Parallelism in algebraic geometry
Examples with Singular and GPI-Space

Anne Frühbis-Krüger
joint work with Janko Böhm SFB/TRR 195
and with Mirko Rahn, Fraunhofer ITWM, Kaiserslautern
(and many others from the SINGULAR and GPI-Space Teams)

Institut für Mathematik
Carl von Ossietzky Universitt Oldenburg

ICERM, 18.02.2021

Common knowledge I

Why parallel?

Common knowledge I

Why parallel?

state of the art hardware:

- multicore computers
- high performance clusters

Common knowledge I

Why parallel?

state of the art hardware:

- multicore computers
- high performance clusters

success stories:

Common knowledge I

Why parallel?

state of the art hardware:

- multicore computers
- high performance clusters

success stories:

- numerical analysis

Common knowledge I

Why parallel?

state of the art hardware:

- multicore computers
- high performance clusters

success stories:

- numerical analysis
- simulation of flows and turbulence

Common knowledge I

Why parallel?

state of the art hardware:

- multicore computers
- high performance clusters

success stories:

- numerical analysis
- simulation of flows and turbulence

practical drawback:
management of distribution of resources and data

Common knowledge II

ideal speed-up:
linear in the number of cores

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)
- communication overhead

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)
- communication overhead
- locking of data

kinds of parallelism:

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)
- communication overhead
- locking of data

kinds of parallelism:

- fine grained
  (synchronize/communicate data in short intervalls)

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)
- communication overhead
- locking of data

kinds of parallelism:

- fine grained
  (synchronize/communicate data in short intervalls)
- coarse grained
  (occasionally synchronize/communicate data)

Common knowledge II

ideal speed-up:
linear in the number of cores

limiting factors in reality:

- longest non-parallelizable part (critical chain)
- communication overhead
- locking of data

kinds of parallelism:

- fine grained
  (synchronize/communicate data in short intervalls)
- coarse grained
  (occasionally synchronize/communicate data)
- embarrassing
  (nearly no communication between tasks)

Parallelism in Algorithmic Algebraic Geometry

Traditionally 3 main approaches:

1. Parallelization on the level of arithmetic
   $\longrightarrow$ fine-grained/shared memory,
   $\longrightarrow$ limited to one machine

Parallelism in Algorithmic Algebraic Geometry

Traditionally 3 main approaches:

1. Parallelization on the level of arithmetic
   $\longrightarrow$ fine-grained/shared memory,
   $\longrightarrow$ limited to one machine
2. Embarassing parallelism of zero-dimensional objects
   $\longrightarrow$ scales up to number of instances

Parallelism in Algorithmic Algebraic Geometry

Traditionally 3 main approaches:

1. Parallelization on the level of arithmetic
   $\longrightarrow$ fine-grained/shared memory,
   $\longrightarrow$ limited to one machine
2. Embarassing parallelism of zero-dimensional objects
   $\longrightarrow$ scales up to number of instances
3. Modular approaches
   $\longrightarrow$ basically borrowed idea from number theory

Parallelism in Algorithmic Algebraic Geometry

Traditionally 3 main approaches:

1. Parallelization on the level of arithmetic
   $\longrightarrow$ fine-grained/shared memory,
   $\longrightarrow$ limited to one machine
2. Embarassing parallelism of zero-dimensional objects
   $\longrightarrow$ scales up to number of instances
3. Modular approaches
   $\longrightarrow$ basically borrowed idea from number theory

More recently: inherent natural parallelism of a setting

Parallelism in Algorithmic Algebraic Geometry

Traditionally 3 main approaches:

1. Parallelization on the level of arithmetic
   $\longrightarrow$ fine-grained/shared memory,
   $\longrightarrow$ limited to one machine
2. Embarassing parallelism of zero-dimensional objects
   $\longrightarrow$ scales up to number of instances
3. Modular approaches
   $\longrightarrow$ basically borrowed idea from number theory

More recently: inherent natural parallelism of a setting
e.g.:

- gfan
- talk of Lars Kastner using polymake and MPTOPCOM

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential
4. manage data exchange
5. manage resources

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential
4. manage data exchange
5. manage resources

Is each of us an expert in all of these tasks?

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential
4. manage data exchange
5. manage resources

Is each of us an expert in all of these tasks? Certainly not.

Using a workflow management system:

- concentrate on 1 and 2

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential
4. manage data exchange
5. manage resources

Is each of us an expert in all of these tasks? Certainly not.

Using a workflow management system:

- concentrate on 1 and 2
- give some thought to 3, allow help from system

Separating computation and coordination layer

Tasks in parallel setting (classically):

1. design a mathematical algorithm
2. implement mathematical algorithm
3. locate and exploit its parallelization potential
4. manage data exchange
5. manage resources

Is each of us an expert in all of these tasks? Certainly not.

Using a workflow management system:

- concentrate on 1 and 2
- give some thought to 3, allow help from system
- leave 4 and 5 to the system

Success stories along these lines

Fan/Graph-Traversals:

- GIT-fan (Böhm, FK, Rahn, Reinbold):
  - Mori chamber decomposition of cone of movable divisors of $M_{0,6}$
  - 12,5 minutes on 640 cores

Success stories along these lines

Fan/Graph-Traversals:

- GIT-fan (Böhm, FK, Rahn, Reinbold):
  - Mori chamber decomposition of cone of movable divisors of $M_{0,6}$
  - 12,5 minutes on 640 cores
- tropical varieties (Bendle, Böhm, Rahn):
  - tropical Grassmannian $\mathcal{G}_{3,8}$
  - less than 20 minutes on 768 cores

Success stories along these lines

Fan/Graph-Traversals:

- GIT-fan (Böhm, FK, Rahn, Reinbold):
  - Mori chamber decomposition of cone of movable divisors of $M_{0,6}$
  - 12,5 minutes on 640 cores
- tropical varieties (Bendle, Böhm, Rahn):
  - tropical Grassmannian $\mathcal{G}_{3,8}$
  - less than 20 minutes on 768 cores

Coverings:

- smoothness test (Böhm, Decker FK, Rahn, Ristau, Pfreundt) – see below
- desing. of 2-dim. schemes/$\mathbb{Z}$ (FK, Gaube, Schober, Ristau)
- order zeta functions (FK, Maglione, Schober, Voll)

What we used: GPI-space

GPI-Space (Fraunhofer ITWM)

- originally designed with applications in finite element methods in mind
- separation of coordination and computation layers

What we used: GPI-space

GPI-Space (Fraunhofer ITWM)

- originally designed with applications in finite element methods in mind
- separation of coordination and computation layers
- distributed, robust, scalable Run-Time-System (resource management, scheduler)

What we used: GPI-space

GPI-Space (Fraunhofer ITWM)

- originally designed with applications in finite element methods in mind
- separation of coordination and computation layers
- distributed, robust, scalable Run-Time-System (resource management, scheduler)
- application independent global memory layer
- Petri net based workflow engine (automated parallel execution, dependency tracking)

What we used: GPI-space

GPI-Space (Fraunhofer ITWM)

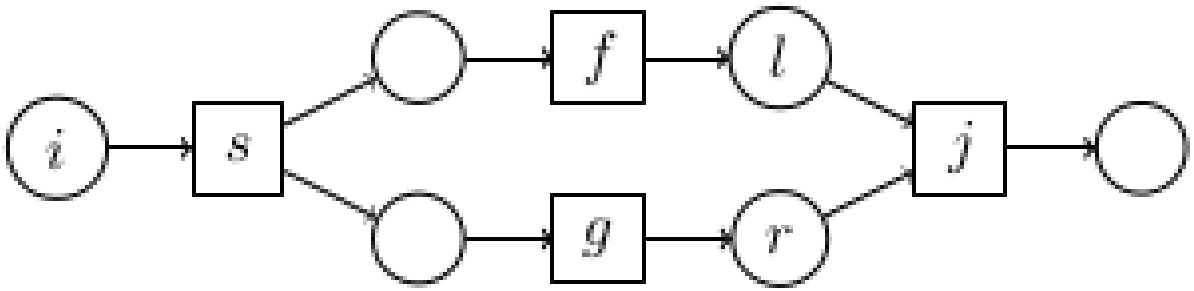
- originally designed with applications in finite element methods in mind
- separation of coordination and computation layers
- distributed, robust, scalable Run-Time-System (resource management, scheduler)
- application independent global memory layer
- Petri net based workflow engine (automated parallel execution, dependency tracking)

biggest advantage:

Parallelisation without modification to the backend

Petri nets

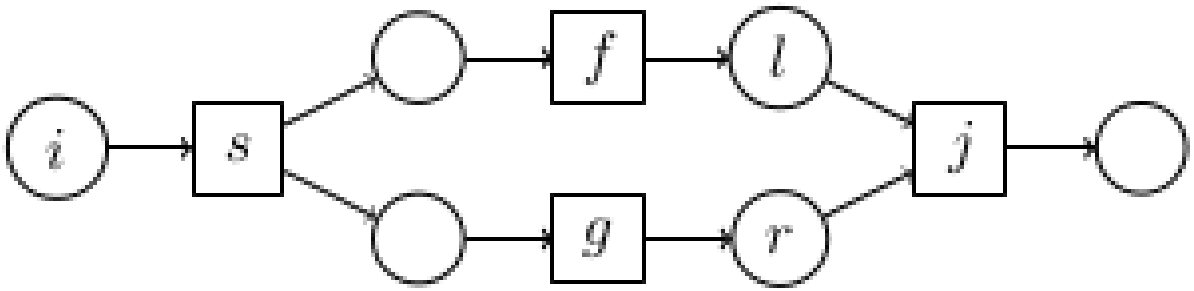
Petri net:



- directed graph

Petri nets

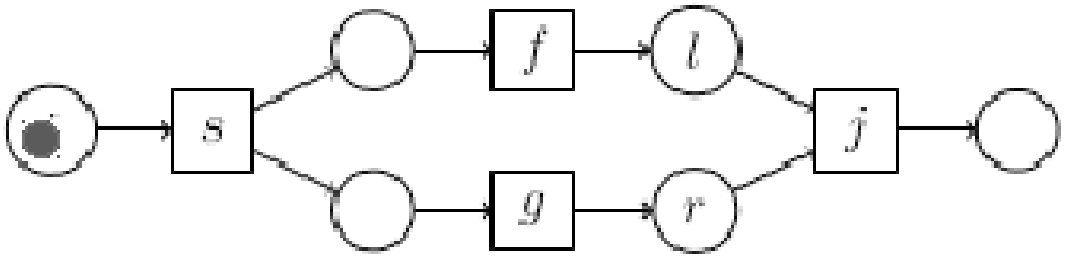
Petri net:



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

Petri nets

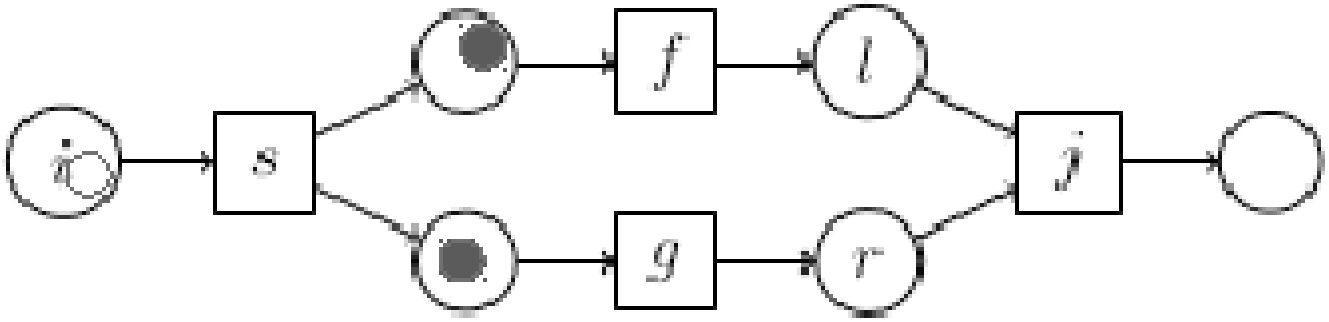
Petri net (running on a single core):



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

Petri nets

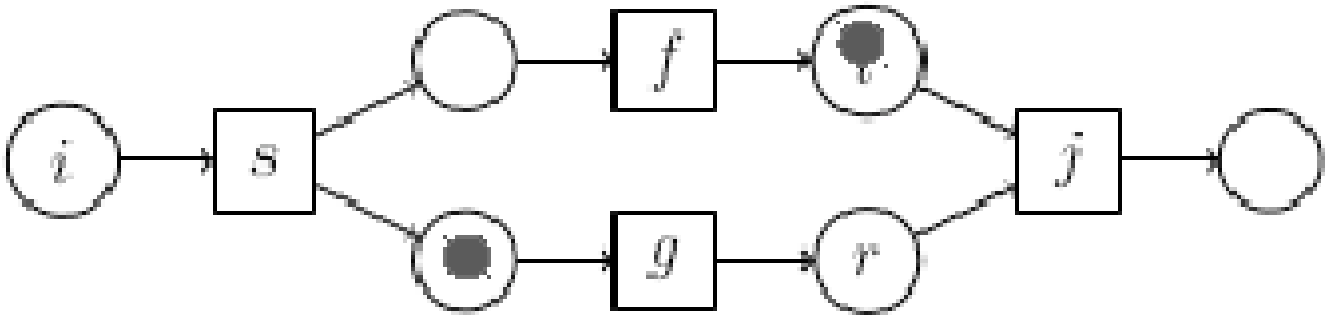
Petri net (running on a single core):



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

Petri nets

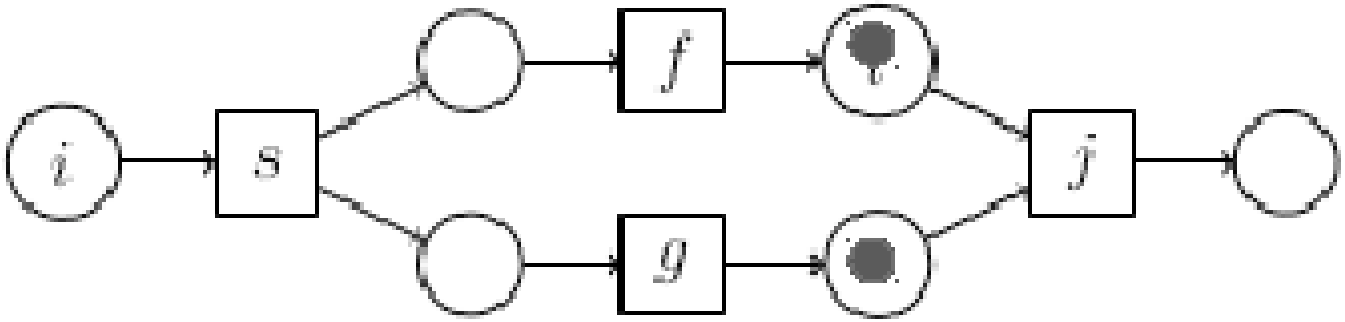
Petri net (running on a single core):



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

Petri nets

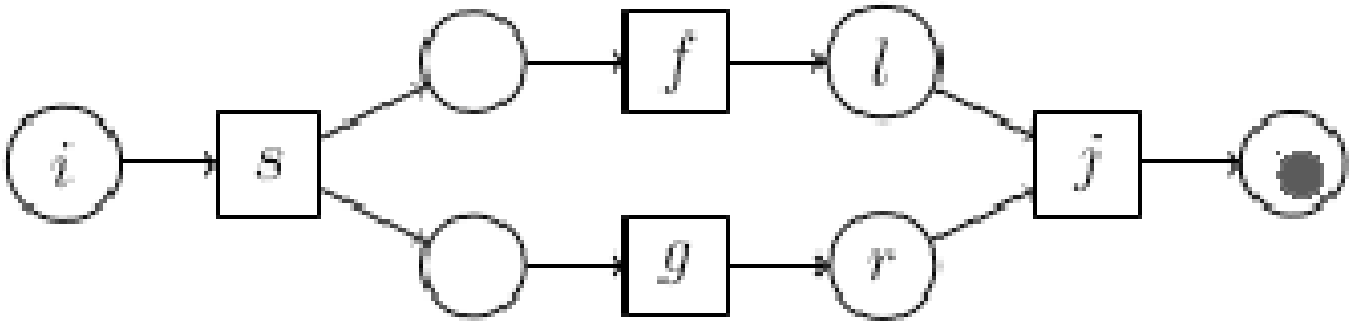
Petri net (running on a single core):



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

Petri nets

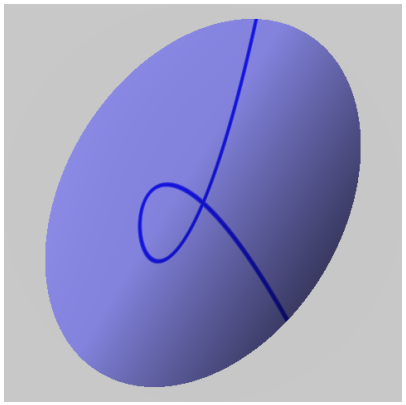
Petri net:



- directed graph
- 2 kinds of vertices: places (circle) and transitions (box)
- places hold marked tokens
- transitions contain functional units

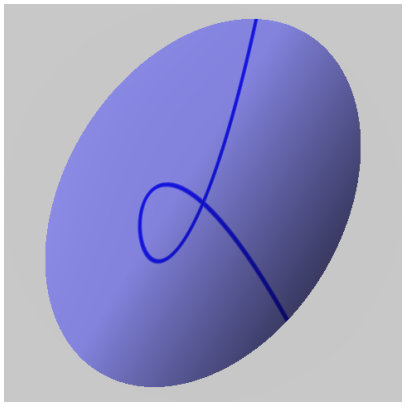
The singular locus I

First for plane curves:



The singular locus I

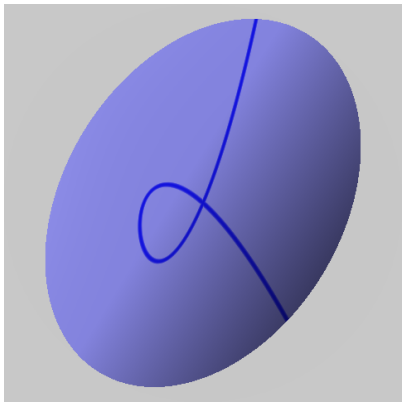
First for plane curves:



$V(f)$ singular at $p \in \mathbb{C}^2$ $\Leftrightarrow$ not a unique tangent direction at p

The singular locus I

First for plane curves:

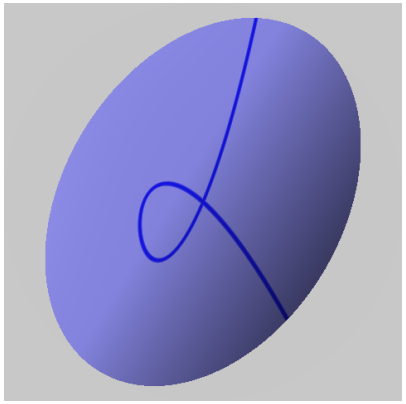


$V(f)$ singular at $p \in \mathbb{C}^2$ $\Leftrightarrow$ not a unique tangent direction at p

$\Leftrightarrow$ tangent space to X at p not of dimension 1

The singular locus I

First for plane curves:



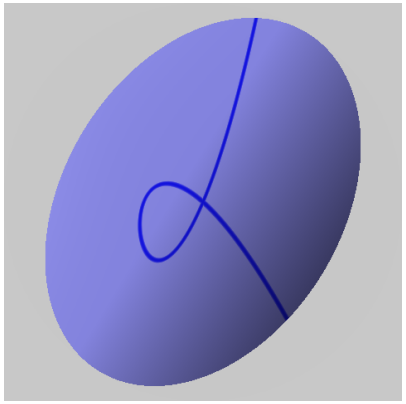
$V(f)$ singular at $p \in \mathbb{C}^2$ $\Leftrightarrow$ not a unique tangent direction at p

$\Leftrightarrow$ tangent space to X at p not of dimension 1

$\Leftrightarrow$ $\frac{\partial f}{\partial x_1}(p) = 0 = \frac{\partial f}{\partial x_2}(p)$

The singular locus I

First for plane curves:



$V(f)$ singular at $p \in \mathbb{C}^2$ $\Leftrightarrow$ not a unique tangent direction at p

$\Leftrightarrow$ tangent space to X at p not of dimension 1

$\Leftrightarrow$ $\frac{\partial f}{\partial x_1}(p) = 0 = \frac{\partial f}{\partial x_2}(p)$

$\Leftrightarrow$ Jacobian matrix J of f at p of rank < 1

The singular locus II

Jacobian Criterion

Let all components of $V(f_1, \ldots, f_m)$ be of dimension d

The singular locus II

Jacobian Criterion

Let all components of $V(f_1, \ldots, f_m)$ be of dimension d

Then:
$V(f_1, \ldots, f_m)$ sing. in $p \in \mathbb{C}^n \quad \Leftrightarrow \quad \dim_{\mathbb{C}} T_p(X) > d$

The singular locus II

Jacobian Criterion

Let all components of $V(f_1, \ldots, f_m)$ be of dimension d

Then:

$$V(f_1, \ldots, f_m) \text{ sing. in } p \in \mathbb{C}^n \quad \Leftrightarrow \quad \dim_{\mathbb{C}} T_p(X) > d$$

$$\Leftrightarrow \quad \text{rank of Jacobian matrix } J \text{ of } (f_1 \ldots, f_m) \text{ at } p < n - d$$

The singular locus II

Jacobian Criterion

Let all components of $V(f_1, \ldots, f_m)$ be of dimension d

Then:

$$V(f_1, \ldots, f_m) \text{ sing. in } p \in \mathbb{C}^n \quad \Leftrightarrow \quad \dim_{\mathbb{C}} T_p(X) > d$$

$$\Leftrightarrow \quad \text{rank of Jacobian matrix } J \text{ of } (f_1 \ldots, f_m) \text{ at } p < n - d$$

singular locus:

$$\text{Sing}(X) \quad := \quad \{p \in X \mid X \text{ singular at } p\}$$

The singular locus II

Jacobian Criterion

Let all components of $V(f_1, \ldots, f_m)$ be of dimension d

Then:

$$\begin{aligned} V(f_1, \ldots, f_m) \text{ sing. in } p \in \mathbb{C}^n \quad &\Leftrightarrow \quad \dim_{\mathbb{C}} T_p(X) > d \\ &\Leftrightarrow \quad \text{rank of Jacobian matrix } J \text{ of } (f_1 \ldots, f_m) \text{ at } p \\ &\qquad\qquad < n - d \end{aligned}$$

singular locus:

$$\begin{aligned} \mathrm{Sing}(X) &:= \{p \in X \mid X \text{ singular at } p\} \\ &= V(f_1, \ldots, f_m \text{ and } (n - \dim(X)) - \\ &\qquad \text{minors of Jacobian matrix } J) \end{aligned}$$

A real-life example – the problem

numerical Godeaux-Surface (from current research/ surface with prescribed properties):

13 variables, dimension 2, 22 polynomials

$\Longrightarrow$ approx. 55 Mio. minors

A real-life example – the problem

numerical Godeaux-Surface (from current research/ surface with prescribed properties):

13 variables, dimension 2, 22 polynomials
$\Longrightarrow$ approx. 55 Mio. minors

Jacobian Criterion tests : $\mathrm{Sing}(X) == \emptyset$?

A real-life example – the problem

numerical Godeaux-Surface (from current research/ surface with prescribed properties):

13 variables, dimension 2, 22 polynomials
$\Longrightarrow$ approx. 55 Mio. minors

Jacobian Criterion tests : $\mathrm{Sing}(X) == \emptyset$?

- that is: $1 \in I + \langle (n-d) \text{ minors von } J \rangle$?
- Groebner Basis computations: all minors as input (size of pairset!)
- certificate of smoothness or singularity after **all** computations

The setting

$\mathbb{K}$ algebraically closed field

The setting

$\mathbb{K}$ algebraically closed field

$f_1, \ldots, f_r \in \mathbb{K}[x_1, \ldots, x_n]$ and $X = V(f_1, \ldots, f_r)$

The setting

$\mathbb{K}$ algebraically closed field

$f_1, \ldots, f_r \in \mathbb{K}[x_1, \ldots, x_n]$ and $X = V(f_1, \ldots, f_r)$

$I_X = I(X) \subset \mathbb{K}[x_1, \ldots, x_n]$ vanishing ideal

The setting

$\mathbb{K}$ algebraically closed field

$f_1, \ldots, f_r \in \mathbb{K}[x_1, \ldots, x_n]$ and $X = V(f_1, \ldots, f_r)$

$I_X = I(X) \subset \mathbb{K}[x_1, \ldots, x_n]$ vanishing ideal

Goal:

Test whether X is singular
without computation of singular locus!

Hironaka's Approach

Hironaka(1964):

resolution of singularities in characteristic zero

Hironaka's Approach

Hironaka(1964):

resolution of singularities in characteristic zero

- finitely many blow-ups in 'good' centers
- termination criterion **without** Jacobian criterion

Hironaka's Approach

Hironaka(1964):

resolution of singularities in characteristic zero

- finitely many blow-ups in 'good' centers
- termination criterion **without** Jacobian criterion

Hironaka's measure for distance to smoothness at p:

$$\nu^*(X, p) := (\mathrm{ord}_p(g_1), \ldots, \mathrm{ord}_p(g_s))$$

where

Hironaka's Approach

Hironaka(1964):

resolution of singularities in characteristic zero

- finitely many blow-ups in 'good' centers
- termination criterion **without** Jacobian criterion

Hironaka's measure for distance to smoothness at p:

$$\nu^*(X, p) := (\mathrm{ord}_p(g_1), \ldots, \mathrm{ord}_p(g_s))$$

where

- $g_1, \ldots, g_s$ minimal standard basis von $I_X \mathcal{O}_{\mathbb{A}^n_{\mathbb{K}}, p}$ w.r.t. local degree ordering, sorted by increasing order.

Hironaka's Approach

Hironaka(1964):

resolution of singularities in characteristic zero

- finitely many blow-ups in 'good' centers
- termination criterion **without** Jacobian criterion

Hironaka's measure for distance to smoothness at p:

$$\nu^*(X,p) := (\mathrm{ord}_p(g_1),\ldots,\mathrm{ord}_p(g_s))$$

where

- $g_1,\ldots,g_s$ minimal standard basis von $I_X\mathcal{O}_{\mathbb{A}^n_{\mathbb{K}},p}$ w.r.t. local degree ordering, sorted by increasing order.
- $\mathrm{ord}_p(g_i) = \max\{k\in\mathbb{Z} \mid g_i \in \mathfrak{m}^k_{\mathbb{A}^n_K,p}\}$

Hironaka's Approach II

defined a moment ago:
$\nu^*(X,p) := (\mathrm{ord}_p(g_1),\ldots,\mathrm{ord}_p(g_s))$

Lemma (Hironaka)

X singular at p

$$\Longleftrightarrow \nu^*(X,p) >_{lex} \underbrace{(1,\ldots,1)}_{n-\dim(X)}$$

Hironaka's Approach II

defined a moment ago:
$\nu^*(X,p) := (\mathrm{ord}_p(g_1),\ldots,\mathrm{ord}_p(g_s))$

Lemma (Hironaka)

X singular at p

$$\Longleftrightarrow \nu^*(X,p) >_{lex} \underbrace{(1,\ldots,1)}_{n-\dim(X)}$$

practical problem:

standard basis locally at **each** point impossible

Locus of Order 1

First entry of ν^*: order of ideal at p

$$\mathrm{ord}_p(I) := \min\{\mathrm{ord}_p(h) \mid h \in I\}$$

Locus of Order 1

First entry of ν^*: order of ideal at p

$$\mathrm{ord}_p(I) := \min\{\mathrm{ord}_p(h) \mid h \in I\}$$

We know:

$$\mathrm{ord}_p(I) = 1 \implies \exists h \in I : \mathrm{ord}_p(h) = 1$$

Locus of Order 1

First entry of ν^*: order of ideal at p

$$\mathrm{ord}_p(I) := \min\{\mathrm{ord}_p(h) \mid h \in I\}$$

We know:

$$\mathrm{ord}_p(I) = 1 \implies \exists h \in I : \mathrm{ord}_p(h) = 1$$

$$\implies \begin{array}{l} \bullet \text{ first entry of } \nu^*(X,p) \text{ is } 1 \\ \bullet\ V(h) \text{ smooth hypersurface locally at } p \\ \quad X \subset V(h) \end{array}$$

Locus of Order 1

First entry of ν^*: order of ideal at p

$$\mathrm{ord}_p(I) := \min\{\mathrm{ord}_p(h) \mid h \in I\}$$

We know:

$$\mathrm{ord}_p(I) = 1 \implies \exists h \in I : \mathrm{ord}_p(h) = 1$$

$$\implies \begin{array}{l} \bullet \text{ first entry of } \nu^*(X,p) \text{ is } 1 \\ \bullet\ V(h) \text{ smooth hypersurface locally at } p \\ \quad X \subset V(h) \end{array}$$

locus of order ≥ 2 of X:

$$\Delta(I_x) = V(\langle f_1, \ldots, f_s, \frac{\partial f_i}{\partial x_j} \mid 1 \leq i \leq s, 1 \leq j \leq n \rangle)$$

Local System of Parameters

we have seen:

$$\mathrm{ord}_p(I) = 1 \Longrightarrow \exists h \in I : V(h) \text{ smooth and } X \subset V(h) \text{ near } p$$

Local System of Parameters

we have seen:

$$\mathrm{ord}_p(I) = 1 \Longrightarrow \exists h \in I : V(h) \text{ smooth and } X \subset V(h) \text{ near } p$$

Choose $V(h)$ locally at p as new ambient space

Local System of Parameters

we have seen:

$$\mathrm{ord}_p(I) = 1 \Longrightarrow \exists h \in I : V(h) \text{ smooth and } X \subset V(h) \text{ near } p$$

Choose $V(h)$ locally at p as new ambient space

$\Longrightarrow$ no longer working in $\mathbb{K}[\underline{x}]$,

Local System of Parameters

we have seen:

$$\mathrm{ord}_p(I) = 1 \Longrightarrow \exists h \in I : V(h) \text{ smooth and } X \subset V(h) \text{ near } p$$

Choose $V(h)$ locally at p as new ambient space

$\Longrightarrow$ no longer working in $\mathbb{K}[\underline{x}]$,
but locally at p at least isomorphic
to $\mathbb{K}[[y_1, \ldots, y_{n-1}]]$ (Cohen structure theorem)

Local System of Parameters

we have seen:

$$\mathrm{ord}_p(I) = 1 \Longrightarrow \exists h \in I : V(h) \text{ smooth and } X \subset V(h) \text{ near } p$$

Choose $V(h)$ locally at p as new ambient space

$\Longrightarrow$ no longer working in $\mathbb{K}[\underline{x}]$,
but locally at p at least isomorphic
to $\mathbb{K}[[y_1, \ldots, y_{n-1}]]$ (Cohen structure theorem)

Transfer to $(\mathbb{K}[x_1, \ldots, x_n]/\langle h \rangle)_g$ possible

Subsequent Entry of ν^*

Sketch of approach:

- Choose appropriate $V(h)$ und $D(g)$ covering X (g suitably chosen derivative of h)
- on each $D(g)$ consider: $X \cap D(g) \subset V(h) \cap D(g)$
- repeat previous construction (slightly modified):
  - $\dim(V(h)) = n - 1$, drop in codimension
  - derivatives w.r.t. local systems of parameters in $V(h) \cap D(g)$
  - provides covering of $X \cap D(g)$ by new $V(h, h_{neu}) \cap D(g_{neu})$
- iterate

Algorithmic Aspects

Observation:

- number of charts increases, while codimension drops

Algorithmic Aspects

Observation:

- number of charts increases, while codimension drops
- sufficiently small codimension makes Jacobian criterion feasable

Algorithmic Aspects

Observation:

- number of charts increases, while codimension drops
- sufficiently small codimension makes Jacobian criterion feasable
- in practice: hybrid approach using descent in ambient space and Jacobian criterion

Algorithmic Aspects

Observation:

- number of charts increases, while codimension drops
- sufficiently small codimension makes Jacobian criterion feasable
- in practice: hybrid approach using descent in ambient space and Jacobian criterion

Challenges:

- choose large neighbourhoods for new ambient space
- keep number of charts low/use easily computable charts
- consistent choice of local system of parameters
- derivatives w.r.t. a system of parameters

Parallel, but not embarassingly parallel

Challenges call for parallel approach:

- choose large neighbourhoods for new ambient space
  $\Longrightarrow$ check whether everything is covered

Parallel, but not embarassingly parallel

Challenges call for parallel approach:

- choose large neighbourhoods for new ambient space
  $\Longrightarrow$ check whether everything is covered
- keep number of charts low/use easily computable charts
  $\Longrightarrow$ let fastest charts win

Parallel, but not embarassingly parallel

Challenges call for parallel approach:

- choose large neighbourhoods for new ambient space
  $\Longrightarrow$ check whether everything is covered
- keep number of charts low/use easily computable charts
  $\Longrightarrow$ let fastest charts win
- derivatives w.r.t. a system of parameters
  $\Longrightarrow$ potentially expensive computations

We really need a powerful parallelization environment:

- with good scheduling
- with good resource management

Parallel, but not embarassingly parallel

Challenges call for parallel approach:

- choose large neighbourhoods for new ambient space
  $\Longrightarrow$ check whether everything is covered
- keep number of charts low/use easily computable charts
  $\Longrightarrow$ let fastest charts win
- derivatives w.r.t. a system of parameters
  $\Longrightarrow$ potentially expensive computations

We really need a powerful parallelization environment:

- with good scheduling
- with good resource management
- without too much technical overhead

Parallel, but not embarassingly parallel

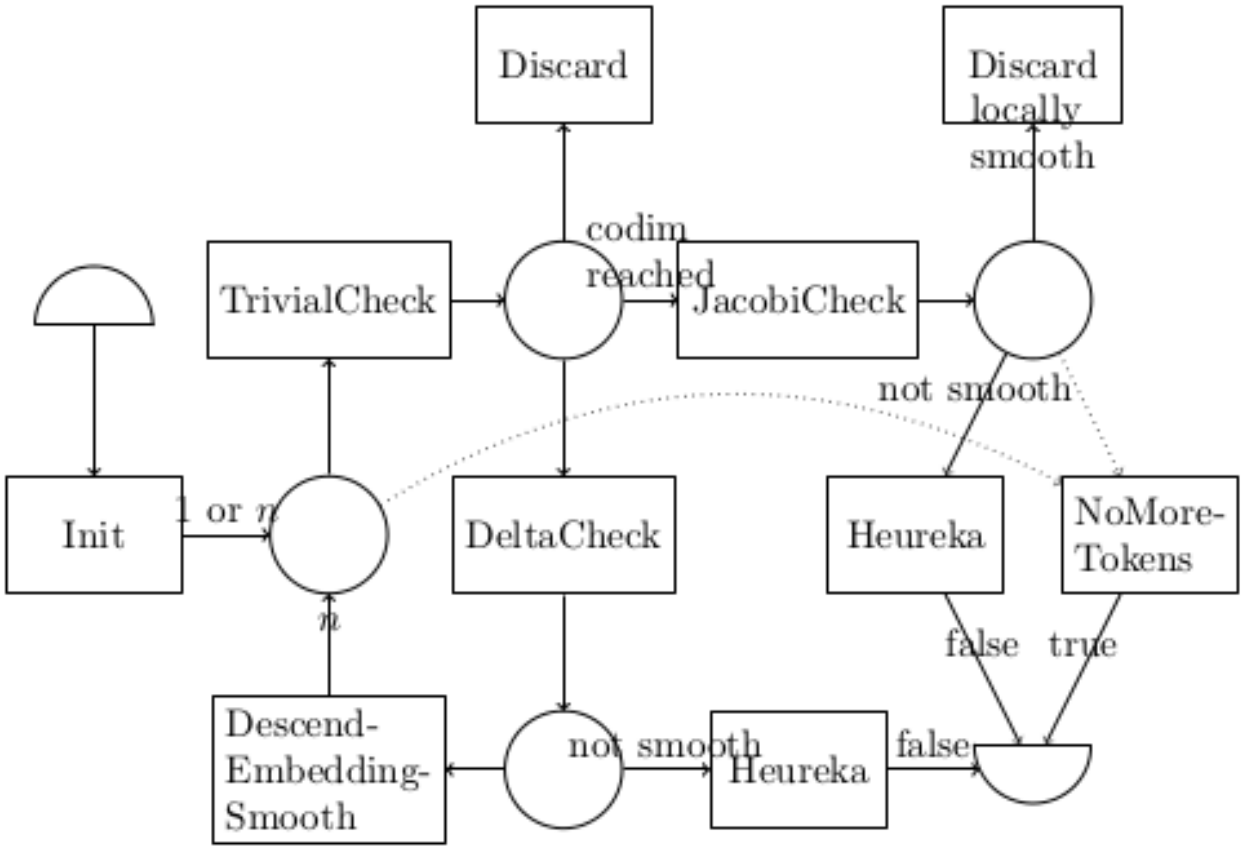
Challenges call for parallel approach:

- choose large neighbourhoods for new ambient space
  $\Longrightarrow$ check whether everything is covered
- keep number of charts low/use easily computable charts
  $\Longrightarrow$ let fastest charts win
- derivatives w.r.t. a system of parameters
  $\Longrightarrow$ potentially expensive computations

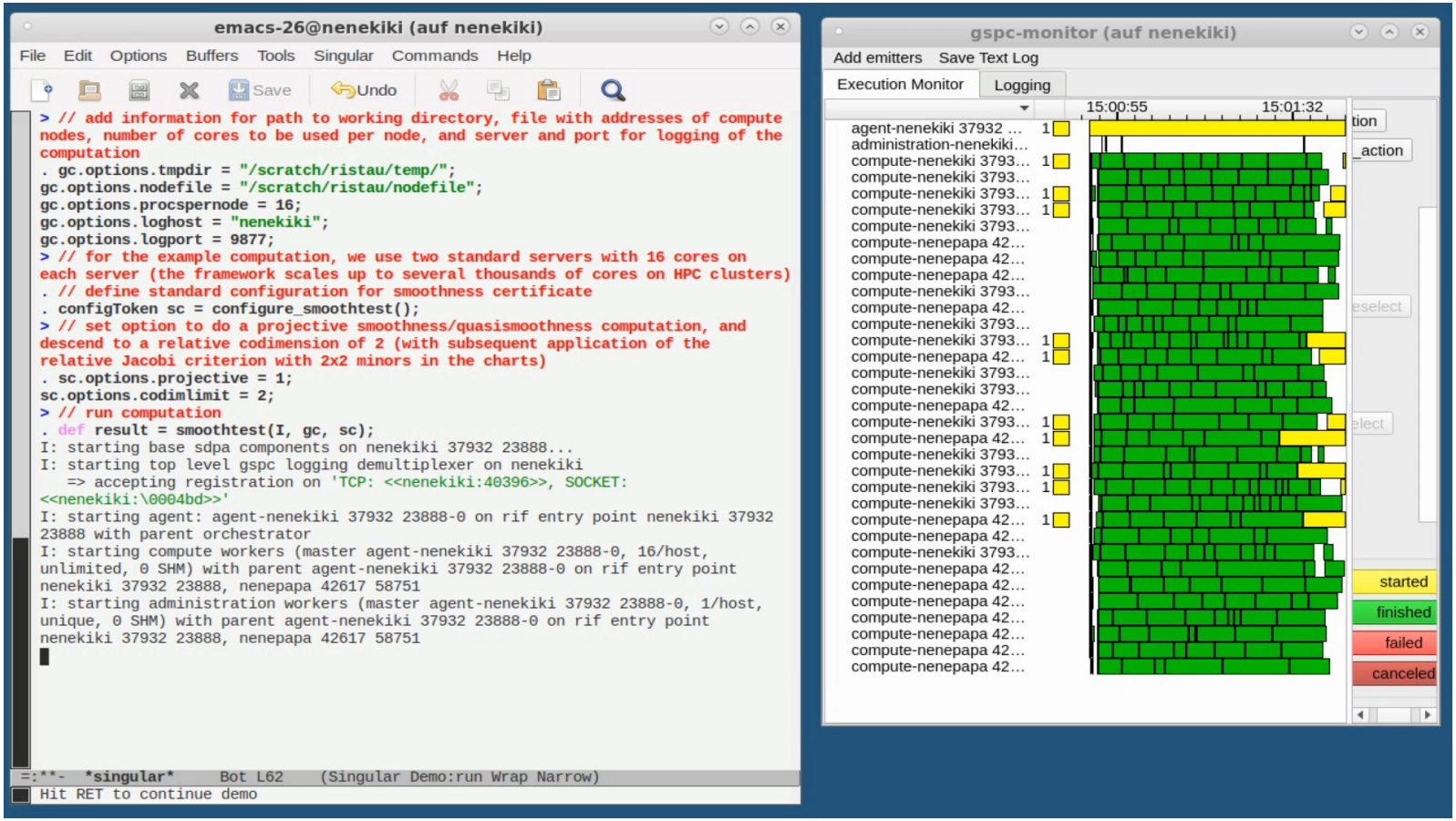
We really need a powerful parallelization environment:

- with good scheduling
- with good resource management
- without too much technical overhead
- robust against huge variations in computing time and size

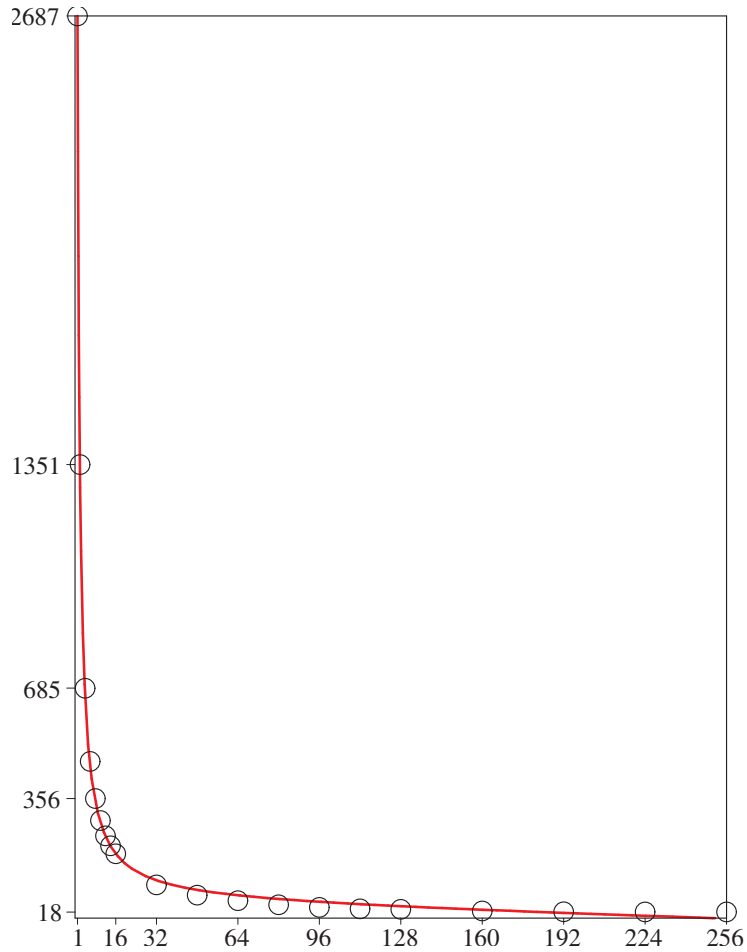
The Algorithm as a Petri-Net



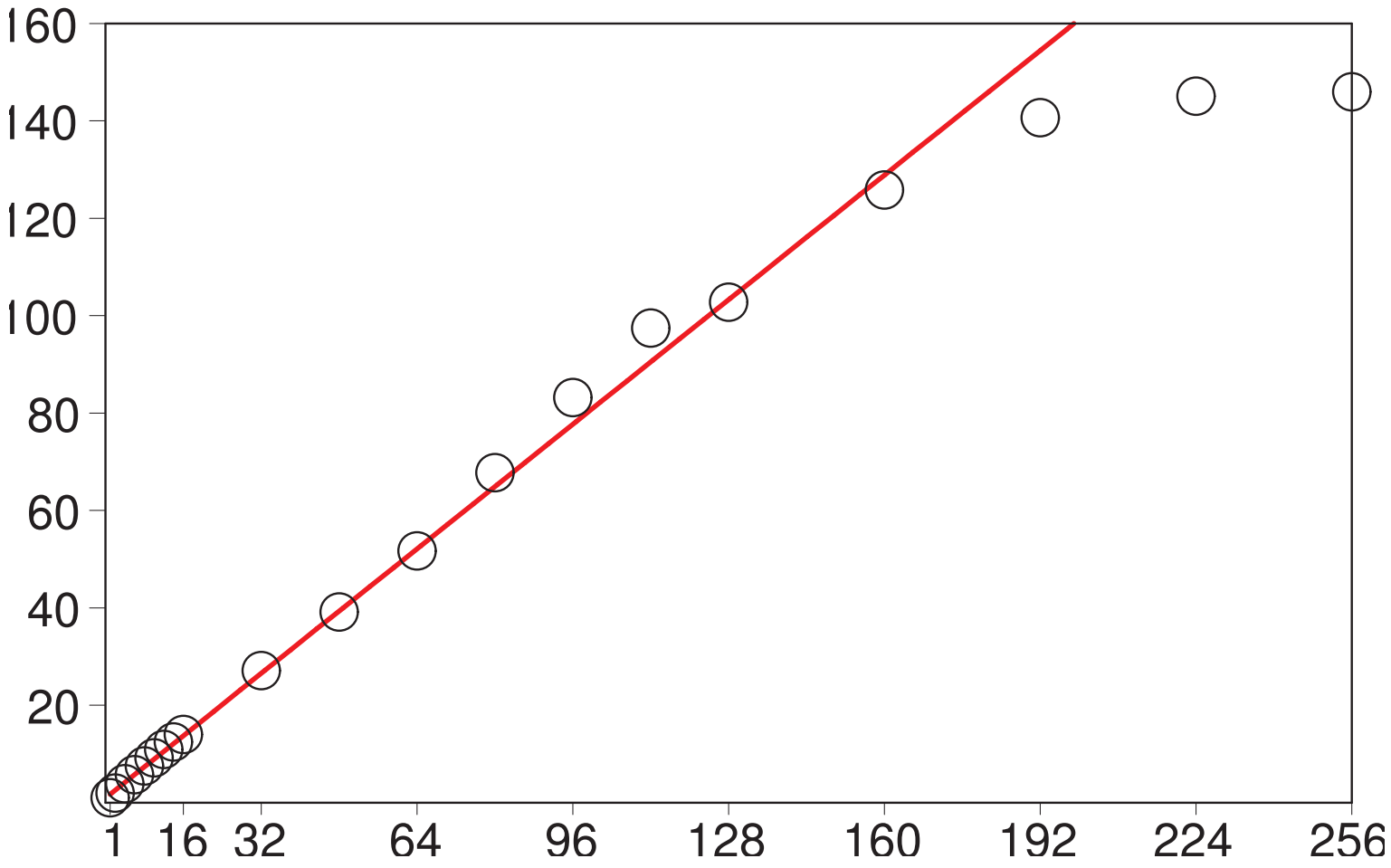
A screenshot



Campedelli-Fläche – Timings



Campedelli-Fläche – Speed-Up versus cores



Links and References

- https://github.com/singular-gpispace/framework
- https://www.mathematik.uni-kl.de/~boehm/singulargpispace
- J. Böhm, W. Decker, A. Frühbis-Krüger, F.-J. Pfreundt, M. Rahn, L. Ristau: *Towards Massively Parallel Computations In Algebraic Geometry*, to appear in Foundations of Computational Mathematics (2021)
- W. Decker, G.-M. Greuel, G. Pfister, H. Schönemann: SINGULAR – A computer algebra system for polynomial computations http://www.singular.uni-kl.de
- Fraunhofer ITWM / F.-J. Pfreundt, M. Rahn, et al.: GPI-space http://www.gpi-space.de

Thank you!