# Solving the Rational Interpolation Problem

Teresa Cortadellas, **Carlos D'Andrea**, Eulàlia Montoro

FoCM 2017 Barcelona

# Lagrange Interpolation

# Lagrange Interpolation

Given $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{K}^2$

# Lagrange Interpolation

Given $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{K}^2$
compute $P \in \mathbb{K}[x]$ of minimal degree
such that $P(x_i) = y_i, i = 1, \ldots, N$

# “Easy” Solution

# "Easy" Solution

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

# "Easy" Solution

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

$$P = \sum_{j=0}^{N-1} a_j x^j$$

# "Easy" Solution

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^{N-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

$$P = \sum_{j=0}^{N-1} a_j x^j = \sum_{j=1}^{N} y_j \left( \prod_{i \neq j} \frac{x - x_i}{x_j - x_i} \right)$$

# Hermite Interpolation

# Hermite Interpolation

Given $x_1, \ldots, x_L,$

$$y_{10}, \ldots, y_{1N_1-1}, \ldots, y_{L0}, \ldots, y_{LN_L-1}$$

# Hermite Interpolation

Given $x_1, \ldots, x_L$,

$y_{10}, \ldots, y_{1N_1-1}, \ldots, y_{L0}, \ldots, y_{LN_L-1}$

compute $P \in \mathbb{K}[x]$ of minimal degree

such that $\boxed{P^{(j)}(x_i) = y_{ij}}$

# How to solve it?

# How to solve it?

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 0 & 1 & 2x1 & \dots & (N-1)x_1^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & *x_L^{N-N_L-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_{10} \\ \vdots \\ y_{LN_L-1} \end{pmatrix}$$

# How to solve it?

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 0 & 1 & 2x1 & \dots & (N-1)x_1^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & *x_L^{N-N_L-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_{10} \\ \vdots \\ y_{LN_L-1} \end{pmatrix}$$

$$P = \sum_{j=0}^{N-1} a_j x^j$$

# How to solve it?

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{N-1} \\ 0 & 1 & 2x1 & \dots & (N-1)x_1^{N-2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & *x_L^{N-N_L-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} y_{10} \\ \vdots \\ y_{LN_L-1} \end{pmatrix}$$

$$P = \sum_{j=0}^{N-1} a_j x^j = \sum_{j=0}^{L} P_j(x)$$

# Computing

- Linear Algebra $\mathcal{O}(N^3)$

# Computing

- Linear Algebra $\mathcal{O}(N^3)$
- Chinese Remainder Theorem (Hermite)

# Computing

- Linear Algebra $\mathcal{O}(N^3)$
- Chinese Remainder Theorem (Hermite)
- Divided Differences

# Computing

- Linear Algebra $\mathcal{O}(N^3)$
- Chinese Remainder Theorem (Hermite)
- Divided Differences

$$\mathcal{O}(N \log^2(N))$$

# Rational Interpolation

# Rational Interpolation

Given $x_1, \ldots, x_L,$

$$y_{10}, \ldots, y_{1N_1-1}, \ldots, y_{L0}, \ldots, y_{LN_L-1}$$

# Rational Interpolation

Given $x_1, \ldots, x_L,$

$y_{10}, \ldots, y_{1N_1-1}, \ldots, y_{L0}, \ldots, y_{LN_L-1}$

compute $R \in \mathbb{K}(x)$ such that

$$\boxed{R^{(j)}(x_i) = y_{ij}}$$

# Rational Interpolation

Given $x_1, \ldots, x_L$,

$y_{10}, \ldots, y_{1N_1-1}, \ldots, y_{L0}, \ldots, y_{LN_L-1}$

compute $R \in \mathbb{K}(x)$ such that

$$\boxed{R^{(j)}(x_i) = y_{ij}}$$

of **minimal degree**

# Sub-Problem (Cauchy/osculatory)

# Sub-Problem (Cauchy/osculatory)

Given $0 \leq n \leq N - 1$

# Sub-Problem (Cauchy/osculatory)

Given $0 \leq n \leq N-1$ compute

$A \in \mathbb{K}[x]_{\leq n}$, $B \in \mathbb{K}[x]_{\leq N-n-1}$

# Sub-Problem (Cauchy/osculatory)

Given $0 \leq n \leq N-1$ compute $A \in \mathbb{K}[x]_{\leq n}$, $B \in \mathbb{K}[x]_{\leq N-n-1}$ such that $R := \frac{A}{B}$ solves the interpolation problem

# Problem with the Sub-Problem

# Problem with the Sub-Problem

It may be unsolvable!

# Problem with the Sub-Problem

It may be unsolvable!

$$N = 3,\ n = 1,\ x_1 = 1, x_2 = 2$$
$$y_{1,0} = 1, y_{1,1} = 0, y_{2,0} = 0$$

# Problem with the Sub-Problem

It may be unsolvable!

$$N = 3,\ n = 1,\ x_1 = 1, x_2 = 2$$

$$y_{1,0} = 1, y_{1,1} = 0, y_{2,0} = 0$$

$$\cancel{\frac{a_0+a_1x}{b_0+b_1x}}$$

# The "Problem" is the denominator

# The "Problem" is the denominator

One needs $B(x_i) \neq 0 \, \forall i = 0, \ldots, L$

# The “Problem” is the denominator

One needs $B(x_i) \neq 0 \, \forall i = 0, \ldots, L$

For “generic” input, there is a unique solution

# The “Problem” is the denominator

One needs $B(x_i) \neq 0 \, \forall i = 0, \ldots, L$

For “generic” input, there is a unique solution

Description of the set of “bad points” given in Cortadellas-**D**-Montoro

`arXiv:1704.08563`

# The Weak Sub-Problem

# The Weak Sub-Problem

Given the data $x_i, y_{ij}$

# The Weak Sub-Problem

Given the data $x_i, y_{ij}$ compute $A \in \mathbb{K}[x]_{\leq n}$, $B \in \mathbb{K}[x]_{\leq N-n-1}$

# The Weak Sub-Problem

Given the data $x_i, y_{ij}$ compute $A \in \mathbb{K}[x]_{\leq n}$, $B \in \mathbb{K}[x]_{\leq N-n-1}$ such that $A(x_i) = y_i B(x_i)$

# The Weak Sub-Problem

Given the data $x_i, y_{ij}$ compute $A \in \mathbb{K}[x]_{\leq n}$, $B \in \mathbb{K}[x]_{\leq N-n-1}$ such that $A(x_i) = y_i B(x_i)$

$$A^{(j)}(x_i) = \sum_{s=0}^{j} (j)_s y_{is} B^{(j-s)}(x_i)$$

# Linear Algebra solves it!

# Linear Algebra solves it!

- The Weak Problem is always solvable

# Linear Algebra solves it!

- The Weak Problem is always solvable
- The solution is "unique"

# Linear Algebra solves it!

- The Weak Problem is always solvable
- The solution is "unique"
- The denominator of "the" solution does not vanish in the $x_i$ 's $\iff$

# Linear Algebra solves it!

- The Weak Problem is always solvable
- The solution is "unique"
- The denominator of "the" solution does not vanish in the $x_i$ 's $\Longleftrightarrow$ the sub-problem can be solved

# How to solve it

# How to solve it

- Linear Algebra

# How to solve it

- Linear Algebra
- Symmetric functions/Subresultants

# How to solve it

- Linear Algebra
- Symmetric functions/Subresultants
- **Euclidean Algorithm**

# How to solve it

- Linear Algebra
- Symmetric functions/Subresultants
- **Euclidean Algorithm**
- Barycentric Coordinates

# How to solve it

- Linear Algebra
- Symmetric functions/Subresultants
- **Euclidean Algorithm**
- Barycentric Coordinates
- Jacobi's Method

# How to solve it

- Linear Algebra
- Symmetric functions/Subresultants
- **Euclidean Algorithm**
- Barycentric Coordinates
- Jacobi's Method
- ...

# Euclidean Algorithm revisited

# Euclidean Algorithm revisited

$$f := \prod_{i=1}^{L}(x - x_i)^{N_i}$$

# Euclidean Algorithm revisited

$$f := \prod_{i=1}^{L} (x - x_i)^{N_i}$$
$$g \in \mathbb{K}[x]_{\leq N-1},\; g^{(j)}(x_i) = y_{ij}$$

# Euclidean Algorithm revisited

$$f := \prod_{i=1}^{L} (x - x_i)^{N_i}$$

$$g \in \mathbb{K}[x]_{\leq N-1},\ g^{(j)}(x_i) = y_{ij}$$

$$\gamma^{(\ell)} f + \beta^{(\ell)} g = \alpha^{(\ell)},\ \ell = 1, \ldots$$

# Euclidean Algorithm revisited

$$f := \prod_{i=0}^{L}(x - x_i)^{N_i}$$

$$g \in \mathbb{K}[x]_{\leq N-1},\ g^{(j)}(x_i) = y_{ij}$$

$$\gamma_{\ell-1}^{(\ell)} f + \beta_{\ell}^{(\ell)} g = \alpha_{N-1-\ell}^{(\ell)},\ \ell = 1, \ldots$$

# Euclidean Algorithm revisited

$$f := \prod_{i=0}^{L}(x - x_i)^{N_i}$$
$$g \in \mathbb{K}[x]_{\leq N-1},\ g^{(j)}(x_i) = y_{ij}$$

$$\gamma_{\ell-1}^{(\ell)} f + \beta_{\ell}^{(\ell)} g = \alpha_{N-1-\ell}^{(\ell)},\ \ell = 1, \ldots$$

$$\ell = N - n - 1,\ A = \alpha^{(\ell)},\ B = \beta^{(\ell)}$$

work!

# But...

# But...

you may have $\alpha^{(\ell)} = \beta^{(\ell)} = 0$ !

# But...

you may have $\alpha^{(\ell)} = \beta^{(\ell)} = 0$ !


However

# But...

you may have $\alpha^{(\ell)} = \beta^{(\ell)} = 0$ !


However picking "the closest" nonzero $\ell$ to $N - n - 1$ works...

# Complexity?

# Complexity?

- Computing $f, g$ : $\mathcal{O}(N \log^2 N)$

# Complexity?

- Computing $f, g$ : $\mathcal{O}(N \log^2 N)$
- One single step of the EEA costs

$$\mathcal{O}(M(N) \log(N))$$

# Complexity?

- Computing $f, g$ : $\mathcal{O}(N \log^2 N)$
- One single step of the EEA costs

$$\mathcal{O}(M(N)\log(N)) = \mathcal{O}(N \log^2 N)$$

# Can you improve this?

# Can you improve this?

- **Linear Algebra** based methods: $\mathcal{O}(N^3)$

# Can you improve this?

- **Linear Algebra** based methods: $\mathcal{O}(N^3)$
- **Barycentric Coordinates:**

# Can you improve this?

- **Linear Algebra** based methods: $\mathcal{O}(N^3)$
- **Barycentric Coordinates:** more stable numerically but yet $\mathcal{O}(N^3)$

# Can you improve this?

- **Linear Algebra** based methods: $\mathcal{O}(N^3)$
- **Barycentric Coordinates:** more stable numerically but yet $\mathcal{O}(N^3)$
- **Jacobi's Method:** single roots + mild conditions $\mathcal{O}(N^2)$

# Barycentric coordinates

**(Schneider - Werner 86)**

# Barycentric coordinates

**(Schneider - Werner 86)**

Any choice of $a_1, \ldots, a_N \neq 0$ makes

$$R = \frac{\sum_{i=1}^{N} \frac{a_i y_i}{(x - x_i)}}{\sum_{i=1}^{N} \frac{a_i}{(x - x_i)}}$$

a "solution" of the WRIP

# Barycentric coordinates

**(Schneider - Werner 86)**

Any choice of $a_1, \ldots, a_N \neq 0$ makes

$$R = \frac{\sum_{i=1}^{N} \frac{a_i y_i}{(x - x_i)}}{\sum_{i=1}^{N} \frac{a_i}{(x - x_i)}}$$

a "solution" of the WRIP

RIP is solvable $\Longleftrightarrow$

# Barycentric coordinates

**(Schneider - Werner 86)**

Any choice of $a_1, \ldots, a_N \neq 0$ makes

$$R = \frac{\sum_{i=1}^{N} \frac{a_i y_i}{(x - x_i)}}{\sum_{i=1}^{N} \frac{a_i}{(x - x_i)}}$$

a "solution" of the WRIP

RIP is solvable $\iff$ $a_i \neq 0$

# Jacobi's method

(Egecioglu-Koc 89)

# Jacobi's method

(Egecioglu-Koc 89)

From a Bézout type identity:

$$C\,f + B\,g = A$$

# Jacobi's method

(Egecioglu-Koc 89)

From a Bézout type identity:

$$C\,f + B\,g = A$$

we pass to $x^k C + x^k \frac{B}{f} g = \frac{x^k A}{f}$

# Jacobi's method

**(Egecioglu-Koc 89)**

From a Bézout type identity:

$$C\,f + B\,g = A$$

we pass to $x^k C + x^k \frac{B}{f} g = \frac{x^k A}{f}$ If $k + \deg(A) \leq N - 2$, $\mathrm{Res}_\infty(\frac{x^k A}{f}) = 0$

# Jacobi's method

(Egecioglu-Koc 89)

From a Bézout type identity:

$$C\,f + B\,g = A$$

we pass to $x^k C + x^k \frac{B}{f} g = \frac{x^k A}{f}$ If $k + \deg(A) \leq N - 2$, $\mathrm{Res}_\infty(\frac{x^k A}{f}) = 0$

Separated equations for the coefficients of $A$ and $B$!

# Our contribution

(Cortadellas-D-Montoro)

# Our contribution

**(Cortadellas-D-Montoro)**

Extend Jacobi's method to:

# Our contribution

**(Cortadellas-D-Montoro)**

Extend Jacobi's method to:

- the case with multiplicites

# Our contribution

(Cortadellas-D-Montoro)

Extend Jacobi's method to:

- the case with multiplicites
- eliminate the "conditions"

# Our contribution

**(Cortadellas-D-Montoro)**

Extend Jacobi's method to:

- the case with multiplicites
- eliminate the "conditions"
- improve the complexity

# Moreover...

# Moreover...

- The matrix of residues is computable, of size $n \times n$, of Hankel type

# Moreover...

- The matrix of residues is computable, of size $n \times n$, of Hankel type
- Computing an element in the kernel of this matrix can be done at cost $\mathcal{O}(n \log^2 n)$

# Moreover...

- The matrix of residues is computable, of size $n \times n$, of Hankel type
- Computing an element in the kernel of this matrix can be done at cost $\mathcal{O}(n \log^2 n)$
- $B$ can be interpolated a posteriori

# Cost of the Algorithm

(Cortadellas-**D**-Montoro)

$$\mathcal{O}(n\log^2(n) + N\log(N))$$

# Cost of the Algorithm

(Cortadellas-**D**-Montoro)

$$\mathcal{O}(n\log^2(n) + N\log(N))$$

Euclides approach: $\mathcal{O}(N\log^2(N))$

# What about the minimal problem?

# What about the minimal problem?

Given $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{K}^2$ compute a **rational function** $R \in \mathbb{K}(x)$ such that $R^{(j)}(x_i) = y_{ij}$ of **minimal** degree

# Weak Version (single root)

(Ravi 97)

# Weak Version (single root)

**(Ravi 97)**

Any pair $(A, B)$ solves the weak RIP

# Weak Version (single root)

**(Ravi 97)**

Any pair $(A, B)$ solves the weak RIP

$$\Longleftrightarrow \begin{pmatrix} x-x_1 & 0 & \dots & 0 & 1 & -y_1 \\ 0 & x-x_2 & \dots & 0 & 1 & -y_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & x-x_N & 1 & -y_N \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \\ A \\ B \end{pmatrix} = \mathbf{0}$$

# Hilbert's Syzygy's Theorem

# Hilbert's Syzygy's Theorem

- The kernel of this matrix is free of rank 2

# Hilbert's Syzygy's Theorem

- The kernel of this matrix is free of rank 2
- There are two generators of degrees $\mu \leq N - \mu$

# Hilbert's Syzygy's Theorem

- The kernel of this matrix is free of rank 2
- There are two generators of degrees $\mu \leq N - \mu$
- $\mu$ should be the minimal degree!

# Don't bother computing that!

(Cortadellas-**D**-Montoro)

# Don't bother computing that!

(Cortadellas-**D**-Montoro)

- $\mu$ is closest degree of a remainder in the EEA to $\frac{N}{2}$
- The "$\mu$-basis" are just two consecutive rows of the EEA!

# From the Weak to the Strong...

(Cortadellas-**D**-Montoro)

# From the Weak to the Strong...

(Cortadellas-**D**-Montoro)
One uses the stratification of "bad points" given in
Cortadellas-**D**-Montoro
arXiv:1704.08563

# To conclude...

# To conclude...

**Thanks**

Agnes, Marta and Thorsten!

# And Thank you!