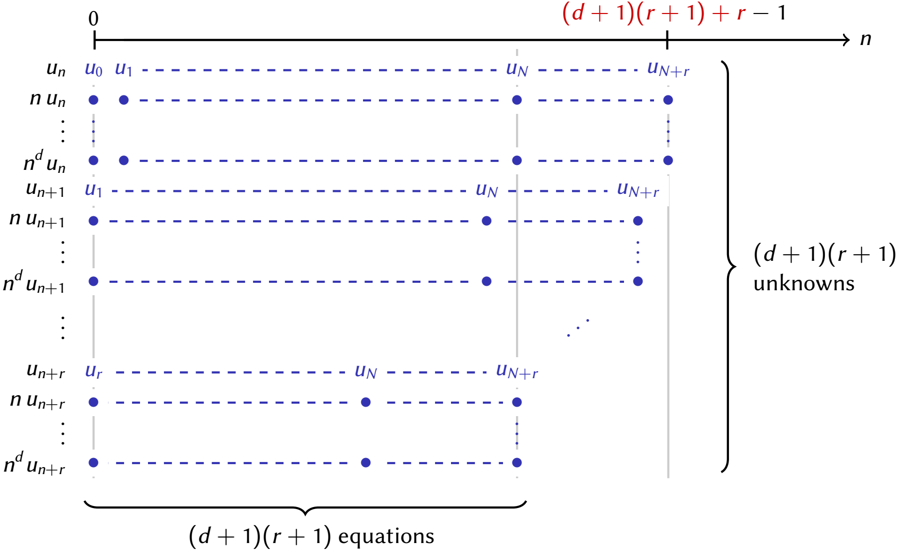
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^d\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^d\right) u_{n+r} = 0$$



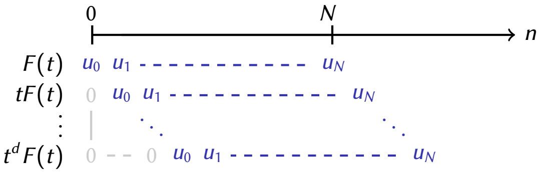
Number of data points required for guessing a recurrence relation

$$\left(a_{0,0} + a_{0,1}n + \cdots + a_{0,d}n^d\right) u_n + \cdots + \left(a_{r,0} + a_{r,1}n + \cdots + a_{r,d}n^d\right) u_{n+r} = 0$$



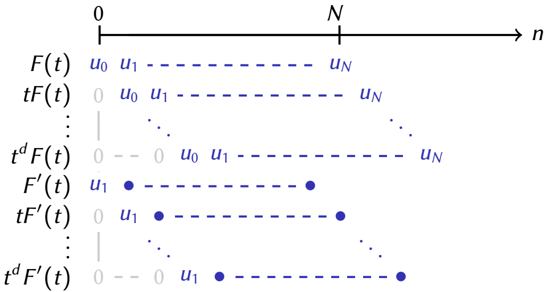
Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



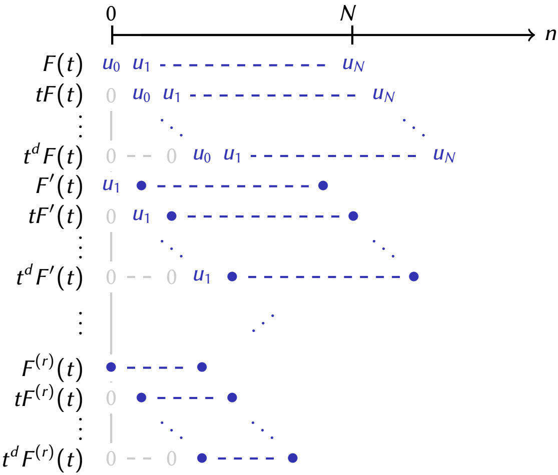
Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



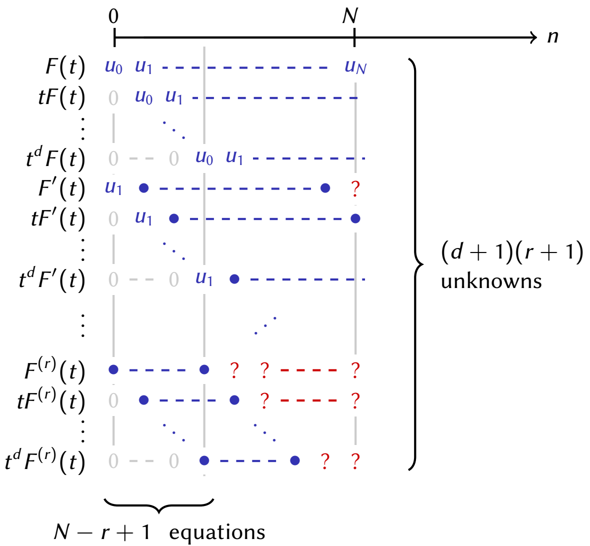
Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



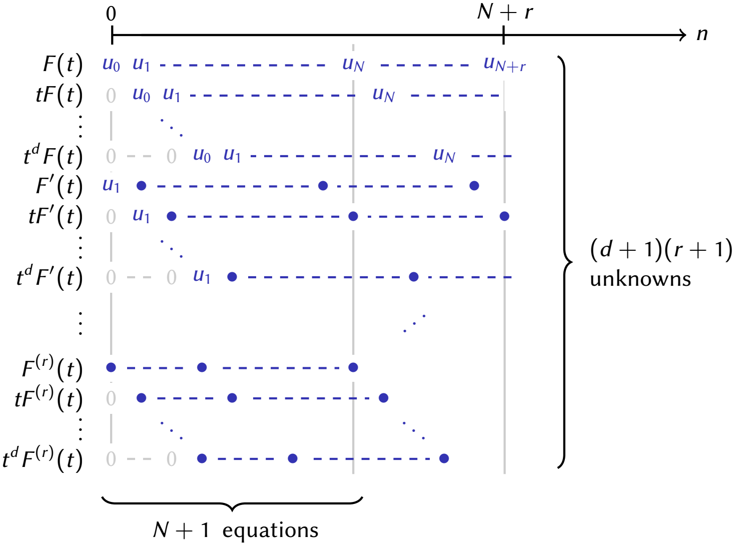
Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



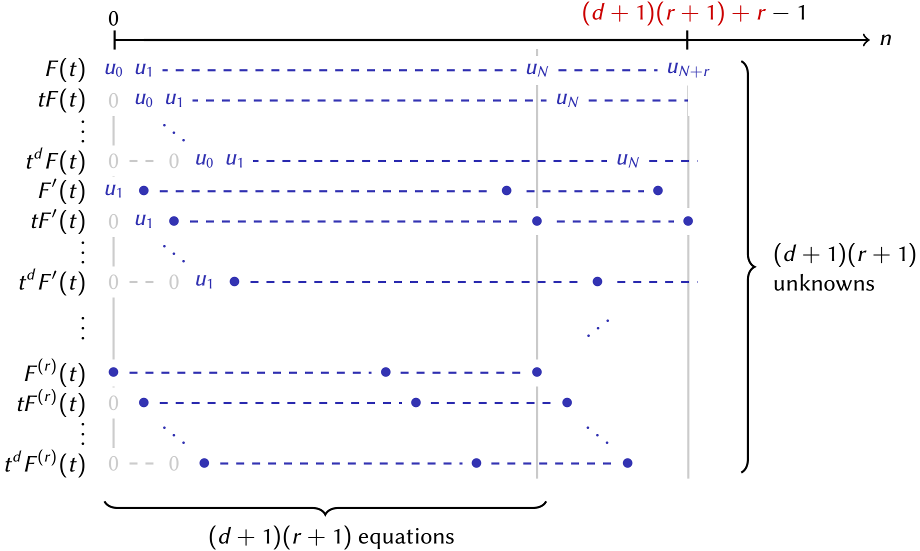
Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



Number of data points required for guessing a differential relation

$$\left(a_{0,0} + a_{0,1}t + \cdots + a_{0,d}t^d\right) F(t) + \cdots + \left(a_{r,0} + a_{r,1}t + \cdots + a_{r,d}t^d\right) F^{(r)}(t) = 0$$



Summary: formulas for the number of guessing points necessary

Type	Equation	Minimal data length
D-finite recurrence	$\sum_{i=0}^{r} \left(\sum_{j=0}^{d} a_{i,j} n^j \right) u_{n+i} = 0$	$(d+1)(r+1)+r$ $= (d+2)(r+1)-1$
D-finite differential	$\sum_{i=0}^{r} \left(\sum_{j=0}^{d} a_{i,j} t^j \right) F^{(j)}(t) = 0$	
Algebraic	$\sum_{i=0}^{r} \left(\sum_{j=0}^{d} a_{i,j} t^j \right) F(t)^i = 0$	$(d+1)(r+1)$

What happens if the system is interlaced with zeroes?

If instead of considering $u = (u_0 \ u_1 \ u_2 \dots)$,
we consider $v = (u_0 \ 0 \dots 0 \ u_1 \ 0 \dots 0 \ u_2 \dots)$...

- We get more data for free !
- But the equations that we want to find become larger.

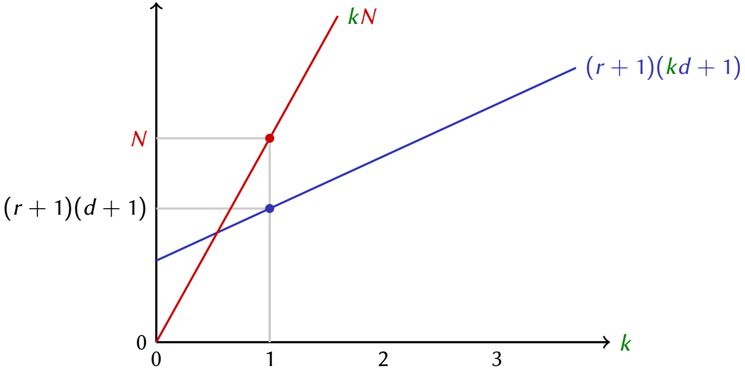
Of course, we expect that the equations will grow as fast or faster than the data!

Is it the case?

Algebraic equations (1)

Before	After
$F(t) = f_0 + f_1 t + \dots$	$G(t) = f_0 + \overbrace{0 + \cdots + 0}^{k-1} + f_1 t^k + \dots$ $= F(t^k)$
N data points	kN data points
$\sum_{i=0}^{d} P_i(t) F(t)^i = 0$	$\sum_{i=0}^{d} P_i(t^k) G(t)^i = 0$
Degree: d	Degree: kd
Order: r	Order: r
$(r+1)(d+1)$	$(r+1)(kd+1)$ $= kd(r+1) + r + 1$

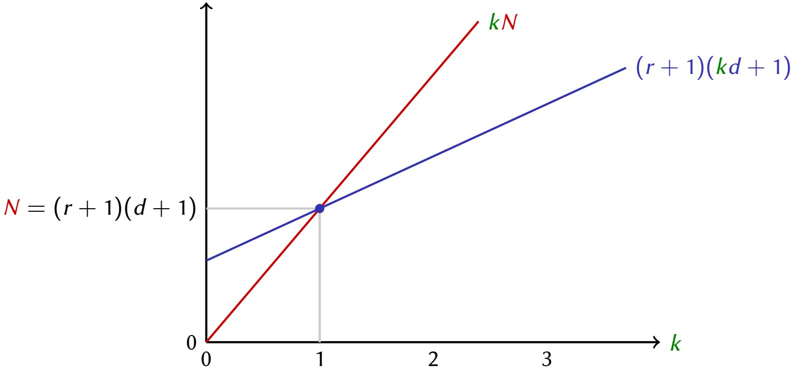
How many points do we need / have ?



- $(r+1)(d+1) \leq N$: enough, nothing to do
- $(r+1)(d+1) > N > (r+1)(d+1) - (r+1)$: enough for some $k > 1$!
- $(r+1)(d+1) - (r+1) \geq N$: not enough, nothing to do

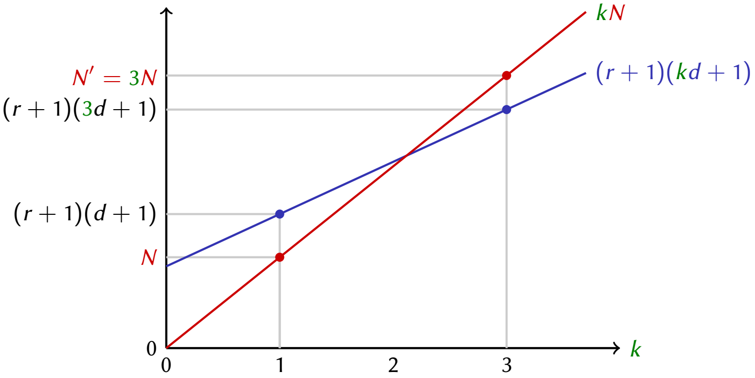
Algebraic equations (2)

How many points do we need / have ?



- $(r+1)(d+1) \leq N$: enough, nothing to do
- $(r+1)(d+1) > N > (r+1)(d+1) - (r+1)$: enough for some $k > 1$!
- $(r+1)(d+1) - (r+1) \geq N$: not enough, nothing to do

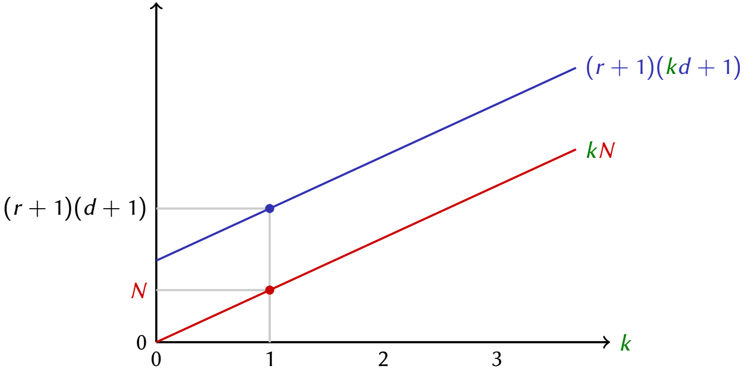
How many points do we need / have ?



- $(r+1)(d+1) \leq N$: enough, nothing to do
- $(r+1)(d+1) > N > (r+1)(d+1) - (r+1)$: enough for some $k > 1$!
- $(r+1)(d+1) - (r+1) \geq N$: not enough, nothing to do

Algebraic equations (2)

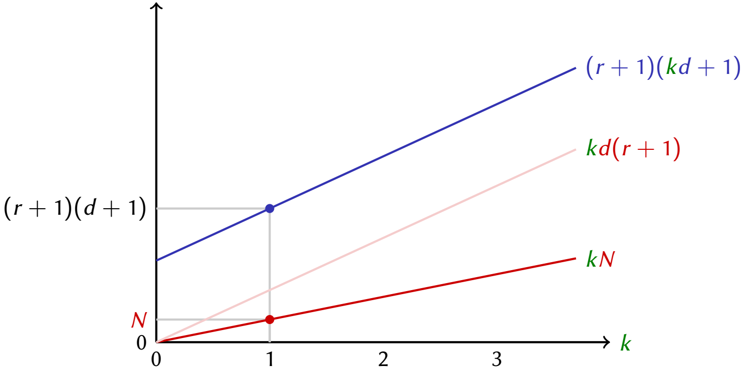
How many points do we need / have ?



- $(r+1)(d+1) \leq N$: enough, nothing to do
- $(r+1)(d+1) > N > (r+1)(d+1) - (r+1)$: enough for some $k > 1$!
- $(r+1)(d+1) - (r+1) \geq N$: not enough, nothing to do

Algebraic equations (2)

How many points do we need / have ?

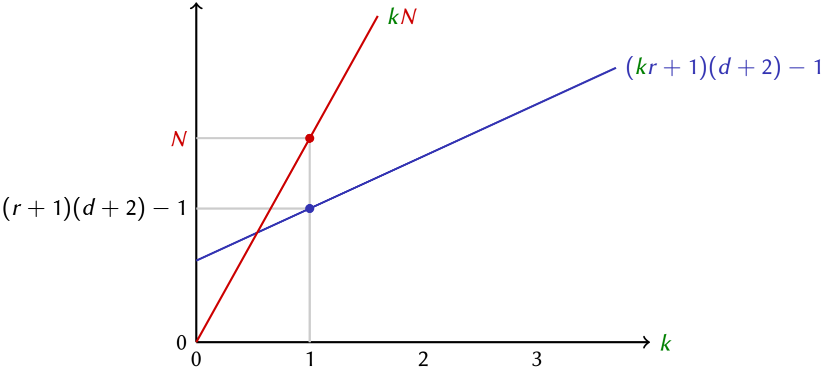


- $(r+1)(d+1) \leq N$: enough, nothing to do
- $(r+1)(d+1) > N > (r+1)(d+1) - (r+1)$: enough for some $k > 1$!
- $(r+1)(d+1) - (r+1) \geq N$: not enough, nothing to do

Recurrence relations (1)

Before	After
$u = (u_0\ u_1\ u_2 \dots)$	$v = (u_0\ \overbrace{0 \dots 0}^{k-1}\ u_1\ 0 \dots 0\ u_2 \dots)$
N data points	kN data points
$\sum_{i=0}^{d} P_i(n) S^i(u) = 0$	$\sum_{i=0}^{d} P_i(kn) S^{ki}(v) = 0$
Degree: d	Degree: d
Order: r	Order: kr
$(r+1)(d+2)-1$	$(kr+1)(d+2)-1$ $= kr(d+2)+d+1$

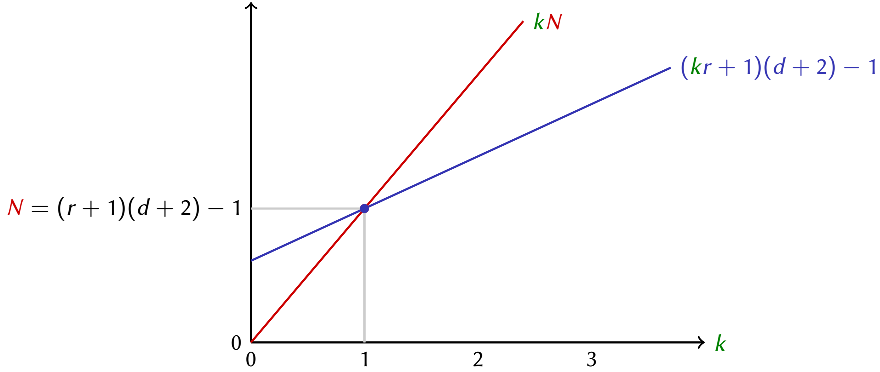
How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N > (r+1)(d+2)-(d+1)$: enough for some $k > 1$!
- $(r+1)(d+2)-(d+1) \geq N$: not enough, nothing to do

Recurrence relations (2)

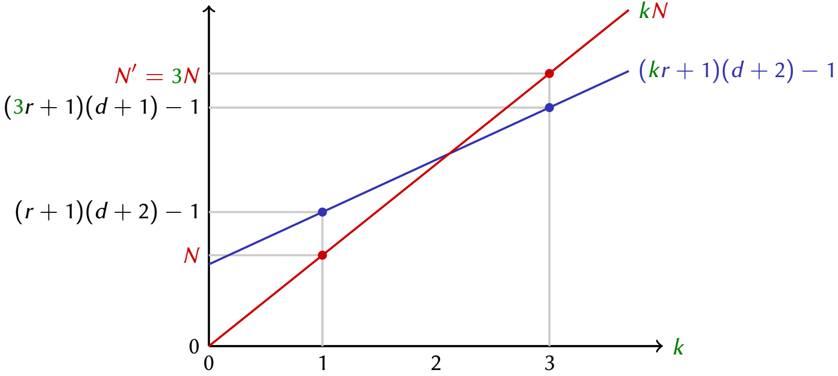
How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N > (r+1)(d+2)-(d+1)$: enough for some $k > 1$!
- $(r+1)(d+2)-(d+1) \geq N$: not enough, nothing to do

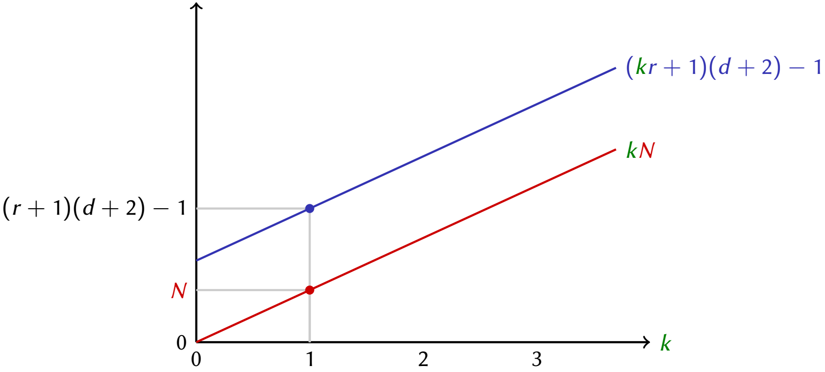
Recurrence relations (2)

How many points do we need / have ?



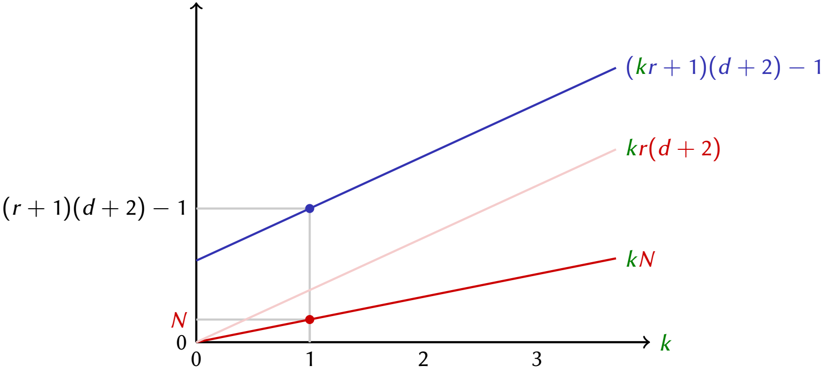
- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N > (r+1)(d+2)-(d+1)$: enough for some $k > 1$!
- $(r+1)(d+2)-(d+1) \geq N$: not enough, nothing to do

How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N > (r+1)(d+2)-(d+1)$: enough for some $k>1$!
- $(r+1)(d+2)-(d+1) \geq N$: not enough, nothing to do

How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N > (r+1)(d+2)-(d+1)$: enough for some $k>1$!
- $(r+1)(d+2)-(d+1) \geq N$: not enough, nothing to do

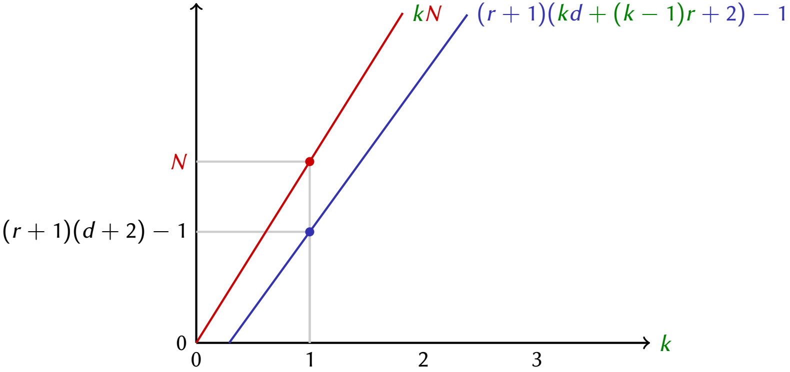
Differential relations (1)

Before	After
$F(t) = f_0 + f_1 t + \dots$	$G(t) = f_0 + \overbrace{0 + \cdots + 0}^{k-1} + f_1 t^k + \dots$ $= F(t^k)$
N data points	kN data points
$\sum_{i=0}^{d} P_i(t) F^{(i)}(t) = 0$	$\sum_{i=0}^{d} Q_i(t) G^{(i)}(t) = 0$
Degree: d	Degree: $kd + (k-1)r$
Order: r	Order: r
$(r+1)(d+2) - 1$	$(r+1)(d + (k-1)r + 2) - 1$ $= k(d+r)(r+1) \underbrace{-(r+1)(r-2) - 1}_{\leq 0}$

$$\boxed{\begin{aligned} G(t) &= F(t^k) \\ G'(t) &= kt^{k-1} F'(t^k) \\ &\vdots \\ G^{(r)}(t) &= \bullet t^{r(k-1)} F^{(r)}(t^k) \\ &\quad + \dots \end{aligned}}$$

Differential relations (2)

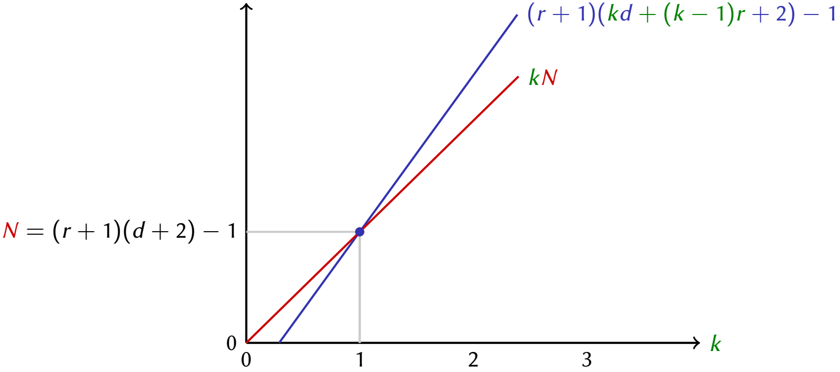
How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N$: not enough, nothing to do
- But it is usually a good idea to ensure that $k = 1$!

Differential relations (2)

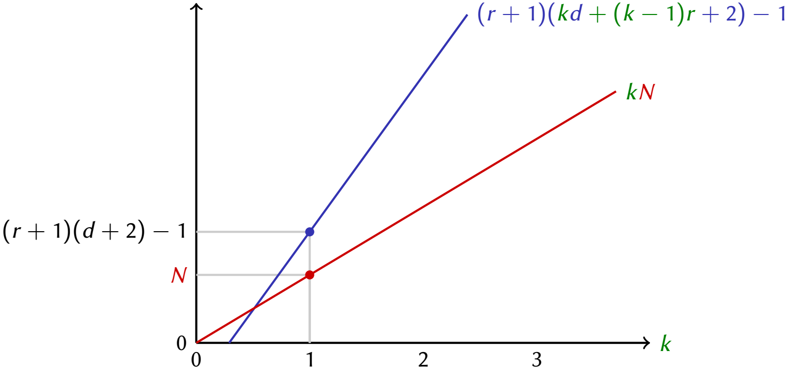
How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N$: not enough, nothing to do
- But it is usually a good idea to ensure that $k=1$!

Differential relations (2)

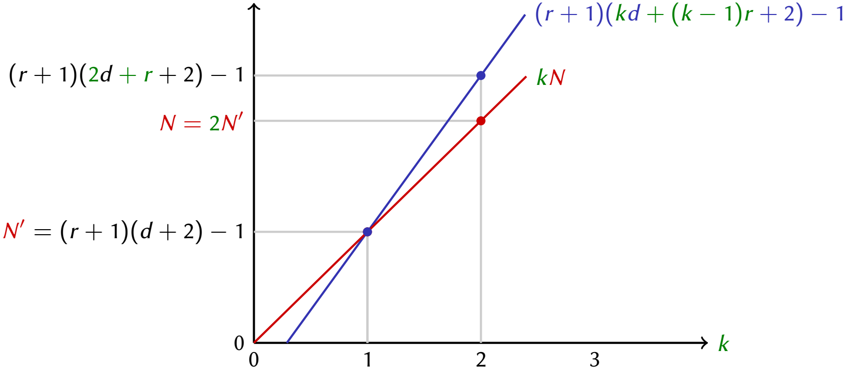
How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N$: not enough, nothing to do
- But it is usually a good idea to ensure that $k=1$!

Differential relations (2)

How many points do we need / have ?



- $(r+1)(d+2)-1 \leq N$: enough, nothing to do
- $(r+1)(d+2)-1 > N$: not enough, nothing to do
- But it is usually a good idea to ensure that $k=1$!

Conclusion and questions

Did we magically create data?

- Maybe not, if we force the algorithm to require less data, it still works
- The algorithm had enough data from the start but did not know it?

Consequences:

- Existing implementations can take advantage of this fact to gain a few points
- Must take into account the cost (space and time) of building and solving a larger system
- Actually if they detect such a structure, they can remove the zeroes unconditionally
- But this will never yield a lot of free data

Best case scenario:

- N is small ($\sim$ a few hundreds)
- Finding the equation would just require $N + 2$ terms
- Data is very expensive, even for just 2 more terms
- Does it ever happen?

Conclusion and questions

Did we magically create data?

- Maybe not, if we force the algorithm to require less data, it still works
- The algorithm had enough data from the start but did not know it?

Consequences:

- Existing implementations can take advantage of this fact to gain a few points
- Must take into account the cost (space and time) of building and solving a larger system
- Actually if they detect such a structure, they can remove the zeroes unconditionally
- But this will never yield a lot of free data

Best case scenario:

- N is small ($\sim$ a few hundreds)
- Finding the equation would just require $N + 2$ terms
- Data is very expensive, even for just 2 more terms
- Does it ever happen?

Conclusion and questions

Did we magically create data?

- Maybe not, if we force the algorithm to require less data, it still works
- The algorithm had enough data from the start but did not know it?

Consequences:

- Existing implementations can take advantage of this fact to gain a few points
- Must take into account the cost (space and time) of building and solving a larger system
- Actually if they detect such a structure, they can remove the zeroes unconditionally
- But this will never yield a lot of free data

Best case scenario:

- N is small ($\sim$ a few hundreds)
- Finding the equation would just require $N + 2$ terms
- Data is very expensive, even for just 2 more terms
- Does it ever happen?

Conclusion and questions

Did we magically create data?

- Maybe not, if we force the algorithm to require less data, it still works
- The algorithm had enough data from the start but did not know it?

Consequences:

- Existing implementations can take advantage of this fact to gain a few points
- Must take into account the cost (space and time) of building and solving a larger system
- Actually if they detect such a structure, they can remove the zeroes unconditionally
- But this will never yield a lot of free data

Best case scenario:

- N is small ($\sim$ a few hundreds)
- Finding the equation would just require $N + 2$ terms
- Data is very expensive, even for just 2 more terms
- Does it ever happen?

Thank you for your attention!