PeACE

Polynômes et Applications via Calculs Efficaces

Matías R. Bender

Since 2019 - Postdoctoral researcher

Inst. für Mathematik - Technische Univ. Berlin

Mentored by P. Bürgisser

2016 - 2019 - Ph.D. in Informatics

LIP6 - Sorbonne Université

Algorithms for sparse polynomial systems: Gröbner basis and resultants

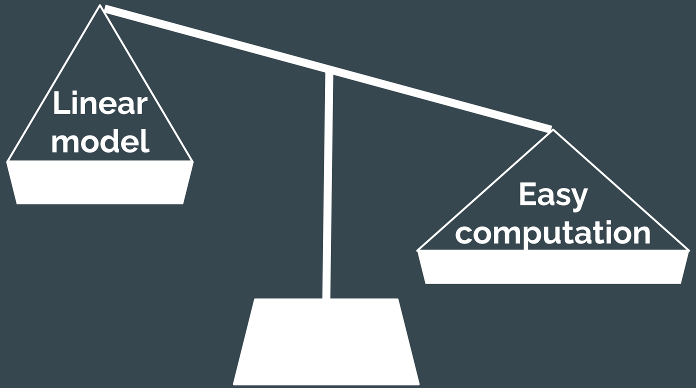
Supervised by J.-C. Faugère & E. Tsigaridas

(Qualified in sections 25, 26, and 27 for MCF)

2015 - M.Sc. in Computer Science

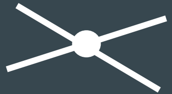
DC - FCEN - Universidad de Buenos Aires

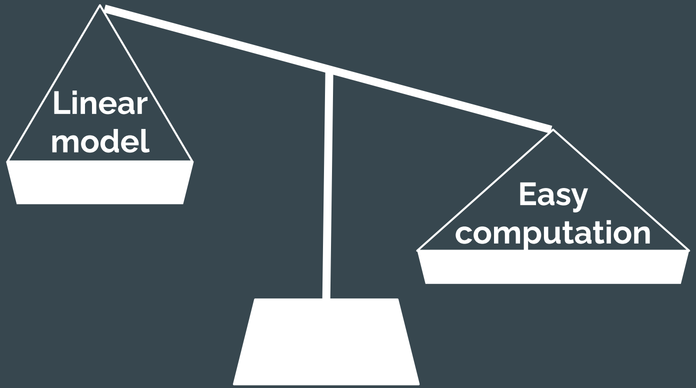
2014 - Exchange - Univ. Autónoma de Madrid



Linear algebra

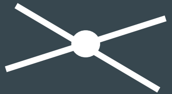
$$\left\{ \begin{array}{c} 3 \cdot x + 2 \cdot y = 5 \\ 4 \cdot x - 3 \cdot y = 2 \end{array} \right\}$$

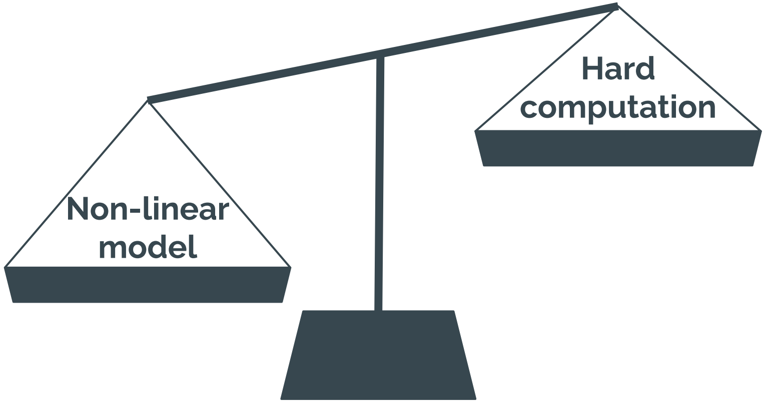




Linear algebra

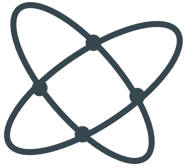
$$\begin{cases} 3 \cdot x + 2 \cdot y = 5 \\ 4 \cdot x - 3 \cdot y = 2 \end{cases}$$

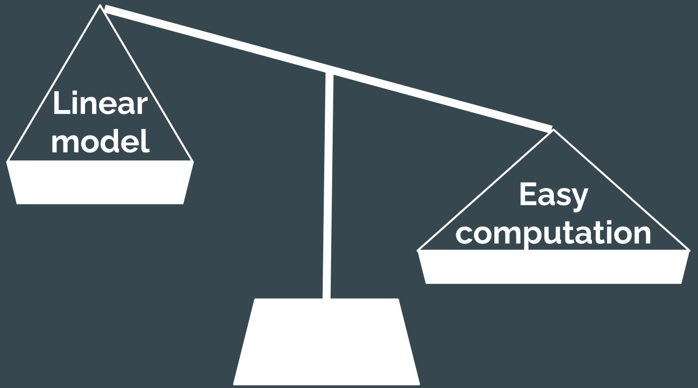




Non-linear algebra

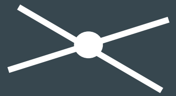
$$\begin{cases} 3 \cdot x^2 + 4 \cdot x \cdot y + 2 \cdot y^2 + x + 2 \cdot y = 5 \\ 6 \cdot x^2 + 2 \cdot x \cdot y + 3 \cdot y^2 - x + 3 \cdot y = 9 \end{cases}$$

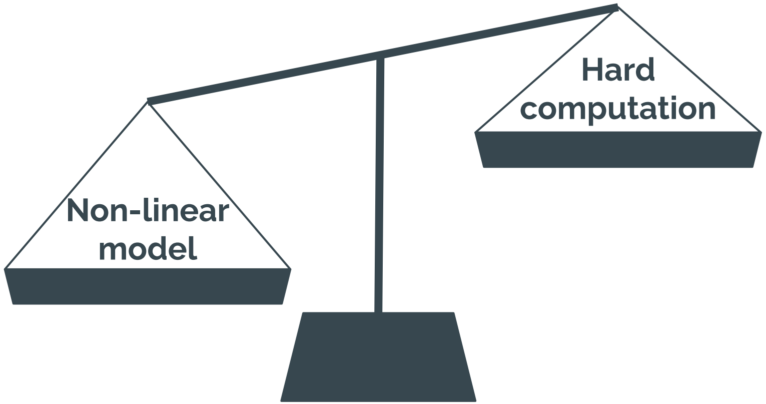




Linear algebra

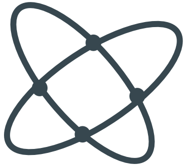
$$\left\{ \begin{array}{l} 3 \cdot x + 2 \cdot y = 5 \\ 4 \cdot x - 3 \cdot y = 2 \end{array} \right\}$$





Non-linear algebra

$$\left\{ \begin{array}{l} 3 \cdot x^2 + 4 \cdot x \cdot y + 2 \cdot y^2 + x + 2 \cdot y = 5 \\ 6 \cdot x^2 + 2 \cdot x \cdot y + 3 \cdot y^2 - x + 3 \cdot y = 9 \end{array} \right\}$$



Pragmatic approach: Use non-linear model, if we can compute with it

My objective: Study trade-off for non-linear systems in practice

Computing with polynomials

Symbolic paradigm (Since PhD)

Numerical paradigm (Since Postdoc)

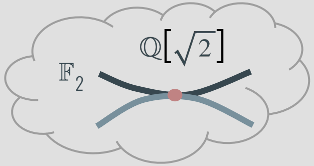
Computing with polynomials

Symbolic paradigm (Since PhD)

Exact computations

$$\sqrt{5 + 2\sqrt{6}} - \sqrt{3} = \sqrt{2}$$

- Finite fields
- Algebraic extensions
- Degenerate situations



Numerical paradigm (Since Postdoc)

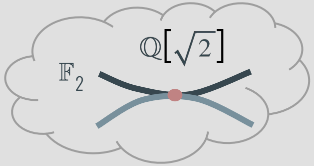
Computing with polynomials

Symbolic paradigm (Since PhD)

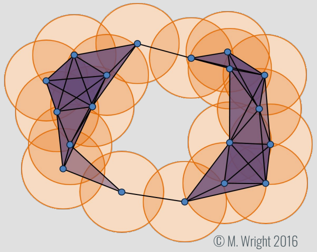
Exact computations

$$\sqrt{5 + 2\sqrt{6}} - \sqrt{3} = \sqrt{2}$$

- Finite fields
- Algebraic extensions
- Degenerate situations



Cryptography



Topological data analysis

Gröbner bases

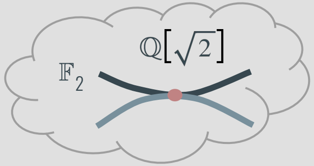
Numerical paradigm (Since Postdoc)

Computing with polynomials

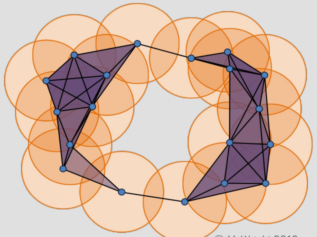
Symbolic paradigm (Since PhD)

Exact computations $\sqrt{5 + 2\sqrt{6}} - \sqrt{3} = \sqrt{2}$

- Finite fields
- Algebraic extensions
- Degenerate situations



Cryptography



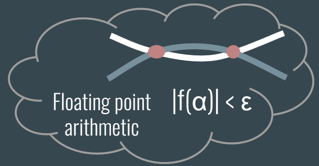
Topological data analysis

Gröbner bases

Numerical paradigm (Since Postdoc)

Numerical manipulations $\sqrt{5 + 2\sqrt{6}} - \sqrt{3} \approx 1.414214$

- Inexact input
- Finite- and multi-precision
- Approximate solutions

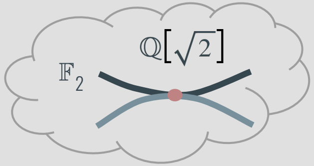


Computing with polynomials

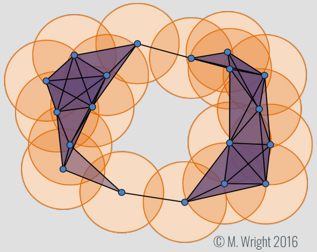
Symbolic paradigm (Since PhD)

Exact computations $\sqrt{5+2\sqrt{6}}-\sqrt{3}=\sqrt{2}$

- Finite fields
- Algebraic extensions
- Degenerate situations



Cryptography



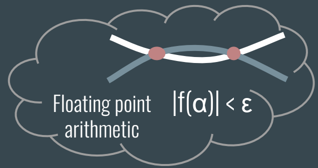
Topological data analysis

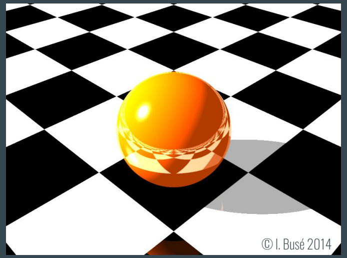
Gröbner bases

Numerical paradigm (Since Postdoc)

Numerical manipulations $\sqrt{5+2\sqrt{6}}-\sqrt{3}\approx 1.414214$

- Inexact input
- Finite- and multi-precision
- Approximate solutions





Geometric modeling

Computer vision

Symbolic-numeric algorithms, homotopy continuation

Computing with polynomials ➡ My research

Symbolic paradigm (Since PhD)

Publications

Journals 1 PhD

Proceedings of International Conferences 4 PhD + 1 Postdoc

Distinguished student author award, ISSAC 2016

Preprints 1 Postdoc

Software

Muphasa (C++), Classifier-construction (Python)

Numerical paradigm (Since Postdoc)

Publications

Journals 2 Postdoc

New! Accepted paper in Mathematics of Computations - AMS

Preprints 1 Postdoc

Software

EigenvalueSolver.jl (Julia), sylvesterMEP (Matlab)

New! Invited tutorial in ISSAC 2022 (flagship conference on symbolic and algebraic computations)

Co-authors

J.-C. Faugère, E. Tsigaridas (INRIA Paris, advisors)
A. Mantzaflaris (INRIA Sophia-Antipolis)
L. Perret (LIP6, Sorbonne Université)

C. Haase, R. Schwieger, H. Siebert (FU Berlin)
S. Telen (Max Planck Inst. for Math. in the Sciences)

O. Gäfvert (Oxford)

M. Lesnick (SUNY Albany)

Simplified dictionary of algebraic tools

Simplified dictionary of algebraic tools

Gröbner bases
(arbitrary systems)

- Main algorithmic tool to manipulate polynomials exactly.
- Non-linear analog of row echelon form.
- They can solve polynomials or certify if no solutions.

Simplified dictionary of algebraic tools

Gröbner bases
(arbitrary systems)

- Main algorithmic tool to manipulate polynomials exactly.
- Non-linear analog of row echelon form.
- They can solve polynomials or certify if no solutions.

Multiplication map
(only finite solutions)

- Solve system by computing eigenvalues.
- Non-linear analog of companion matrix of univariate polynomial.
- Compatible with numerical computations!

Solving polynomial systems via linearization

Polynomial system $(f_1, \dots, f_n)$ $\longrightarrow$

- Gröbner bases
- Multiplication map

Analogy for linear systems

Linear system $(f_1, \dots, f_n)$ $\longrightarrow$

- Exact solution
- Approximated solution

Solving polynomial systems via linearization

Polynomial system $(f_1,\dots,f_n)$ → Linearization (structured matrix) $(g_1,\dots,g_n) \rightarrow \Sigma\, g_i f_i$, $\text{degree}(g_i f_i) < d$ → { Gröbner bases; Multiplication map }

Analogy for linear systems

Linear system $(f_1,\dots,f_n)$ → Matrix $(c_1,\dots,c_n) \rightarrow \Sigma\, c_i f_i$, scalar c_i → { Exact solution; Approximated solution }

Solving polynomial systems via linearization

Polynomial system $(f_1,\ldots,f_n)$ → Linearization (structured matrix) $(g_1,\ldots,g_n) \to \Sigma\, g_i f_i$, $\mathrm{degree}(g_i f_i) < d$ → Gröbner bases / Multiplication map

(Degree d determines size, depends on system
For almost all dense systems: degree known)

Analogy for linear systems

Linear system $(f_1,\ldots,f_n)$ → Matrix $(c_1,\ldots,c_n) \to \Sigma\, c_i f_i$, scalar c_i → Exact solution / Approximated solution

Solving polynomial systems via linearization

Polynomial system $(f_1,\dots,f_n)$ → Linearization (structured matrix) $(g_1,\dots,g_n) \to \Sigma\, g_i f_i$, degree$(g_i f_i) < d$ $\xrightarrow{\text{Exact computation}}$ Gröbner bases

(Degree d determines size, depends on system
For almost all dense systems: degree known)

Multiplication map

Analogy for linear systems

Linear system $(f_1,\dots,f_n)$ → Matrix $(c_1,\dots,c_n) \to \Sigma\, c_i f_i$, scalar c_i $\xrightarrow{\text{Exact computation}}$ Exact solution

Approximated solution

Solving polynomial systems via linearization

(Important: speed up - exploit matrix structure)

Polynomial system $(f_1,\dots,f_n)$ → Linearization (structured matrix) $(g_1,\dots,g_n) \to \Sigma\, g_i f_i$, $\text{degree}(g_i f_i) < d$ → *Exact computation* → Gröbner bases

(e.g., Buchberger, Faugère, Lazard, Mora, Stillman...)

Multiplication map

(Degree d determines size, depends on system
For almost all dense systems: degree known)

Analogy for linear systems

Linear system $(f_1,\dots,f_n)$ → Matrix $(c_1,\dots,c_n) \to \Sigma\, c_i f_i$, scalar c_i → *Exact computation* → Exact solution

Approximated solution

Solving polynomial systems via linearization

(Important: speed up - exploit matrix structure)

Polynomial system $(f_1,\ldots,f_n)$ → Linearization (structured matrix) $(g_1,\ldots,g_n) \rightarrow \Sigma\, g_i f_i$ degree$(g_i f_i) < d$

(Degree d determines size, depends on system
For almost all dense systems: degree known)

Exact computation → Gröbner bases

(e.g., Buchberger, Faugère, Lazard, Mora, Stillman...)

Numerical linear algebra → Multiplication map

Analogy for linear systems

Linear system $(f_1,\ldots,f_n)$ → Matrix $(c_1,\ldots,c_n) \rightarrow \Sigma\, c_i f_i$ scalar c_i

Exact computation → Exact solution

Numerical linear algebra → Approximated solution

Solving polynomial systems via linearization

(Important: speed up - exploit matrix structure)

Polynomial system $(f_1,\ldots,f_n)$ → Linearization (structured matrix) $(g_1,\ldots,g_n) \rightarrow \Sigma\, g_i f_i$, $\text{degree}(g_i f_i) < d$

(Degree d determines size, depends on system
For almost all dense systems: degree known)

Exact computation → Gröbner bases

(e.g., Buchberger, Faugère, Lazard, Mora, Stillman...)

Numerical linear algebra → Multiplication map

(e.g., Canny, Emiris, Möller, Mourrain, Stetter...)

(Important: stability)

Analogy for linear systems

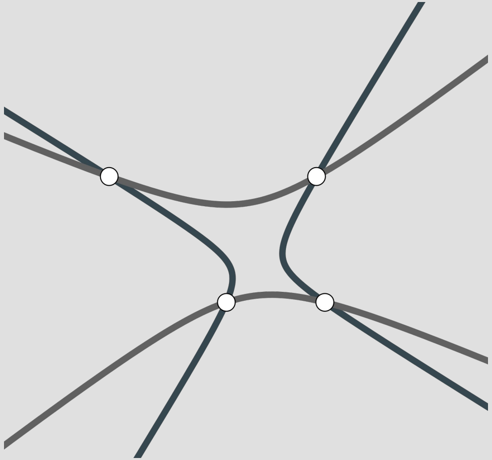
Linear system $(f_1,\ldots,f_n)$ → Matrix $(c_1,\ldots,c_n) \rightarrow \Sigma\, c_i f_i$, scalar c_i

Exact computation → Exact solution

Numerical linear algebra → Approximated solution

In practice, polynomials have structure

Generic system



$$\left\{ \begin{array}{l} x^2 + x \cdot y - y^2 - x - 2 \cdot y - 1 = 0 \\ -x^2 - x \cdot y + 3 \cdot y^2 + x + y - 7 = 0 \end{array} \right\}$$

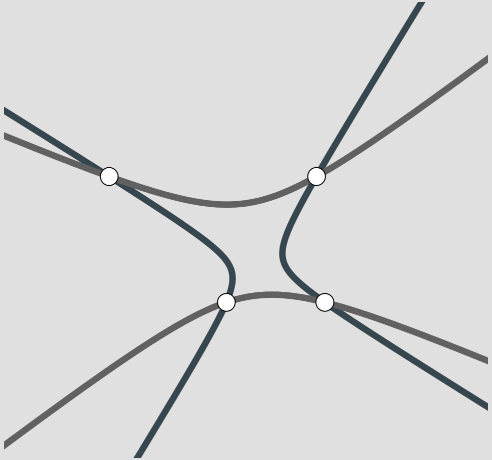
Generalized eigenvalue problem

$$\begin{pmatrix} 20 & 6 \\ -12 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = y \begin{pmatrix} -1 & -4 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}$$

Sparsity: Polynomial with a few monomials

In practice, polynomials have structure

Generic system



$$\begin{cases} x^2 + x \cdot y - y^2 - x - 2 \cdot y - 1 = 0 \\ -x^2 - x \cdot y + 3 \cdot y^2 + x + y - 7 = 0 \end{cases}$$

Generalized eigenvalue problem

$$\begin{cases} 0 \cdot x^2 - 4 \cdot x \cdot y - 0 \cdot y^2 + 6 \cdot x - y + 20 = 0 \\ 0 \cdot x^2 + 2 \cdot x \cdot y - 0 \cdot y^2 + 3 \cdot x + 4 \cdot y - 12 = 0 \end{cases}$$

Sparsity: Polynomial with a few monomials

Solving sparse polynomial systems

(Speed up - exploit matrix structure)

Sparse
Polynomial system $(f_1,\ldots,f_n)$ $\longrightarrow$ Linearization $(g_1,\ldots,g_n) \rightarrow \Sigma\, g_i f_i$ degree$(g_i f_i) < d$

(Degree determines size)
(For almost all dense systems, known degree)

Exact computation $\longrightarrow$ Gröbner bases

Numerical linear algebra $\longrightarrow$ Multiplication map

(Stability)

Solving sparse polynomial systems

(Speed up - exploit matrix structure)

Sparse
Polynomial system $(f_1,\dots,f_n)$ → Linearization $(g_1,\dots,g_n) \longrightarrow \Sigma\, g_i f_i$, degree$(g_i f_i) < d$

Exact computation → Gröbner bases

Numerical linear algebra → Multiplication map

(Degree determines size)
~~(For almost all dense systems, known degree)~~

(Stability)

We don’t know the maximal degree
General bound: double exponential!

Solving sparse polynomial systems

- Study special systems, e.g., bi- or weighted homogeneous.
- Discover the structure of the matrix "on the fly".

Matrix has different structure
we can not exploit - inefficient

~~(Speed up - exploit matrix structure)~~

Sparse
Polynomial system $(f_1, \dots, f_n)$ $\longrightarrow$ Linearization $(g_1, \dots, g_n) \longrightarrow \sum g_i f_i$ degree$(g_i f_i) < d$

Exact computation $\longrightarrow$ Gröbner bases

Numerical linear algebra $\longrightarrow$ Multiplication map

(Degree determines size)
~~(For almost all dense systems, known degree)~~

We don't know the maximal degree
General bound: double exponential!

(Stability)

Solving sparse polynomial systems

- Study special systems, e.g., bi- or weighted homogeneous.
- Discover the structure of the matrix "on the fly".

Matrix has different structure
we can not exploit - inefficient

~~(Speed up - exploit matrix structure)~~

Sparse
Polynomial system $(f_1,\ldots,f_n)$ →

Linearization
$(g_1,\ldots,g_n) \to \Sigma\, g_i f_i$
$\mathrm{degree}(g_i f_i) < d$

Exact computation → Gröbner bases

Numerical linear algebra → Multiplication map

(Degree determines size)
~~(For almost all dense systems, known degree)~~

We don't know the maximal degree
General bound: double exponential!

~~(Stability)~~

Solution at infinity - ill-conditioned

- Sparse resultant.

Sparse

Polynomial
system
$(f_1,\ldots,f_n)$

$\longrightarrow$

Sparse linearization
$(g_1,\ldots,g_n) \to \sum g_i f_i$
sparse g_i

Exact computation $\longrightarrow$ Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra $\longrightarrow$ Approximation
[BT21+] [BT22]

Contributions

- during PhD
- during Postdoc

Sparse

Polynomial system $(f_1,\ldots,f_n)$

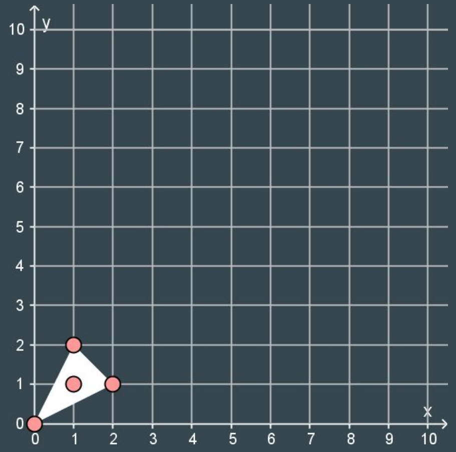
Sparse linearization $(g_1,\ldots,g_n) \rightarrow \Sigma\, g_i f_i$ sparse g_i

(e.g., Canny, Cox, D'Andrea, Dickenstein, Emiris, Sturmfels...)

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Mathematical foundation ⟶ Toric geometry



$1 + x\,y + x^2 y + x\,y^2 \in \mathbb{C}[x,y]$

Contributions

- during PhD
- during Postdoc

Sparse

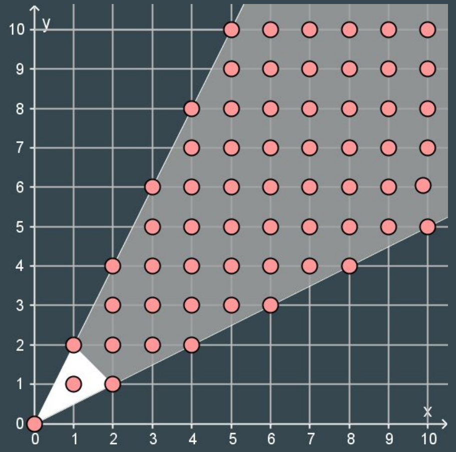
Polynomial system $(f_1,\dots,f_n)$ → Sparse linearization $(g_1,\dots,g_n) \to \Sigma\, g_i f_i$ sparse g_i

(e.g., Canny, Cox, D'Andrea, Dickenstein, Emiris, Sturmfels...)

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Mathematical foundation → Toric geometry



$$1 + x\,y + x^2\,y + x\,y^2 \in \mathbb{C}\left[x\,y, x^2\,y, x\,y^2\right] \subseteq \mathbb{C}[x,y]$$

Contributions

- during PhD
- during Postdoc

(Koszul-F5 criterion: speed up - exploit structure of sparse linearization)

Sparse

Polynomial system $(f_1,\dots,f_n)$ → Sparse linearization $(g_1,\dots,g_n) \rightarrow \sum g_i f_i$ sparse g_i

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Contributions

- during PhD
- during Postdoc

(Size matrix $\longrightarrow$ sparsity of system,
Minkowski sum of Newton polytopes)

(Koszul-F5 criterion: speed up -
exploit structure of sparse linearization)

Sparse
Polynomial system $(f_1,\dots,f_n)$

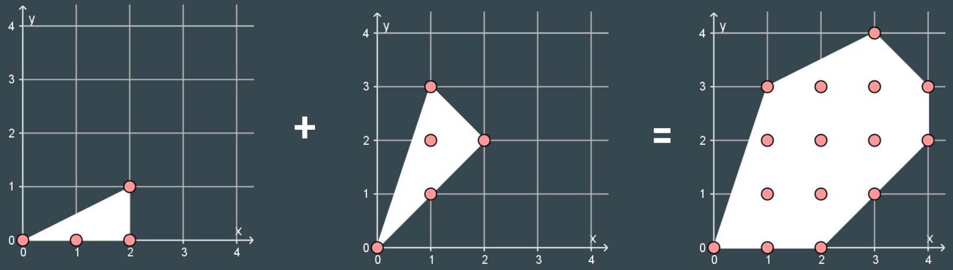
Sparse linearization
$(g_1,\dots,g_n) \longrightarrow \Sigma\, g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]



$1 + x + x^2 + x\,y$ $\quad$ $1 + x\,y + x\,y^2 + x\,y^3 + x^2\,y^2$

Contributions

- during PhD
- during Postdoc

Improved bounds for special systems, e.g., multi-homogeneous. ⟵ (Size matrix ⟶ sparsity of system, Minkowski sum of Newton polytopes)

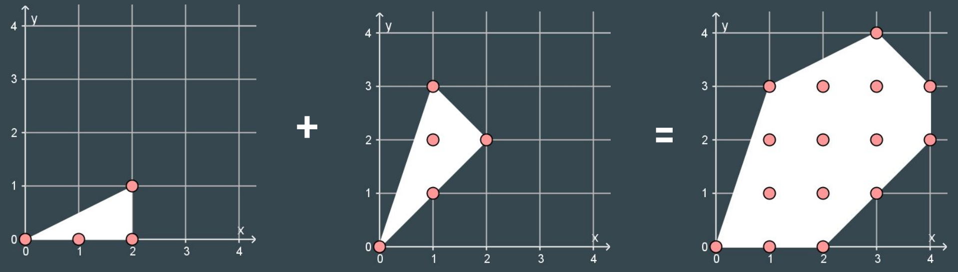
(Koszul-F5 criterion: speed up - exploit structure of sparse linearization)

Sparse

Polynomial system $(f_1,\ldots,f_n)$ ⟶ Sparse linearization $(g_1,\ldots,g_n) \longrightarrow \Sigma\, g_i f_i$ sparse g_i

Exact computation ⟶ Gröbner bases [BFT18] [BFT19]

Numerical linear algebra ⟶ Approximation [BT21+] [BT22]



$1 + x + x^2 + x\,y$ + $1 + x\,y + x\,y^2 + x\,y^3 + x^2\,y^2$

Contributions

- during PhD
- during Postdoc

Improved bounds for special systems, e.g., multi-homogeneous. ⟵ (Size matrix ⟶ sparsity of system, Minkowski sum of Newton polytopes)

(Koszul-F5 criterion: speed up - exploit structure of sparse linearization)

Sparse

Polynomial system $(f_1,\dots,f_n)$ ⟶ Sparse linearization $(g_1,\dots,g_n) \rightarrow \Sigma\, g_i f_i$ sparse g_i

Exact computation ⟶ Gröbner bases [BFT18] [BFT19]

Numerical linear algebra ⟶ Approximation [BT21+] [BT22]

(Reduce problem to eigenvalue computations, avoid problems at infinity -> give *small* coordinates)

Contributions

- during PhD
- during Postdoc

Improved bounds for special systems, e.g., multi-homogeneous. ⟵ (Size matrix ⟶ sparsity of system, Minkowski sum of Newton polytopes)

(Koszul-F5 criterion: speed up - exploit structure of sparse linearization)

Sparse
Polynomial system $(f_1,\dots,f_n)$ ⟶ Sparse linearization $(g_1,\dots,g_n) \longrightarrow \Sigma\, g_i f_i$ sparse g_i

Exact computation ⟶ Gröbner bases [BFT18] [BFT19]

Numerical linear algebra ⟶ Approximation [BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

(Reduce problem to eigenvalue computations, avoid problems at infinity -> give *small* coordinates)

(Beats homotopy continuation for overdetermined sys)

Contributions

- during PhD
- during Postdoc

(Size matrix ⟶ sparsity of system,
Minkowski sum of Newton polytopes)

Sparse
Polynomial
system
$(f_1,\dots,f_n)$

Sparse linearization
$(g_1,\dots,g_n) \longrightarrow \Sigma\, g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Contributions

- during PhD
- during Postdoc

(Size matrix ⟶ sparsity of system,
Minkowski sum of Newton polytopes)

Sparse
Polynomial
system
$(f_1,\ldots,f_n)$

Sparse linearization
$(g_1,\ldots,g_n) \longrightarrow \Sigma\, g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Sparse
multilinear

Smaller linearization
[BFMT18]

Contributions

- during PhD
- during Postdoc

(Size matrix → sparsity of system,
Minkowski sum of Newton polytopes)

Sparse
Polynomial system $(f_1,\dots,f_n)$

Sparse linearization
$(g_1,\dots,g_n) \to \sum g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Sparse multilinear

Smaller linearization
[BFMT18]

Applications

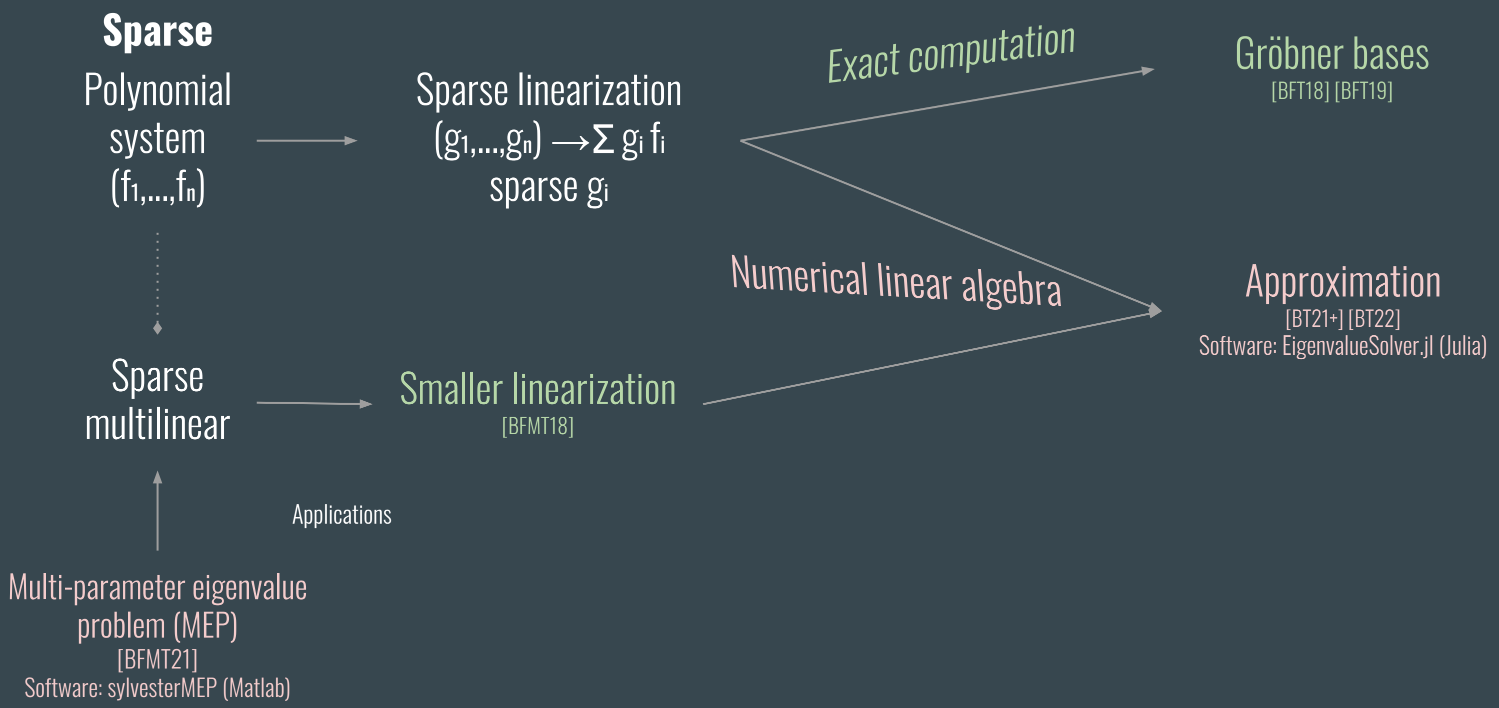
Multi-parameter eigenvalue problem (MEP)
[BFMT21]
Software: sylvesterMEP (Matlab)

Generalization of *Generalized eigenvalue problem.*

Applications to Sturm-Liouville theory [Atkinson '72, Volkmer '88]

Contributions

- during PhD
- during Postdoc



Contributions

Sparse

Polynomial system $(f_1,\ldots,f_n)$

Sparse linearization
$(g_1,\ldots,g_n) \rightarrow \Sigma\, g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Sparse multilinear

Smaller linearization
[BFMT18]

Applications

Multi-parameter eigenvalue problem (MEP)
[BFMT21]
Software: sylvesterMEP (Matlab)

Symmetric tensor decomposition

(binary forms)
[BFPT16] [BFPT21]

Quasi-optimal algorithm to decompose binary forms (improves Common, Dür, Mourrain, Helmke, Sylvester)

$$-4704\,x^3 + 3576\,x^2 y - 912\,x\,y^2 + 78\,y^3 = \frac{1}{6}(-26\,x + 7\,y)^3 - \frac{1}{6}(22\,x - 5\,y)^3$$

Contributions

Sparse

Polynomial system $(f_1,\dots,f_n)$

Sparse linearization
$(g_1,\dots,g_n) \rightarrow \Sigma\, g_i f_i$
sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Sparse multilinear

Smaller linearization
[BFMT18]

Applications

Multi-parameter eigenvalue problem (MEP)
[BFMT21]
Software: sylvesterMEP (Matlab)

Topological data analysis (TDA)
(multi-param. persistent homology, MPH)
[BGL22+]
Software: Muphasa (C++)

Can compute MPH using Gröbner bases (Carlsson, Singh & Zomorodian 2009)
but, in practice, available implementations can not compute MPH!

Dedicated algorithm and implementation to compute MPH using Gröbner bases

Symmetric tensor decomposition
(binary forms)
[BFPT16] [BFPT21]

Contributions

Sparse

Polynomial system $(f_1,\dots,f_n)$

Sparse linearization $(g_1,\dots,g_n) \to \Sigma\, g_i f_i$ sparse g_i

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]
Software: EigenvalueSolver.jl (Julia)

Sparse multilinear

Smaller linearization
[BFMT18]

Applications

Multi-parameter eigenvalue problem (MEP)
[BFMT21]
Software: sylvesterMEP (Matlab)

Topological data analysis (TDA)
(multi-param. persistent homology, MPH)
[BGL22+]
Software: Muphasa (C++)

Symmetric tensor decomposition
(binary forms)
[BFPT16] [BFPT21]

Computational biology
(classifiers on Boolean networks)
[SBSH20]
Software: Classifier-construction (Python)

Contributions

Sparse
Polynomial
system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse
multilinear

Smaller linearization
[BFMT18]

Future work

Non-finite solutions
Compute Gröbner bases

Sparse
Polynomial
system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse
multilinear

Smaller linearization
[BFMT18]

Future work

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse
Polynomial system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse multilinear

Smaller linearization
[BFMT18]

Future work

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse
Polynomial
system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse
multilinear

Smaller linearization
[BFMT18]

Precision analysis
Relate the condition number of matrices to the conditioning of systems
(missing in dense case also!)

Future work

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse
Polynomial system → Sparse linearization

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18]

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems
(missing in dense case also!)

Future work

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse
Polynomial system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse multilinear

Smaller linearization
[BFMT18]

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n$ x $\mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems
(missing in dense case also!)

Future work

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse
Polynomial system

Sparse linearization

Exact computation

Gröbner bases
[BFT18] [BFT19]

Numerical linear algebra

Approximation
[BT21+] [BT22]

Sparse multilinear

Smaller linearization
[BFMT18]

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Future work

Future work

M. Skomra
M. Haiech

Structured systems
Use toric degenerations to treat them as sparse

T. Vaccon
P.-J. Spaenlehauer

Non-finite solutions
Compute Gröbner bases

T. Vaccon
P.-J. Spaenlehauer

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18] → Approximation

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

J.-B. Lasserre, V. Magron
S. Lazard, G. Moroz

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

M. Joldes, V. Magron
G. Moroz

Integration

- **LAAS - Toulouse**
  Laboratoire d'Analyse et d'Architecture des Systèmes
- **XLIM - Limoges**
  Institut de Recherche XLIM
- **LORIA - Nancy**
  Laboratoire Lorrain de recherche en Informatique et ses Applications

Integration

- LAAS (Toulouse)
- XLIM (Limoges)
- LORIA (Nancy)

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18]

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Future work

Future work

Integration
- LAAS (Toulouse)
- XLIM (Limoges)
- LORIA (Nancy)

Linear differential systems
(extension to Weyl algebras)
M. Barkatou, T. Cluzeau, M. Haiech, J.-A. Weil

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Sparse linearization $\xrightarrow{\text{Exact computation}}$ Gröbner bases [BFT18] [BFT19]

Sparse linearization $\xrightarrow{\text{Numerical linear algebra}}$ Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18] → Approximation

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Future work

Linear differential systems
(extension to Weyl algebras)

M. Barkatou, T. Cluzeau, M. Haiech, J.-A. Weil

Cryptography
(point-counting, PQC)

P. Gaborit
E. Thomé, P. Gaudry, A. Guillevic

Integration
- **LAAS** (Toulouse)
- **XLIM** (Limoges)
- **LORIA** (Nancy)

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Sparse multilinear → Smaller linearization [BFMT18]

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Linear differential systems
(extension to Weyl algebras)

M. Barkatou, T. Cluzeau, M. Haiech, J.-A. Weil

Cryptography
(point-counting, PQC)

P. Gaborit
E. Thomé, P. Gaudry, A. Guillevic

Integration
- **LAAS** (Toulouse)
- **XLIM** (Limoges)
- **LORIA** (Nancy)

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Sparse multilinear → Smaller linearization [BFMT18]

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Polynomial optimization
(Exploiting sparsity)
(Lasserre hierarchy)

D. Henrion, M. Korda, J.-B. Lasserre, V. Magron, M. Skomra
S. Naldi

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Future work

Linear differential systems
(extension to Weyl algebras)
M. Barkatou, T. Cluzeau, M. Haiech, J.-A. Weil

Cryptography
(point-counting, PQC)
P. Gaborit
E. Thomé, P. Gaudry, A. Guillevic

Integration
- **LAAS** (Toulouse)
- **XLIM** (Limoges)
- **LORIA** (Nancy)

Structured systems
Use toric degenerations to treat them as sparse

Non-finite solutions
Compute Gröbner bases

Efficient implementations
C/C++ implementation, incorporate other linear algebra speed-ups (F4)

Sparse Polynomial system → Sparse linearization

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18] → Approximation

Polynomial optimization
(Exploiting sparsity)
(Lasserre hierarchy)
D. Henrion, M. Korda, J.-B. Lasserre, V. Magron, M. Skomra
S. Naldi

Extension to other sparse
Multihomogenous systems, e.g., in $\mathbb{P}^n \times \mathbb{P}^m$

Precision analysis
Relate the condition number of matrices to the conditioning of systems (missing in dense case also!)

Geometric modelling
(Intersections of surfaces and curves)
L. Dupont, S. Lazard, G. Moroz, M. Pouget

Future work

Integration

- LAAS (Toulouse)
- XLIM (Limoges)
- LORIA (Nancy)

Linear differential systems

Cryptography

Structured systems

Non-finite solutions

Efficient implementations

Sparse Polynomial system → Sparse linearization

Sparse linearization —*Exact computation*→ Gröbner bases [BFT18] [BFT19]

Sparse linearization —Numerical linear algebra→ Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18] → Approximation

Polynomial optimization

Extension to other sparse

Precision analysis

Geometric modelling

Linear differential systems

Cryptography

Structured systems

Non-finite solutions

Efficient implementations

Mini CV

- Now - Postdoc
- 2019 - PhD
- 2015 - Master

Integration

- **LAAS** (Toulouse)
- **XLIM** (Limoges)
- **LORIA** (Nancy)

Sparse Polynomial system → Sparse linearization

Exact computation → Gröbner bases [BFT18] [BFT19]

Numerical linear algebra → Approximation [BT21+] [BT22]

Sparse multilinear → Smaller linearization [BFMT18]

Polynomial optimization

My research

- during PhD (symbolic)
- during Postdoc (numeric + apps)
- Future

Extension to other sparse

Precision analysis

Geometric modelling

Multi-param. eigenvalue problem
[BFMT21]
Software: sylvesterMEP (Matlab)

Symmetric tensor decomp. (binary forms)
[BFPT16] [BFPT21]

Topological data analysis
[BGL22+]
Software: Muphasa (C++)

Computational biology
[SBSH20]
Software: Classifier-construction (Python)

Publications

Journals	1 PhD + 2 Postdoc
Proceedings of International Conferences	4 PhD + 1 Postdoc
Preprints	2 Postdoc

Distinguished student author award, ISSAC 2016

Invited tutorial, ISSAC 2022 (on computer algebra)

Publications

Journals

[BT22] Toric Eigenvalue Methods for Solving Sparse Polynomial Systems. Matías R. Bender and Simon Telen. Mathematics of Computation - AMS, 2022. In press. (Recently accepted)

[BFMT21] Koszul-type determinantal formulas for families of mixed multilinear systems. Matías R. Bender, Jean-Charles Faugère, Angelos Mantzaflaris, and Elias Tsigaridas. SIAM Journal on Applied Algebra and Geometry, 2021.

[BFPT21] A nearly optimal algorithm to decompose binary forms. Matías R. Bender, Jean-Charles Faugère, Ludovic Perret, and Elias Tsigaridas. Journal of Symbolic Computation, 2021.

International conferences

[SBSH20] Classifier Construction in Boolean Networks Using Algebraic Methods. Robert Schwieger, Matías R. Bender, Heike Siebert, and Christian Haase. Computational Methods in Systems Biology, Lecture Notes in Computer Science, 2020.

[BFT19] Gröbner Basis over Semigroup Algebras: Algorithms and Applications for Sparse Polynomial Systems. Matías R. Bender, Jean-Charles Faugère, and Elias Tsigaridas. Proceedings of the 44th International Symposium on Symbolic and Algebraic Computation, 2019.

[BFMT18] Towards Mixed Gröbner Basis algorithms: the Multihomogeneous and Sparse case. Matías R. Bender, Jean-Charles Faugère, and Elias Tsigaridas. Proceedings of the 43th International Symposium on Symbolic and Algebraic Computation, 2018.

[BFT18] Bilinear systems with two supports: Koszul resultant matrices, eigenvalues, and eigenvectors. Matías R. Bender, Jean-Charles Faugère, Angelos Mantzaflaris, and Elias Tsigaridas. Proceedings of the 43th International Symposium on Symbolic and Algebraic Computation, 2018.

[BFPT16] A Superfast Algorithm to Decompose Binary Forms. Matías R. Bender, Jean-Charles Faugère, Ludovic Perret, and Elias Tsigaridas. Proceedings of the 41th International Symposium on Symbolic and Algebraic Computation, 2016 (Distinguished student author award)

Preprints

[BGL22+] Efficient computation of multiparameter persistence. Matías R. Bender, Oliver Gäfvert, and Michael Lesnick. [kth:diva-294302], to be submitted

[BT21+] Yet another eigenvalue algorithm for solving polynomial systems. Matías R. Bender and Simon Telen. [arXiv: 2105.08472], submitted.