Prony's method in several variables, oligonomials and Linear Algebra

Tomas Sauer

Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung
Fakultät für Informatik und Mathematik

FORWISS

Universität Passau

Bernried, March 2016

The History

Introduction

The History

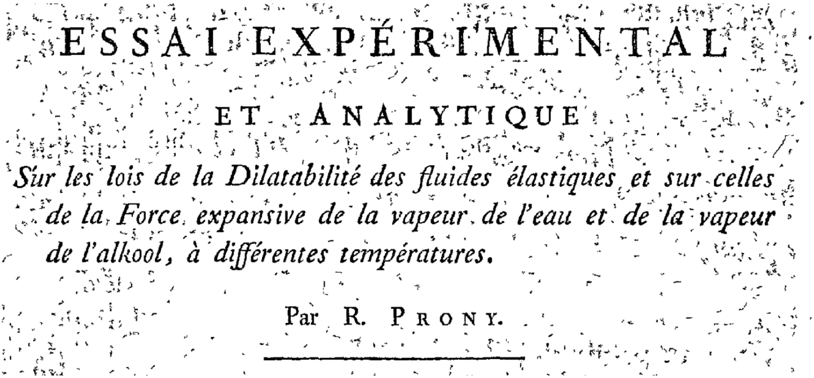
Introduction

1795: *My name is R. Prony and I have a problem with alcohol.*

The History

Introduction

1795: *My name is R. Prony and I have a problem with alcohol.*


ESSAI EXPÉRIMENTAL

ET ANALYTIQUE

Sur les lois de la Dilatabilité des fluides élastiques et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures.

Par R. Prony.

The Problem

Prony's problem

For a *exponential polynomial*

$$f(x) = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T x}, \qquad \Omega \subset (\mathbb{R} + i\mathbb{R}/2\pi\mathbb{Z})^s, \quad 0 \neq f_\omega \in \mathbb{C},$$

determine Ω and f_ω from samples of f.

Oligonomials

For an *algebraic polynomial*

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha, \qquad A \subset \mathbb{N}_0^s, \, 0 \neq f_\alpha \in \mathbb{C}, \qquad \#A < \infty,$$

determine A and f_α from samples of f.

The Problem

UNIVERSITÄT PASSAU

Prony's problem

For a *exponential polynomial*

$$f(x) = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T x}, \qquad \Omega \subset \left(\mathbb{R} + i\mathbb{R}/2\pi\mathbb{Z}\right)^s, \quad 0 \neq f_\omega \in \mathbb{C},$$

determine Ω and f_ω from samples of f.

Oligonomials

For an *algebraic polynomial*

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha, \qquad A \subset \mathbb{N}_0^s, \, 0 \neq f_\alpha \in \mathbb{C}, \qquad \#A < \infty,$$

determine A and f_α from samples of f.

The Problem

Prony's problem

For a *exponential polynomial*

$$f(x) = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T x}, \qquad \Omega \subset (\mathbb{R} + i\mathbb{R}/2\pi\mathbb{Z})^s, \quad 0 \neq f_\omega \in \mathbb{C},$$

determine Ω and f_ω from samples of f.

Oligonomials

For an *algebraic polynomial*

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha, \qquad A \subset \mathbb{N}_0^s, \, 0 \neq f_\alpha \in \mathbb{C}, \qquad \#A < \infty,$$

determine A and f_α from samples of f.

The Problem

Prony's problem

For a sparse *exponential polynomial*

$$f(x) = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T x}, \qquad \Omega \subset (\mathbb{R} + i\mathbb{R}/2\pi\mathbb{Z})^s, \quad 0 \neq f_\omega \in \mathbb{C},$$

determine Ω and f_ω from samples of f.

Oligonomials (fewnomials)

For a sparse *algebraic polynomial*

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha, \qquad A \subset \mathbb{N}_0^s, \, 0 \neq f_\alpha \in \mathbb{C}, \qquad \#A < \infty,$$

determine A and f_α from samples of f.

Remarks on Prony's Problem

UNIVERSITÄT PASSAU

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

UNIVERSITÄT PASSAU

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

UNIVERSITÄT PASSAU

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

UNIVERSITÄT PASSAU

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex straightforward.

Remarks on Prony's Problem

Assumptions

1. Problem is *sparse*: $\#\Omega / \#A$ small.
2. Frequencies $\omega \in \Omega$ and powers $\alpha \in A$ can vary.
3. No embedding in $\Pi_{\deg A}$.
4. A priory knowledge: $\#\Omega$ or $\#A$.

Problem structure (Prony)

1. Frequencies: *nonlinear* problem.
2. Coefficients: *linear* problem.
3. Evaluation points: Grids $\Gamma \subset \mathbb{Z}^s$.
4. Real only – complex conjugation straightforward.

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n = \left\{ p(x) = \sum_{|\alpha| \le n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ of degree n.
3. *Degree*

$$\deg p = \max \{ |\alpha| : f_\alpha \neq 0 \},$$

Coefficient vectors

$$p \simeq \mathbf{p} = [p_\alpha : |\alpha| \le \deg p] .$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n = \left\{ p(x) = \sum_{|\alpha| \le n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ of degree n.
3. *Degree*

$$\deg p = \max \{|\alpha| : f_\alpha \neq 0\},$$

Coefficient vectors

$$p \simeq \mathbf{p} = [p_\alpha : |\alpha| \le \deg p] .$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n = \left\{ p(x) = \sum_{|\alpha| \le n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ of degree n.
3. *Degree*

$$\deg p = \max \{ |\alpha| : f_\alpha \neq 0 \},$$

Coefficient vectors

$$p \simeq \mathbf{p} = [p_\alpha : |\alpha| \le \deg p] .$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n^0 = \left\{ p(x) = \sum_{|\alpha|=n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ *homogeneous* of degree n.
3. *Degree*

$$\deg p = \max \{ |\alpha| : f_\alpha \neq 0 \},$$

Coefficient vectors

$$p \simeq \mathbf{p} = [p_\alpha : |\alpha| \leq \deg p] .$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n^0 = \left\{ p(x) = \sum_{|\alpha|=n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ *homogeneous* of degree n.
3. *Degree*

$$\deg p = \max \{ |\alpha| : f_\alpha \neq 0 \},$$

Coefficient vectors

$$p \simeq \mathbf{p} = [p_\alpha : |\alpha| \leq \deg p] .$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n^0 = \left\{ p(x) = \sum_{|\alpha|=n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ *homogeneous* of degree n.
3. *Degree* and *homogeneous decomposition*

$$\deg p = \max\{|\alpha| : f_\alpha \neq 0\}, \qquad p(x) = \sum_{j=0}^{\deg p} p_j^0(x), \quad p_j^0 \in \Pi_j^0.$$

Coefficient vectors

$$p \simeq \boldsymbol{p} = [p_\alpha : |\alpha| \le \deg p].$$

Notation

Polynomials

1. $\Pi = \mathbb{R}[x] = \mathbb{R}[x_1, \dots, x_s]$.
2. $\Pi_n^0 = \left\{ p(x) = \sum_{|\alpha|=n} p_\alpha x^\alpha : p_\alpha \in \mathbb{R} \right\}$ *homogeneous* of degree n.
3. *Degree* and *homogeneous decomposition*

$$\deg p = \max \{|\alpha| : f_\alpha \neq 0\}, \qquad p(x) = \sum_{j=0}^{\deg p} p_j^0(x), \quad p_j^0 \in \Pi_j^0.$$

Coefficient vectors

$$p \simeq \boldsymbol{p} = [p_\alpha : |\alpha| \le \deg p].$$

Prony's Trick

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$:

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

UNIVERSITÄT PASSAU

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$:

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$:

$$\boldsymbol{F}_n \boldsymbol{p}$$

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

Prony's Trick

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(F_n p)_\alpha =$$

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation …

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(F_n p)_\alpha = \sum_{|\beta| \le n} f(\alpha+\beta)\, p_\beta$$

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation …

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(\boldsymbol{F}_n \boldsymbol{p})_\alpha = \sum_{|\beta| \le n} \sum_{\omega \in \Omega} f_\omega e^{\omega^T(\alpha+\beta)} p_\beta$$

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

Prony's Trick

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation …

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(F_n p)_\alpha = \sum_{|\beta| \le n} \sum_{\omega \in \Omega} f_\omega e^{\omega^T \alpha} e^{\omega^T \beta} p_\beta$$

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation …

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(\boldsymbol{F}_n \boldsymbol{p})_\alpha = \sum_{\omega\in\Omega} f_\omega e^{\omega^T\alpha} \sum_{|\beta|\le n} e^{\omega^T\beta} p_\beta$$

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

Prony's Trick

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(F_n p)_\alpha = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T \alpha} \, p(e^\omega),$$

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation …

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(\boldsymbol{F}_n \boldsymbol{p})_\alpha = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T \alpha} p(e^\omega), \qquad e^\omega = (e^{\omega_1}, \dots, e^{\omega_s}) =: x_\omega$$

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

Prony's Trick

UNIVERSITÄT PASSAU

A Hankel matrix

$$F_n := \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(F_n p)_\alpha = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T \alpha} p(x_\omega), \qquad e^\omega = (e^{\omega_1}, \dots, e^{\omega_s}) =: x_\omega$$

Consequence

$p(X_\Omega) = 0$ implies $F_n p = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(\boldsymbol{F}_n \boldsymbol{p})_\alpha = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T \alpha} p(x_\omega), \qquad e^\omega = (e^{\omega_1}, \dots, e^{\omega_s}) =: x_\omega$$

$X_\Omega := \{x_\omega : \omega \in \Omega\} = e^\Omega.$

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

Prony's Trick

A Hankel matrix

$$\boldsymbol{F}_n := \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le n \end{array} \right] \in \mathbb{R}^{d_n \times d_n}, \qquad d_n = \binom{n+s}{s}.$$

A computation ...

For $p \in \Pi_n$ and $|\alpha| \le n$:

$$(\boldsymbol{F}_n \boldsymbol{p})_\alpha = \sum_{\omega \in \Omega} f_\omega \, e^{\omega^T \alpha} p(x_\omega), \qquad e^\omega = (e^{\omega_1}, \dots, e^{\omega_s}) =: x_\omega$$

$X_\Omega := \{x_\omega : \omega \in \Omega\} = e^\Omega$.

Consequence

$p(X_\Omega) = 0$ implies $\boldsymbol{F}_n \boldsymbol{p} = 0$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Moment problem ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

4. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
5. Converse?
6. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow p \in \ker F_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker F_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse?
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$ gives $\boldsymbol{F}_0 =$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$ gives $\boldsymbol{F}_0 = f_\omega e^{\omega^T 0} + f_{\omega'} e^{\omega'^T 0}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem ...

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$ gives $\boldsymbol{F}_0 = f_\omega + f_{\omega'}$

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

UNIVERSITÄT PASSAU

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$ gives $\boldsymbol{F}_0 = 0$.

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

The Ideal Prony

The ideal

1. $I_\Omega = I(X_\Omega) = \{p \in \Pi : p(X_\Omega) = 0\}$.
2. Zero dimensional ideal.
3. Interpolation problem …

Prony again

1. $p \in I_\Omega \cap \Pi_n \Rightarrow \boldsymbol{p} \in \ker \boldsymbol{F}_n$.
2. Converse? No!
3. $\Omega = \{\omega, \omega'\}, f_\omega = -f_{\omega'}$ gives $\boldsymbol{F}_0 = 0$.

Theorem

For n large enough we have $I_\Omega \cap \Pi_n \simeq \ker \boldsymbol{F}_n$.

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"?*

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q}$$

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"*?

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q}$$

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"?*

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q}$$

Minimal Degree Interpolation

UNIVERSITÄT PASSAU

Goal

Answer the question *What is "large enough"?*

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q}$$

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"?*

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \leq \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q}$$

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"?*

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q} =: d(X)$$

Minimal Degree Interpolation

Goal

Answer the question *What is "large enough"?* $n \ge d(X_\Omega)$.

Interpolation space

$\mathscr{P} \subset \Pi$ *degree reducing interpolation space* with respect to a *finite* $X \subset \mathbb{R}^s$: for $g \in \Pi$ there exists a *unique* $p \in \mathscr{P}$ such that

1. $p(X) = g(X)$.
2. $\deg p \le \deg g$.

Theorem

If $\mathscr{P}, \mathscr{Q}$ are degree reducing interpolation spaces for X then

$$\deg \mathscr{P} = \max\{\deg p : p \in \mathscr{P}\} = \deg \mathscr{Q} =: d(X)$$

Ideals and Bases

H–bases

1. $H \subset \Pi$ *H–basis* for I if

$$g \in I \qquad \Leftrightarrow \qquad g = \sum_{h \in H} g_h \, h, \quad \deg g_h + \deg h \le \deg g.$$

2. Alternative: *homogeneous ideals*. (Macaulay 1916)

Definition

(Homogeneous) leading form

$$\Lambda(p) = p^0_{\deg p} = \sum_{|\alpha| = \deg p} p_\alpha \, (\cdot)^\alpha,$$

Ideals and Bases

UNIVERSITÄT PASSAU

H–bases

1. $H \subset \Pi$ *H–basis* for I if

$$g \in I \qquad \Leftrightarrow \qquad g = \sum_{h \in H} g_h \, h, \quad \deg g_h + \deg h \le \deg g.$$

2. Alternative: *homogeneous ideals.* (Macaulay 1916)

Definition

(Homogeneous) leading form

$$\Lambda(p) = p^0_{\deg p} = \sum_{|\alpha| = \deg p} p_\alpha \, (\cdot)^\alpha,$$

Ideals and Bases

H–bases

1. $H \subset \Pi$ *H–basis* for I if

$$g \in I \qquad \Leftrightarrow \qquad g = \sum_{h \in H} g_h \, h, \quad \deg g_h + \deg h \le \deg g.$$

2. Alternative: *homogeneous ideals*. (Macaulay 1916)

Definition

(Homogeneous) leading form

$$\Lambda(p) = p^0_{\deg p} = \sum_{|\alpha| = \deg p} p_\alpha \, (\cdot)^\alpha,$$

Ideals and Bases

H–bases

1. $H \subset \Pi$ *H–basis* for I if

$$g \in I \qquad \Leftrightarrow \qquad g = \sum_{h \in H} g_h \, h, \quad \deg g_h + \deg h \le \deg g.$$

2. Alternative: *homogeneous ideals*. (Macaulay 1916)

Definition

(Homogeneous) leading form

$$\Lambda(p) = p^0_{\deg p} = \sum_{|\alpha| = \deg p} p_\alpha \, (\cdot)^\alpha,$$

Ideals and Bases

H–bases

1. $H \subset \Pi$ *H–basis* for I if

$$g \in I \qquad \Leftrightarrow \qquad g = \sum_{h \in H} g_h \, h, \quad \deg g_h + \deg h \le \deg g.$$

2. Alternative: *homogeneous ideals*. (Macaulay 1916)

Definition

(Homogeneous) leading form and *ideal generated by G*

$$\Lambda(p) = p^0_{\deg p} = \sum_{|\alpha| = \deg p} p_\alpha \, (\cdot)^\alpha, \qquad \langle G \rangle = \left\{ \sum_{g \in G} q_g \, g : q_g \in \Pi \right\}.$$

Reduction

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$.
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

UNIVERSITÄT PASSAU

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

UNIVERSITÄT PASSAU

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
    1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$
    2. Replace $p \leftarrow p - \sum_{g \in G} p_g g - r$.
    3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

UNIVERSITÄT PASSAU

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \, \Lambda(g) + r$
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g \, g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g \, g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi_j^0$.

Reduction

UNIVERSITÄT PASSAU

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$ such that $r \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_{\deg p}$.
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g \, g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g \, g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_j$.

Reduction

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \Lambda(g) + r$ such that $r \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_{\deg p}$.
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g \, g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g \, g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_j$.

Reduction

UNIVERSITÄT PASSAU

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \, \Lambda(g) + r$ such that $r \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_{\deg p}$.
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g \, g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g \, g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_j$.

Reduction

Further ingredient

1. Inner product $(\cdot,\cdot) : \Pi \times \Pi \to \mathbb{R}$.
2. For example $(p,q) = \boldsymbol{p}^T \boldsymbol{q} = \sum_\alpha p_\alpha q_\alpha$.

Division with remainder for p, G

1. While $p \neq 0$
   1. Decompose $\Lambda(p) = \sum_{g \in G} p_g \, \Lambda(g) + r$ such that $r \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_{\deg p}$.
   2. Replace $p \leftarrow p - \sum_{g \in G} p_g \, g - r$.
   3. Sum up the p_g and r.
2. **Result:** $p = \sum_{g \in G} p_g \, g + r$ where $r_j \perp \langle \Lambda(G) \rangle^0 \cap \Pi^0_j$.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h\, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h\, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g\in G} p_h\, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$

Definition

1. $\nu_I(p) := r = p - \sum_{g\in G} p_h\, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h\, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h\, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem

$N_{I(X)}$ is a degree reducing interpolation space for X.

Reduction II

Theorem

If H is an *H–basis* and $p = \sum_{g \in G} p_h \, h + r$ computed by reduction, then

1. $r = 0$ iff $p \in \langle H \rangle$.
2. r depends only on $\langle H \rangle$ and $(\cdot, \cdot)$.

Definition

1. $\nu_I(p) := r = p - \sum_{g \in G} p_h \, h$ *normal form* of p modulo $\langle H \rangle =: I$.
2. $N_I := \nu_I(\Pi) \simeq \Pi / I$ *inverse system* for I. (Macaulay '16, Gröbner '37)

Theorem (reduction is interpolation)

$N_{I(X)}$ is a degree reducing interpolation space for X.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space
3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space
3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space
3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

UNIVERSITÄT PASSAU

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$x \in X$.

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\ell_x := \nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\ell_x := \nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\ell_x := \nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\ell_x := \nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

The Normal Form Space

Normal forms for points

1. Given $X \subset \mathbb{R}^s$, $\#X < \infty$.
2. Basis for normal form space

$$\ell_x := \nu_{I(X)} \left(\prod_{x' \in X \setminus \{x\}} \frac{(x - x')(\cdot - x')}{\|x - x'\|^2} \right), \qquad x \in X.$$

3. Satisfies $\ell_x(x') = \delta_{xx'}$, $x, x' \in X$.

Theorem

For $p \in \Pi$ are equivalent:

1. p is reduced, i.e. $p = \nu_{I(X)}(p)$.
2. $p \in N_{I(X)}$.

Where is My Prony?

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$F_{n,k} = \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$F_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$F_{n,k} = \left[f(\alpha+\beta) : \begin{array}{c} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

UNIVERSITÄT PASSAU

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$F_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

UNIVERSITÄT PASSAU

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$\boldsymbol{F}_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$[\boldsymbol{F}_{n,k-1} \, *] = \boldsymbol{F}_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The “magic” number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

UNIVERSITÄT PASSAU

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$[\boldsymbol{F}_{n,k-1} *] = \boldsymbol{F}_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$[\boldsymbol{F}_{n,k-1} \, *] = \boldsymbol{F}_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Where is My Prony?

UNIVERSITÄT PASSAU

Goal

1. Compute H–basis and normal form space N_Ω for $I_\Omega = I(e^\Omega)$.
2. Compute frequencies Ω (standard).
3. Compute coefficients f_ω (linear system).

Slightly different Hankel matrix

$$[\boldsymbol{F}_{n,k-1} \ast] = \boldsymbol{F}_{n,k} = \left[f(\alpha+\beta) : \begin{array}{l} |\alpha| \le n \\ |\beta| \le k \end{array} \right], \qquad k \le n.$$

The "magic" number

1. Fundamental guess: $n > d(X_\Omega) = \deg N_\Omega$, e.g., $n \ge \#\Omega$.
2. Also needed in 1d Prony!

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1.$
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

UNIVERSITÄT PASSAU

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \quad \text{and} \quad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

UNIVERSITÄT PASSAU

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} \left[f(\alpha) : |\alpha| \le n \right] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$. Computes *affine Hilbert function*.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \geq \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \leq n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \leq n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$. Computes *affine Hilbert function*.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \ldots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$. Computes *affine Hilbert function*.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

UNIVERSITÄT PASSAU

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$. Computes *affine Hilbert function*.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Ideal and Prony

Theorem ($F_{n,k}$ and the ideal)

If $n \ge \deg N_\Omega$ then

1. $\operatorname{rank} F_{n,0} = \operatorname{rank} [f(\alpha) : |\alpha| \le n] = 1$.
2. $\ker F_{n,k} \simeq I(X_\Omega) \cap \Pi_k$, $k \le n$.
3. $\operatorname{rank} F_{n,k} = \dim(N_\Omega \cap \Pi_k)$. Computes *affine Hilbert function*.
4. $\operatorname{rank} F_{n,k} < \operatorname{rank} F_{n,k+1}$ iff $k < \deg N_\Omega$.

Basic idea

1. Compute $F_{n,0}, F_{n,1}, \dots$ until $\operatorname{rank} F_{n,k} = \operatorname{rank} F_{n,k+1}$.
2. In each step compute extend

$$(H \cap \Pi_k) \to (H \cap \Pi_{k+1}) \qquad \text{and} \qquad (N \cap \Pi_k) \to (N \cap \Pi_{k+1})$$

by means of Linear Algebra.

Linear Algebra I

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\le k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{jv_j}],$$
2. decompose
$$V_*^T H_{\le k-1} = QR = [Q_1\, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\le k-1}$: $H_k = V_* Q_2$

Linear Algebra I

Kernel vs. rank

With $\boldsymbol{F}_{n,k} = \boldsymbol{U}_k \Sigma \boldsymbol{V}_k^T$:

1. $\ker \boldsymbol{F}_{n,k}$: $\boldsymbol{V}_*$ columns of $\boldsymbol{V}$ for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} \boldsymbol{F}_{n,k}$: $\boldsymbol{V}_+$ columns wrto nonzero singular values.

The H–basis update

1. With
$$\boldsymbol{H}_{\le k-1} = [\boldsymbol{H}_0 \dots \boldsymbol{H}_{k-1}], \qquad \boldsymbol{H}_j = [\boldsymbol{h}_{j1} \dots \boldsymbol{h}_{j\nu_j}],$$
2. decompose
$$\boldsymbol{V}_*^+ \boldsymbol{H}_{\le k-1} = \boldsymbol{Q}\boldsymbol{R} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R} \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $\boldsymbol{H}_{\le k-1}$: $\boldsymbol{H}_k = \boldsymbol{V}_* \boldsymbol{Q}_2$

Linear Algebra I

UNIVERSITÄT PASSAU

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\le k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{j\nu_j}],$$
2. decompose
$$V_*^T H_{\le k-1} = QR = [Q_1\, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\le k-1}$: $H_k = V_* Q_2$

Linear Algebra I

Kernel vs. rank

With $\boldsymbol{F}_{n,k} = \boldsymbol{U}_k \Sigma \boldsymbol{V}_k^T$, $\boldsymbol{V}_k = [\boldsymbol{V}_+ \, \boldsymbol{V}_*]$:

1. $\ker \boldsymbol{F}_{n,k}$: $\boldsymbol{V}_*$ columns of $\boldsymbol{V}$ for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} \boldsymbol{F}_{n,k}$: $\boldsymbol{V}_+$ columns wrto nonzero singular values.

The H–basis update

1. With

$$\boldsymbol{H}_{\le k-1} = [\boldsymbol{H}_0 \dots \boldsymbol{H}_{k-1}], \qquad \boldsymbol{H}_j = [\boldsymbol{h}_{j1} \dots \boldsymbol{h}_{jv_j}],$$

2. decompose

$$\boldsymbol{V}_*^\dagger \boldsymbol{H}_{\le k-1} = \boldsymbol{Q}\boldsymbol{R} = [\boldsymbol{Q}_1 \, \boldsymbol{Q}_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

3. *Orthogonal complement* of $\boldsymbol{H}_{\le k-1}$: $\boldsymbol{H}_k = \boldsymbol{V}_* \boldsymbol{Q}_2$

Linear Algebra I

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$, $V_k = [V_+ \, V_*]$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\le k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{jv_j}],$$
2. decompose
$$V_*^+ H_{\le k-1} = QR = [Q_1 \, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\le k-1}$: $H_k = V_* Q_2$

Linear Algebra I

UNIVERSITÄT PASSAU

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$, $V_k = [V_+ \, V_*]$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\leq k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{jv_j}],$$
2. decompose
$$V_*^+ H_{\leq k-1} = QR = [Q_1 \, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\leq k-1}$: $H_k = V_* Q_2$

Linear Algebra I

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$, $V_k = [V_+ \, V_*]$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\le k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{j\nu_j}],$$
2. decompose
$$V_*^+ H_{\le k-1} = QR = [Q_1 \, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\le k-1}$: $H_k = V_* Q_2$

UNIVERSITÄT PASSAU

Kernel vs. rank

With $F_{n,k} = U_k \Sigma V_k^T$, $V_k = [V_+ \, V_*]$:

1. $\ker F_{n,k}$: V_* columns of V for $\sigma_{\ell,\ell} = 0$.
2. $\operatorname{rank} F_{n,k}$: V_+ columns wrto nonzero singular values.

The H–basis update

1. With
$$H_{\le k-1} = [H_0 \dots H_{k-1}], \qquad H_j = [h_{j1} \dots h_{jv_j}],$$
2. decompose
$$V_*^+ H_{\le k-1} = QR = [Q_1 \, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$
3. *Orthogonal complement* of $H_{\le k-1}$: $H_k = V_* Q_2$

Linear Algebra II

The normal forms

1. General idea: $N_k = \Pi_k^0 \ominus \Lambda(H_k)$.
2. Again
$$\Lambda(H_k) = Q_k \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \qquad Q_k = [Q_{k,1} \; Q_{k,2}].$$
3. $N_j = Q_{j,2}$.

Reduction

1. With $k := \deg p$ solve $R_k c = Q_{k,1}^T \Lambda(p)$.
2. Set $r_k^0 = \Lambda(p) - \Lambda(H_k)\, c$.
3. Replace p by $p - H_k\, c - r_k^0$.

Linear Algebra II

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1}\ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(p)$.
2. Set $r_k^0 = \Lambda(p) - \Lambda(\boldsymbol{H}_k)\, \boldsymbol{c}$.
3. Replace p by $p - \boldsymbol{H}_k\, \boldsymbol{c} - r_k^0$.

Linear Algebra II

UNIVERSITÄT PASSAU

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1}\ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(p)$.
2. Set $r_k^0 = \Lambda(p) - \Lambda(\boldsymbol{H}_k)\,\boldsymbol{c}$.
3. Replace p by $p - \boldsymbol{H}_k\,\boldsymbol{c} - r_k^0$.

Linear Algebra II

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1} \ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(p)$.
2. Set $r_k^0 = \Lambda(p) - \Lambda(\boldsymbol{H}_k)\,\boldsymbol{c}$.
3. Replace p by $p - \boldsymbol{H}_k\,\boldsymbol{c} - r_k^0$.

Linear Algebra II

The normal forms

1. General idea: $N_k = \Pi_k^0 \ominus \Lambda(H_k)$.
2. Again
$$\Lambda(H_k) = Q_k \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \qquad Q_k = [Q_{k,1} \; Q_{k,2}].$$
3. $N_j = Q_{j,2}$.

Reduction

1. With $k := \deg p$ solve $R_k c = Q_{k,1}^T \Lambda(p)$.
2. Set $r_k^0 = \Lambda(p) - \Lambda(H_k)\, c$.
3. Replace p by $p - H_k\, c - r_k^0$.

Linear Algebra II

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1}\ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(\boldsymbol{p})$.
2. Set $\boldsymbol{r}_k^0 = \Lambda(\boldsymbol{p}) - \Lambda(\boldsymbol{H}_k)\,\boldsymbol{c}$.
3. Replace $\boldsymbol{p}$ by $\boldsymbol{p} - \boldsymbol{H}_k\,\boldsymbol{c} - \boldsymbol{r}_k^0$.

Linear Algebra II

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1}\ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(\boldsymbol{p})$.
2. Set $\boldsymbol{r}_k^0 = \Lambda(\boldsymbol{p}) - \Lambda(\boldsymbol{H}_k)\,\boldsymbol{c}$.
3. Replace $\boldsymbol{p}$ by $\boldsymbol{p} - \boldsymbol{H}_k\,\boldsymbol{c} - \boldsymbol{r}_k^0$.

Linear Algebra II

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1} \ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(\boldsymbol{p})$.
2. Set $\boldsymbol{r}_k^0 = \Lambda(\boldsymbol{p}) - \Lambda(\boldsymbol{H}_k)\, \boldsymbol{c}$.
3. Replace $\boldsymbol{p}$ by $\boldsymbol{p} - \boldsymbol{H}_k\, \boldsymbol{c} - \boldsymbol{r}_k^0$.

The normal forms

1. General idea: $\boldsymbol{N}_k = \Pi_k^0 \ominus \Lambda(\boldsymbol{H}_k)$.
2. Again
$$\Lambda(\boldsymbol{H}_k) = \boldsymbol{Q}_k \begin{bmatrix} \boldsymbol{R}_k \\ 0 \end{bmatrix}, \qquad \boldsymbol{Q}_k = [\boldsymbol{Q}_{k,1} \ \boldsymbol{Q}_{k,2}].$$
3. $\boldsymbol{N}_j = \boldsymbol{Q}_{j,2}$.

Reduction

1. With $k := \deg p$ solve $\boldsymbol{R}_k \boldsymbol{c} = \boldsymbol{Q}_{k,1}^T \Lambda(\boldsymbol{p})$.
2. Set $\boldsymbol{r}_k^0 = \Lambda(\boldsymbol{p}) - \Lambda(\boldsymbol{H}_k)\, \boldsymbol{c}$.
3. Replace $\boldsymbol{p}$ by $\boldsymbol{p} - \boldsymbol{H}_k\, \boldsymbol{c} - \boldsymbol{r}_k^0$.

until $\boldsymbol{p} = 0$.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $H_{\leq k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $N_{\leq k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $N_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

UNIVERSITÄT PASSAU

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\leq k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\leq k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

Ideals and Normal Forms

Theorem

If $k > \deg N_\Omega$

1. The columns of $\boldsymbol{H}_{\le k}$ are coefficient vectors of an H–basis of I_Ω.
2. The matrix $\boldsymbol{N}_{\le k}$ contains a graded homogeneous basis for N_Ω.

Remarks

1. All by means of standard Linear Algebra (`Matlab`).
2. H–basis is *very* redundant.
3. Reduction is simple and fast.
4. Numerically stable due to *orthogonality*.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication: $N_\Omega \ni p$
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega$$

Theorem (Stetter, Sticklberger, …)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication: $N_\Omega \ni p$
2. Linear operation on $N_\Omega \to$ matrix M_j for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega$$

Theorem (Stetter, Sticklberger, …)

The *eigenvalues* of the M_j are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication: $N_\Omega \ni p \mapsto \quad (\cdot)_j p \quad \in \Pi$
2. Linear operation on $N_\Omega \to$ matrix M_j for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the M_j are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix M_j for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the M_j are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega\in\Omega} p(x_\omega)\,\ell_\omega$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega \qquad \Rightarrow$$

Theorem (Stetter, Sticklberger, …)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega \in \Omega} p(x_\omega)\, \ell_\omega \qquad \Rightarrow \qquad \nu\left((\cdot)_j \ell_\omega\right)$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega\in\Omega} p(x_\omega)\,\ell_\omega \qquad \Rightarrow \qquad \nu\left((\cdot)_j \ell_\omega\right) = \sum_{\omega'\in\Omega} (x_{\omega'})_j \ell_\omega(x_{\omega'})\,\ell_{\omega'}$$

Theorem (Stetter, Sticklberger, . . .)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega\in\Omega} p(x_\omega)\,\ell_\omega \qquad \Rightarrow \qquad \nu\left((\cdot)_j \ell_\omega\right) = (x_\omega)_j\,\ell_\omega$$

Theorem (Stetter, Sticklberger, ...)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* ℓ_ω, $\omega \in \Omega$.

(Frobenius) Companion Matrices

Multiplication tables

1. Multiplication modulo ideal: $N_\Omega \ni p \mapsto \nu\left((\cdot)_j p\right) \in N_\Omega$.
2. Linear operation on $N_\Omega \to$ matrix $\boldsymbol{M}_j$ for a basis N of N_Ω.

Observation (reduction is interpolation)

$$\nu(p) = \sum_{\omega\in\Omega} p(x_\omega)\,\ell_\omega \qquad \Rightarrow \qquad \nu\left((\cdot)_j \ell_\omega\right) = (x_\omega)_j\,\ell_\omega$$

Theorem (Stetter, Sticklberger, . . .)

The *eigenvalues* of the $\boldsymbol{M}_j$ are $(z_\omega)_j$ and the *eigenvectors* $\boldsymbol{\ell}_\omega$, $\omega \in \Omega$.

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation

$$N_k$$

3. Eigenvalue computation and matching …

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation

$$N_k$$

3. Eigenvalue computation and matching …

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation

$$N_k$$

3. Eigenvalue computation and matching …

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation of multiplication

$$L_{k,j} N_k, \qquad L_{k,j} = \sum_{|\alpha|=k} e_{\alpha+\epsilon_j} e_\alpha^T.$$

3. Eigenvalue computation and matching …

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation of multiplication and reduction

$$\left(I - Q_{k,1} Q_{k,1}^T \right) L_{k,j} N_k, \qquad L_{k,j} = \sum_{|\alpha|=k} e_{\alpha+\epsilon_j} e_\alpha^T.$$

3. Eigenvalue computation and matching …

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation of multiplication and reduction

$$M_j \leftarrow \left(I - Q_{k,1} Q_{k,1}^T \right) L_{k,j} N_k, \qquad L_{k,j} = \sum_{|\alpha|=k} e_{\alpha+\epsilon_j} e_\alpha^T.$$

3. Eigenvalue computation and matching ...

Linear Algebra III

Companion matrices

1. Block structure:

$$M_j = \begin{bmatrix} M^j_{0,0} & M^j_{0,1} & \dots & M^j_{0,m-1} & M^j_{0,m} \\ M^j_{1,0} & M^j_{1,1} & \dots & M^j_{1,m-1} & M^j_{1,m} \\ & M^j_{2,1} & \dots & M^j_{2,m-1} & M^j_{2,m} \\ & & \ddots & \vdots & \vdots \\ & & & M^j_{m,m-1} & M^j_{m,m} \end{bmatrix}$$

2. Computation of multiplication and reduction

$$M_j \leftarrow \left(I - Q_{k,1} Q_{k,1}^T \right) L_{k,j} N_k, \qquad L_{k,j} = \sum_{|\alpha|=k} e_{\alpha+\epsilon_j} e_\alpha^T.$$

3. Eigenvalue computation and matching …

The Linear System

Solve . . .

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $F_{n,0}$.

The Linear System

Solve …

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $F_{n,0}$.

The Linear System

Solve ...

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $F_{n,0}$.

The Linear System

Solve …

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $F_{n,0}$.

The Linear System

Solve . . .

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $F_{n,0}$.

The Linear System

Solve …

$$\left[e^{\omega^T \alpha} : \begin{array}{c} |\alpha| \le k \\ \omega \in \Omega \end{array} \right] [f_\omega : \omega \in \Omega] = [f(\alpha) : |\alpha| \le k] .$$

Properties

1. Transpose of *Vandermonde matrix* for X_Ω.
2. Full rank.
3. Overdetermined system, stabilizes.
4. Right hand side: part of $\boldsymbol{F}_{n,0}$.

"Complete" Algorithm

UNIVERSITÄT PASSAU

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

“Complete” Algorithm

UNIVERSITÄT PASSAU

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

"Complete" Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

"Complete" Algorithm

UNIVERSITÄT PASSAU

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

"Complete" Algorithm

UNIVERSITÄT PASSAU

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $\boldsymbol{F}_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

“Complete” Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab`/`Octave`.
2. Works for complex frequencies immediately.

“Complete” Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $\boldsymbol{F}_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

"Complete" Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_{\Omega}$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

“Complete” Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $F_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

"Complete" Algorithm

Procedure

1. Get a good guess for n.
2. Compute H–basis and normal form space from $\boldsymbol{F}_{n,k}$.
3. Compute multiplication tables.
4. Determine and match eigenvalues.
5. Compute $\Omega = \log X_\Omega$.
6. Setup Vandermonde matrix and solve for transpose.

Remarks

1. Surprisingly easy to implement in `Matlab/Octave`.
2. Works for complex frequencies immediately.

Some Results

Random frequencies & coefficients, real, 100 tests

parameters			average error		max error	
s	# freq.	n	coeff	freq	coeff	freq
2	5	3	1.3688e-11	1.8332e-09	3.5131e-09	2.4165e-07
2	10	5	4.9366e-08	2.6388e-06	7.3010e-05	5.3330e-04
2	15	8	7.0614e-07	2.9725e-04	1.4659e-04	4.4493e-02
2	20	9	Inf	Inf	NaN	NaN
3	20	6	1.5874e-08	1.4165e-06	4.7337e-05	8.9382e-04
4	20	5	8.4712e-12	4.6565e-11	9.0309e-09	3.7456e-09
5	20	5	1.6879e-12	5.9416e-11	1.9510e-09	1.3243e-08
5	50	5	1.1079e-10	6.6070e-10	3.1709e-07	6.6913e-08
5	100	6	2.9307e-09	1.9431e-08	1.0034e-05	1.3912e-06
5	150	8	1.3142e-08	8.4199e-08	5.7281e-06	4.3975e-06

Some Results

Random frequencies, purely imaginary, 100 tests

parameters			average error		max error	
s	# freq.	n	coeff	freq	coeff	freq
2	10	5	1.3476e-14	3.4744e-13	6.0290e-12	1.3724e-10
2	20	7	2.5148e-14	1.2420e-12	3.2103e-11	7.8847e-10
2	50	11	5.9357e-14	3.9721e-12	1.1845e-10	5.5214e-09
2	100	15	9.0480e-13	5.7684e-11	8.8308e-09	2.0468e-07
5	100	6	2.3796e-15	4.3794e-15	3.1431e-11	3.2918e-14
5	150	8	2.3954e-15	4.7773e-15	1.1702e-11	6.9726e-14

Some Results

Line through origin, top real, bottom imaginary, 100 tests

parameters			average error		max error	
s	# freq.	fail	coeff	freq	coeff	freq
2	3	0	5.1668e-06	1.6241e-04	0.0023195	0.0243550
2	4	0	2.8912e-06	2.9318e-03	9.9505e-04	5.8547e-01
2	5	5	3.1901e-05	2.1641e-02	0.0058405	1.8753920
2	10	100	∅	∅	∅	∅
3	5	5	1.4484e-05	2.8744e-02	0.0016677	1.5547492
3	4	14	2.1197e-05	1.2439e-01	4.2678e-03	1.8590e+01
3	5	24	3.8699e-04	3.9617e-02	0.057250	1.326782
2	5	0	2.1330e-12	3.6481e-11	3.9225e-10	4.5345e-09
2	10	0	3.1867e-06	5.9222e-03	0.0018326	1.7269961
2	20	11	8.0145e-06	4.2270e-03	0.0071972	1.0316399
3	10	1	0.0025437	0.0206981	1.0404	4.3874

Oligonomials

UNIVERSITÄT PASSAU

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o-$ $\simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider

$$F_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{l} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$

3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$;
4. Round to next integer: $A = [\Xi^{-T} \Omega]$.

Oligonomials

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha x^\alpha.$$

$O\lambda\iota\gamma o- \simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider
$$F_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{l} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$
3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$:
4. Round to next integer: $A = [\Xi^{-T}\Omega]$.

Oligonomials

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o- \simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider

$$F_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{c} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$

3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$:
4. Round to next integer: $A = [\Xi^{-T} \Omega]$.

Oligonomials

UNIVERSITÄT PASSAU

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o- \simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider

$$\boldsymbol{F}_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{c} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$

3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$:
4. Round to next integer: $A = [\Xi^{-T} \Omega]$.

Oligonomials

UNIVERSITÄT PASSAU

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o-$ $\simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider
$$F_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{l} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$
3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$:
4. Round to next integer: $A = [\Xi^{-T} \Omega]$.

Oligonomials

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o- \simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider
$$\boldsymbol{F}_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{l} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$
3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$: *exponential grid*.
4. Round to next integer: $A = [\Xi^{-T} \Omega]$.

Oligonomials

UNIVERSITÄT PASSAU

Recall

Sparse polynomials

$$f(x) = \sum_{\alpha \in A} f_\alpha \, x^\alpha.$$

$O\lambda\iota\gamma o- \simeq$ "few"

Simple transformation

1. Pick any invertible $\Xi \in \mathbb{C}^{s \times s}$.
2. Consider
$$\boldsymbol{F}_{n,k}(\Xi) := \left[e^{\Xi(\beta+\gamma)} : \begin{array}{l} |\beta| \le n \\ |\gamma| \le k \end{array} \right].$$
3. Prony situation with $\Omega = \Xi^T A = \{\Xi^T \alpha : \alpha \in A\}$: *exponential grid*.
4. Round to next integer: $A = [\Xi^{-T}\Omega]$.

Summary

Prony's method is ...

1. ...nonlinear algebra by linear means.
2. ...really implementable.
3. ...fast and surprisingly stable.
4. ...polynomial interpolation.

Summary

Prony's method is …

1. …nonlinear algebra by linear means.
2. …really implementable.
3. …fast and surprisingly stable.
4. …polynomial interpolation.

Summary

UNIVERSITÄT PASSAU

Prony's method is …

1. …nonlinear algebra by linear means.
2. …really implementable.
3. …fast and surprisingly stable.
4. …polynomial interpolation.

Summary

Prony's method is . . .

1. . . . nonlinear algebra by linear means.
2. . . . really implementable.
3. . . . fast and surprisingly stable.
4. . . . polynomial interpolation.

Summary

Prony's method is . . .

1. . . . nonlinear algebra by linear means.
2. . . . really implementable.
3. . . . fast and surprisingly stable.
4. . . . polynomial interpolation.

Summary

Prony's method is . . .

1. . . . nonlinear algebra by linear means.
2. . . . really implementable.
3. . . . fast and surprisingly stable.
4. . . . polynomial interpolation.

Thank you!