polynomial matrices:
kernel bases, quasi-linear GCD, and applications
introduction

shifted reduced forms

fast algorithms

applications
introduction

⇓ earlier in the course ⇓

⇓ in this lecture ⇓
addition $f + g$, multiplication $f \ast g$

division with remainder $f = qg + r$

truncated inverse $f^{-1} \mod X^d$

extended GCD $uf + vg = \gcd(f, g)$

multipoint eval. $f \mapsto f(\alpha_1), \ldots, f(\alpha_d)$

interpolation $f(\alpha_1), \ldots, f(\alpha_d) \mapsto f$

Padé approximation $f = \frac{p}{q} \mod X^d$

minpoly of linearly recurrent sequence
introduction

⇓ earlier in the course ⇓

$O(M(d))$

- addition $f + g$, multiplication $f * g$
- division with remainder $f = qg + r$
- truncated inverse $f^{-1} \mod X^d$
- extended GCD $uf + vg = \gcd(f, g)$

$O(M(d) \log(d))$

- multipoint eval. $f \mapsto f(\alpha_1), \ldots, f(\alpha_d)$
- interpolation $f(\alpha_1), \ldots, f(\alpha_d) \mapsto f$
- Padé approximation $f = \frac{p}{q} \mod X^d$
- minpoly of linearly recurrent sequence

⇓ in this lecture ⇓
introduction

\[ \Downarrow \text{earlier in the course} \Downarrow \]

\[ \begin{align*}
O(M(d)) \\
\text{addition } f + g, \text{ multiplication } f \cdot g \\
\text{division with remainder } f = qg + r \\
\text{truncated inverse } f^{-1} \mod X^d \\
\text{extended GCD } uf + vg = \gcd(f, g)
\end{align*} \]

\[ \begin{align*}
O(M(d) \log(d)) \\
\text{multipoint eval. } f \mapsto f(\alpha_1), \ldots, f(\alpha_d) \\
\text{interpolation } f(\alpha_1), \ldots, f(\alpha_d) \mapsto f \\
\text{Padé approximation } f = \frac{p}{q} \mod X^d \\
\text{minpoly of linearly recurrent sequence}
\end{align*} \]

\[ \Downarrow \text{in this lecture} \Downarrow \]

**Padé approximation, sequence minpoly, extended GCD**

\[ O(M(d) \log(d)) \text{ operations in } K \]

**matrix versions of these problems**

\[ O(m^\omega M(d) \log(d)) \text{ operations in } K \]

or a tiny bit more for matrix-GCD
given power series \( p(X) \) and \( q(X) \) over \( \mathbb{K} \) at precision \( d \),
with \( q(X) \) invertible,
→ compute \( \frac{p(X)}{q(X)} \mod X^d \)
given power series $p(X)$ and $q(X)$ over $\mathbb{K}$ at precision $d$, with $q(X)$ invertible, compute $\frac{p(X)}{q(X)} \mod X^d$.

algo?? $O(??)$
inv+mul: $O(M(d))$
given power series $p(X)$ and $q(X)$ over $\mathbb{K}$ at precision $d$, with $q(X)$ invertible, 
→ compute $\frac{p(X)}{q(X)} \mod X^d$ 

given $M(X) \in \mathbb{K}[X]$ of degree $d > 0$, given polynomials $p(X)$ and $q(X)$ over $\mathbb{K}$ of degree $< d$, with $q(X)$ invertible modulo $M(X)$, 
→ compute $\frac{p(X)}{q(X)} \mod M(X)$
given power series $p(X)$ and $q(X)$ over $\mathbb{K}$ at precision $d$, with $q(X)$ invertible,
→ compute $\frac{p(X)}{q(X)} \mod X^d$

given $M(X) \in \mathbb{K}[X]$ of degree $d > 0$, given polynomials $p(X)$ and $q(X)$ over $\mathbb{K}$ of degree $< d$, with $q(X)$ invertible modulo $M(X)$, what does that mean?
→ compute $\frac{p(X)}{q(X)} \mod M(X)$

$\text{algo?? } O(??) \quad \text{inv+mul: } O(M(d))$

$xgcd+mul+rem \quad O(M(d) \log(d))$
given power series $p(X)$ and $q(X)$ over $\mathbb{K}$ at precision $d$, with $q(X)$ invertible,
→ compute $\frac{p(X)}{q(X)} \mod X^d$algo?? $O(??)$

inv+mul: $O(M(d))$

given $M(X) \in \mathbb{K}[X]$ of degree $d > 0$,
given polynomials $p(X)$ and $q(X)$ over $\mathbb{K}$ of degree $< d$,
with $q(X)$ invertible modulo $M(X)$, what does that mean?
→ compute $\frac{p(X)}{q(X)} \mod M(X)$algo?? $O(??)$

xgcd+mul+rem $O(M(d) \log(d))$

given $M(X) = (X - \alpha_1) \cdots (X - \alpha_d) \in \mathbb{K}[X]$, for pairwise distinct $\alpha_1, \ldots, \alpha_d \in \mathbb{K}$,
given polynomials $p(X)$ and $q(X)$ over $\mathbb{K}$ of degree $< d$,
with $q(X)$ invertible modulo $M(X)$, what does that mean?
→ compute $\frac{p(X)}{q(X)} \mod M(X)$algo?? $O(??)$
given **power series** \( p(X) \) and \( q(X) \) over \( \mathbb{K} \) at precision \( d \),
with \( q(X) \) invertible,
\[ \rightarrow \text{compute} \quad \frac{p(X)}{q(X)} \mod X^d \quad \text{algo?? } O(??) \]
\[ \text{inv+mul: } O(M(d)) \]

given \( M(X) \in \mathbb{K}[X] \) of degree \( d > 0 \),
given **polynomials** \( p(X) \) and \( q(X) \) over \( \mathbb{K} \) of degree \( < d \),
with \( q(X) \) invertible modulo \( M(X) \),
\[ \rightarrow \text{compute} \quad \frac{p(X)}{q(X)} \mod M(X) \quad \text{what does that mean?} \]
\[ \text{xgcd+mul+rem } O(M(d) \log(d)) \]

given \( M(X) = (X - \alpha_1) \cdots (X - \alpha_d) \in \mathbb{K}[X] \),
for pairwise distinct \( \alpha_1, \ldots, \alpha_d \in \mathbb{K} \),
given **polynomials** \( p(X) \) and \( q(X) \) over \( \mathbb{K} \) of degree \( < d \),
with \( q(X) \) invertible modulo \( M(X) \),
\[ \rightarrow \text{compute} \quad \frac{p(X)}{q(X)} \mod M(X) \quad \text{what does that mean?} \]
\[ \text{eval+div+interp } O(M(d) \log(d)) \]
Generating series of LRS and rational functions

**Theorem**

Given a monic polynomial $P$ of degree $d$, a sequence $(a_n)_{n \in \mathbb{N}}$, and the series $A = \sum_{n \in \mathbb{N}} a_n x^n$, both following assertions are equivalent:

1. $(a_n)_{n \in \mathbb{N}}$ is an LRS with characteristic polynomial $P$;
2. there exists $N \in \mathbb{K}[X]$ of degree $\leq d$ such that $A = N/\text{rec } P$ in $\mathbb{K}[X]$.

When these assertions hold, if moreover $P$ is the minimal polynomial of $(a_n)_{n \in \mathbb{N}}$, then

$$d = \max\{1 + \deg N, \deg \text{rec } P\} := m \quad \text{and} \quad \gcd(N, \text{rec } P) = 1.$$
introduction

rational approximation and interpolation

linearily recurrent sequences – reminder from October 21

Generating series of LRS and rational functions

Theorem

Given a monic polynomial $P$ of degree $d$, a sequence $(a_n)_{n \in \mathbb{N}}$, and the series $A = \sum_{n \in \mathbb{N}} a_n X^n$, both following assertions are equivalent:

1. $(a_n)_{n \in \mathbb{N}}$ is an LRS with characteristic polynomial $P$;
2. there exists $N \in \mathbb{K}[X]$ of degree $< d$ such that $A = N / \text{rec} \, P$ in $\mathbb{K}[[X]]$.

When these assertions hold, if moreover $P$ is the minimal polynomial of $(a_n)_{n \in \mathbb{N}}$, then

$$d = \max\{1 + \deg N, \deg \text{rec} \, P\} := m \quad \text{and} \quad \gcd(N, \text{rec} \, P) = 1.$$

expand $\frac{\text{rev}(P)}{N} \mod X^\delta$

numerator $N$ and charpoly $P$

first $\delta$ terms of the LRS $(a_n)_{n \in \mathbb{N}}$
introduction

rational approximation and interpolation

linearily recurrent sequences – reminder from October 21

Generating series of LRS and rational functions

Theorem
Given a monic polynomial $P$ of degree $d$, a sequence $(a_n)_{n \in \mathbb{N}}$, and the series $A = \sum_{n \in \mathbb{N}} a_n X^n$, both following assertions are equivalent:
1. $(a_n)_{n \in \mathbb{N}}$ is an LRS with characteristic polynomial $P$;
2. there exists $N \in \mathbb{K}[X]$ of degree $< d$ such that $A = N / \text{rec } P$ in $\mathbb{K}[X]$.

When these assertions hold, if moreover $P$ is the minimal polynomial of $(a_n)_{n \in \mathbb{N}}$, then
$$d = \max\{1 + \deg N, \deg \text{rec } P\} := m \quad \text{and} \quad \gcd(N, \text{rec } P) = 1.$$
Padé approximation:
given power series $f(X)$ at precision $d$,
→ compute $p(X), q(X)$ such that $f = \frac{p}{q} \mod X^d$
introduction

rational approximation and interpolation

**Padé approximation:**

given *power series* $f(X)$ at precision $d$,

→ compute $p(X)$, $q(X)$ such that $f = \frac{p}{q} \mod X^d$

opinions on this algorithmic problem?
Padé approximation:
given power series $f(X)$ at precision $d$,
given degree constraints $d_1, d_2 > 0$,
→ compute polynomials $(p(X), q(X))$ of degrees $< (d_1, d_2)$
and such that $f = \frac{p}{q} \text{ mod } X^d$
Padé approximation:
given power series \( f(X) \) at precision \( d \),
given degree constraints \( d_1, d_2 > 0 \),
→ compute polynomials \((p(X), q(X))\) of degrees < \((d_1, d_2)\)
and such that \( f = \frac{p}{q} \mod X^d \)

Cauchy interpolation:
given \( M(X) = (X - \alpha_1) \cdots (X - \alpha_d) \in \mathbb{K}[X] \),
for pairwise distinct \( \alpha_1, \ldots, \alpha_d \in \mathbb{K} \),
given degree constraints \( d_1, d_2 > 0 \),
→ compute polynomials \((p(X), q(X))\) of degrees < \((d_1, d_2)\)
and such that \( f = \frac{p}{q} \mod M(X) \)
Padé approximation:
given power series $f(X)$ at precision $d$,
given degree constraints $d_1, d_2 > 0$,
$\rightarrow$ compute polynomials $(p(X), q(X))$ of degrees $< (d_1, d_2)$
and such that $f = \frac{p}{q} \mod X^d$

Cauchy interpolation:
given $M(X) = (X - \alpha_1) \cdots (X - \alpha_d) \in \mathbb{K}[X]$,
for pairwise distinct $\alpha_1, \ldots, \alpha_d \in \mathbb{K}$,
given degree constraints $d_1, d_2 > 0$,
$\rightarrow$ compute polynomials $(p(X), q(X))$ of degrees $< (d_1, d_2)$
and such that $f = \frac{p}{q} \mod M(X)$

- degree constraints specified by the context
- usual choices have $d_1 + d_2 \approx d$ and existence of a solution
\[ K = \mathbb{F}_7 \]
\[ f = 2X^7 + 2X^6 + 5X^4 + 2X^2 + 4 \]
\[ d = 8, d_1 = 3, d_2 = 6 \]
\[ \rightarrow \text{look for } (p, q) \text{ of degree } < (3, 6) \text{ such that } f = \frac{p}{q} \mod X^8 \]

\[
\begin{bmatrix}
    q & p \\
    -1 & 1
\end{bmatrix}
\begin{bmatrix}
    f
\end{bmatrix}
= 0 \mod X^8
\]
\[ \mathbb{K} = \mathbb{F}_7 \]
\[ f = 2X^7 + 2X^6 + 5X^4 + 2X^2 + 4 \]
\[ d = 8, \ d_1 = 3, \ d_2 = 6 \]
\[ \rightarrow \ \text{look for} \ (p, \ q) \ \text{of degree} \ < (3, 6) \ \text{such that} \ f = \frac{p}{q} \ \text{mod} \ X^8 \]

\[
\begin{bmatrix}
q & p \\
-1 & 0
\end{bmatrix}
\begin{bmatrix}
f
\end{bmatrix}
= 0 \ \text{mod} \ X^8
\]

\[
\begin{bmatrix}
4 & 0 & 2 & 0 & 5 & 0 & 2 & 2 \\
4 & 0 & 2 & 0 & 5 & 0 & 2 \\
4 & 0 & 2 & 0 & 5 & 0 \\
4 & 0 & 2 & 0 \\
4 & 0 & 2 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0
\end{bmatrix}
= 0
\]
$K = \mathbb{F}_7$

$f = 2X^7 + 2X^6 + 5X^4 + 2X^2 + 4$

$d = 8, d_1 = 3, d_2 = 6$

→ look for $(p, q)$ of degree $< (3, 6)$ such that $f = \frac{p}{q} \mod X^8$

\[
\begin{bmatrix}
q & p
\end{bmatrix}
\begin{bmatrix}
f
\end{bmatrix}
= 0 \mod X^8
\]

\[
\begin{bmatrix}
q_0 & q_1 & q_2 & q_3 & q_4 & 1 & p_0 & p_1 & p_2
\end{bmatrix}
= 0
\]
Sur la généralisation des fractions continues algébriques;

PAR M. H. PADÉ,

Docteur ès Sciences mathématiques,
Professeur au lycée de Lille.

[1894, Journal de mathématiques pures et appliquées]

INTRODUCTION.

M. Hermite s’est, dans un travail récemment paru (1), occupé de la généralisation des fractions continues algébriques. La question est de déterminer les polynomes $X_1, X_2, \ldots, X_n$, de degrés $\mu_1, \mu_2, \ldots, \mu_n$, qui satisfont à l’équation

$$S_1 X_1 + S_2 X_2 + \ldots + S_n X_n = S x^{\mu_1 + \mu_2 + \ldots + \mu_n + n - 1},$$

$S_1, S_2, \ldots, S_n$ étant des séries entières données, et $S$ une série égale-ment entière. Ou plutôt, il s’agit d’obtenir un algorithme qui permette le calcul de proche en proche de ces systèmes de $n$ polynomes, et qui soit analogue à l’algorithme par lequel le numérateur et le dénomina-teur d’une réduite d’une fraction continue se déduisent des numéra-teurs et dénominateurs des réduites précédentes. D’élégantes consi-
Hermite-Padé approximation

[Hermite 1893, Padé 1894]

input:
- polynomials $f_1, \ldots, f_m \in K[X]$
- precision $d \in \mathbb{Z}_{>0}$
- degree bounds $d_1, \ldots, d_m \in \mathbb{Z}_{>0}$

output:
polynomials $p_1, \ldots, p_m \in K[X]$ such that
- $p_1 f_1 + \cdots + p_m f_m = 0 \mod X^d$
- $\text{cdeg}([p_1 \cdots p_m]) < (d_1, \ldots, d_m)$

(Padé approximation: particular case $m = 2$ and $f_2 = -1$)
M-Padé approximation / vector rational interpolation
[Cauchy 1821, Mahler 1968]

input:
▶ polynomials $f_1, \ldots, f_m \in \mathbb{K}[X]$
▶ pairwise distinct points $\alpha_1, \ldots, \alpha_d \in \mathbb{K}$
▶ degree bounds $d_1, \ldots, d_m \in \mathbb{Z}_{>0}$

output:
polynomials $p_1, \ldots, p_m \in \mathbb{K}[X]$ such that
▶ $p_1(\alpha_i)f_1(\alpha_i) + \cdots + p_m(\alpha_i)f_m(\alpha_i) = 0$ for all $1 \leq i \leq d$
▶ $\text{cdeg}([p_1 \cdots p_m]) < (d_1, \ldots, d_m)$

(rational interpolation: particular case $m = 2$ and $f_2 = -1$)
in this lecture: modular equation and fast algebraic algorithms


input:

- polynomials $f_1, \ldots, f_m \in \mathbb{K}[X]$
- field elements $\alpha_1, \ldots, \alpha_d \in \mathbb{K}$  \hspace{1cm} \leadsto \text{not necessarily distinct}
- degree bounds $d_1, \ldots, d_m \in \mathbb{Z}_{>0}$  \hspace{1cm} \leadsto \text{general “shift” } s \in \mathbb{Z}^m$

output:

polynomials $p_1, \ldots, p_m \in \mathbb{K}[X]$ such that

- $p_1 f_1 + \cdots + p_m f_m = 0 \mod \prod_{1 \leq i \leq d} (X - \alpha_i)$
- $\mathrm{cdeg}([p_1 \cdots p_m]) < (d_1, \ldots, d_m)$  \hspace{1cm} \leadsto \text{minimal } s\text{-row degree}$

(Hermite-Padé: $\alpha_1 = \cdots = \alpha_d = 0$; interpolation: pairwise distinct points)
application of vector rational interpolation:
given pairwise distinct points \( \{(\alpha_i, \beta_i), 1 \leq i \leq 8\} \)
\( = \{(24, 80), (31, 73), (15, 73), (32, 35), (83, 66), (27, 46), (20, 91), (59, 64)\} \),
compute a **bivariate** polynomial \( p(X, Y) \in \mathbb{K}[X, Y] \)
such that \( p(\alpha_i, \beta_i) = 0 \) for \( 1 \leq i \leq 8 \)

\[
M(X) = (X - 24) \cdots (X - 59) \\
L(X) = \text{Lagrange interpolant}
\]

\( \rightarrow \) solutions = ideal \( \langle M(X), Y - L(X) \rangle \)

solutions of smaller \( X \)-degree: \( p(X, Y) = p_0(X) + p_1(X)Y + p_2(X)Y^2 \)

\[
p(X, L(X)) = \begin{bmatrix} p_0 & p_1 & p_2 \end{bmatrix} \begin{bmatrix} 1 \\ L \\ L^2 \end{bmatrix} = 0 \text{ mod } M(X)
\]

- instance of **univariate** rational vector interpolation
- with a **structured** input equation (powers of \( L \text{ mod } M \))
application of vector rational interpolation:
given pairwise distinct points \( \{ (\alpha_i, \beta_i), 1 \leq i \leq 8 \} \)
\[= \{(24, 80), (31, 73), (15, 73), (32, 35), (83, 66), (27, 46), (20, 91), (59, 64)\}, \]
compute a \textbf{bivariate} polynomial \( p(X, Y) \in K[X, Y] \)
such that \( p(\alpha_i, \beta_i) = 0 \) for \( 1 \leq i \leq 8 \)

add \textbf{degree constraints}: seek \( p(X, Y) \) of the form
\[
p_{00} + p_{01}X + p_{02}X^2 + p_{03}X^3 + p_{04}X^4 + (p_{10} + p_{11}X + p_{12}X^2)Y + p_{20}Y^2:
\]

\[
\begin{bmatrix}
p_{00} & p_{01} & p_{02} & p_{03} & p_{04} & p_{10} & p_{11} & p_{12} & p_{20}
\end{bmatrix}
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
\alpha_1 & \alpha_2 & \cdots & \alpha_8 \\
\alpha_1^2 & \alpha_2^2 & \cdots & \alpha_8^2 \\
\alpha_1^3 & \alpha_2^3 & \cdots & \alpha_8^3 \\
\alpha_1^4 & \alpha_2^4 & \cdots & \alpha_8^4 \\
\beta_1 & \beta_2 & \cdots & \beta_8 \\
\alpha_1 \beta_1 & \alpha_2 \beta_2 & \cdots & \alpha_8 \beta_8 \\
\alpha_1^2 \beta_1 & \alpha_2^2 \beta_2 & \cdots & \alpha_8^2 \beta_8 \\
\beta_1^2 & \beta_2^2 & \cdots & \beta_8^2
\end{bmatrix} = 0
\]

\begin{itemize}
  \item \( K \)-linear system
  \item \textbf{two levels} of structure
\end{itemize}

\[p(X, Y) = (2X^4 + 56X^3 + 42X^2 + 48X + 15) + (72X^2 + 12X + 30)Y + Y^2\]
polynomial matrices: reminder and motivation

why polynomial matrices here?
omitting degree constraints, the set of solutions is
\[ S = \{(p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M}\]

recall \( M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \)
omitting degree constraints, the set of solutions is
\[ S = \{ (p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M \} \]

\[ M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \]

\( S \) is a “free \( K[X] \)-module of rank \( m \)”, meaning:
- stable under \( K[X] \)-linear combinations
- admits a basis consisting of \( m \) elements
- basis = \( K[X] \)-linear independence + generates all solutions
omitting degree constraints, the set of solutions is
\[ S = \{(p_1, \ldots, p_m) \in \mathbb{K}[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M\} \]

recall \( M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \)

\( S \) is a “free \( \mathbb{K}[X] \)-module of rank \( m \)”, meaning:
- stable under \( \mathbb{K}[X] \)-linear combinations
- admits a basis consisting of \( m \) elements
- basis = \( \mathbb{K}[X] \)-linear independence + generates all solutions

\[ \Rightarrow S \subset \mathbb{K}[X]^m \Rightarrow S \text{ has rank } \leq m \]
\[ \Rightarrow M(X)\mathbb{K}[X]^m \subset S \Rightarrow S \text{ has rank } \geq m \]

remark: solutions are not considered modulo \( M \)
e.g. \((M, 0, \ldots, 0)\) is in \( S \) and may appear in a basis
omitting degree constraints, the set of solutions is
\[ S = \{(p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \text{ mod } M\} \]

why polynomial matrices here?

<table>
<thead>
<tr>
<th>basis of solutions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ square nonsingular matrix ( P ) in ( K[X]^{m \times m} )</td>
</tr>
<tr>
<td>▶ each row of ( P ) is a solution</td>
</tr>
<tr>
<td>▶ any solution is a ( K[X] )-combination ( uP, u \in K[X]^{1 \times m} )</td>
</tr>
</tbody>
</table>

i.e. \( S \) is the \( K[X] \)-row space of \( P \)
omitting degree constraints, the set of solutions is
\[ S = \{ (p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M \} \]
recall \( M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \)

\textbf{why polynomial matrices here?}

\textbf{basis of solutions:}
- square nonsingular matrix \( P \) in \( K[X]^{m \times m} \)
- each row of \( P \) is a solution
- any solution is a \( K[X] \)-combination \( uP, u \in K[X]^{1 \times m} \)

\( i.e. \ S \) is the \( K[X] \)-row space of \( P \)

\textbf{prove:} \( \det(P) \) is a divisor of \( M(X)^m \)
polynomial matrices: reminder and motivation

why polynomial matrices here?

omitting degree constraints, the set of solutions is
\[ S = \left\{ (p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M \right\} \]

recall \( M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \)

basis of solutions:
- square nonsingular matrix \( P \) in \( K[X]^{m \times m} \)
- each row of \( P \) is a solution
- any solution is a \( K[X] \)-combination \( uP, u \in K[X]^{1 \times m} \)

i.e. \( S \) is the \( K[X] \)-row space of \( P \)

prove: \( \det(P) \) is a divisor of \( M(X)^m \)

prove: any other basis is \( UP \) for \( U \in K[X]^{m \times m} \) with \( \det(U) \in K \setminus \{0\} \)
omitting degree constraints, the set of solutions is
\[ S = \{(p_1, \ldots, p_m) \in K[X]^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \mod M\} \]

recalling \( M(X) = \prod_{1 \leq i \leq d} (X - \alpha_i) \)

**basis of solutions:**
- square nonsingular matrix \( P \) in \( K[X]^{m \times m} \)
- each row of \( P \) is a solution
- any solution is a \( K[X] \)-combination \( uP, u \in K[X]^{1 \times m} \)

i.e. \( S \) is the \( K[X] \)-row space of \( P \)

computing a basis of \( S \) with “minimal degrees”
- has many more applications than a single small-degree solution
- is in most cases the fastest known strategy anyway(!)

\( \rightsquigarrow \) degree minimality ensured via shifted reduced forms
introduction

polynomial matrices: reminder and motivation

\[
A = \begin{bmatrix}
3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\
5 & 5X^2 + 3X + 1 & 5X + 3 \\
3X^3 + X^2 + 5X + 3 & 6X + 5 & 2X + 1
\end{bmatrix} \in \mathbb{K}[X]^{3\times 3}
\]

3 × 3 matrix of degree 3 with entries in \( \mathbb{K}[X] = \mathbb{F}_7[X] \)

operations in \( \mathbb{K}[X]^{m\times m}_{<d} \):

- combination of matrix and polynomial computations
- addition in \( O(m^2d) \), naive multiplication in \( O(m^3d^2) \)
- some tools shared with \( \mathbb{K} \)-matrices, others specific to \( \mathbb{K}[X] \)-matrices

[Contor-Kaltofen'91]

multiplication in \( O(m^\omega d \log(d) + m^2d \log(d) \log \log(d)) \)

\( \in O(m^\omega M(d)) \subset O^*(m^\omega d) \)
introduction

polynomial matrices: reminder and motivation

\[
A = \begin{bmatrix}
3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\
5 & 5X^2 + 3X + 1 & 5X + 3 \\
3X^3 + X^2 + 5X + 3 & 6X + 5 & 2X + 1
\end{bmatrix} \in \mathbb{K}[X]^{3 \times 3}
\]

3 × 3 matrix of degree 3 with entries in \( \mathbb{K}[X] = \mathbb{F}_7[X] \)

operations in \( \mathbb{K}[X]_{<d}^{m \times m} \):

- combination of matrix and polynomial computations
- addition in \( O(m^2d) \), naive multiplication in \( O(m^3d^2) \)
- some tools shared with \( \mathbb{K} \)-matrices, others specific to \( \mathbb{K}[X] \)-matrices

[Cantor-Kaltofen'91]

multiplication in \( O(m^\omega d \log(d) + m^2d \log(d) \log \log(d)) \)

\( \in O(m^\omega M(d)) \subset O^\sim(m^\omega d) \)

- Newton truncated inversion, matrix-QuoRem \( \rightarrow \) fast \( O^\sim(m^\omega d) \)
- inversion and determinant via evaluation-interpolation \( \rightarrow \) medium \( O^\sim(m^{\omega+1}d) \)
- vector rational approximation & interpolation \( \rightarrow \) ???
reductions of most problems to polynomial matrix multiplication
matrix $m \times m$ of degree $d$
of “average” degree $D$
$\rightarrow O(\omega d)$
$\frac{D}{m} \rightarrow O(\omega \frac{D}{m})$

classical matrix operations
- multiplication
- kernel, system solving
- rank, determinant
- inversion $O(3d)$

univariate specific operations
- truncated inverse, QuoRem
- Hermite-Padé approximation
- vector rational interpolation
- syzygies / modular equations

transformation to normal forms
- triangularization: Hermite form
- row reduction: Popov form
- diagonalization: Smith form
polynomial matrices: reminder and motivation

reductions of most problems to polynomial matrix multiplication
matrix $m \times m$ of degree $d$
of “average” degree $\frac{D}{m} \rightarrow O(\omega^d m)$
$\rightarrow O(\omega^D m)$

classical matrix operations
▶ multiplication
▶ kernel, system solving
▶ rank, determinant
▶ inversion $O(\omega^3 m)$

univariate specific operations
▶ truncated inverse, QuoRem
▶ Hermite-Padé approximation
▶ vector rational interpolation
▶ syzygies / modular equations

transformation to normal forms
▶ triangularization: Hermite form
▶ row reduction: Popov form
▶ diagonalization: Smith form
polynomial matrices: reminder and motivation

Reductions of most problems to polynomial matrix multiplication

Matrix $m \times m$ of degree $d$ of “average” degree $\frac{D}{m} \rightarrow O(\tilde{m}^\omega d)$

Classical matrix operations
- Multiplication
- Kernel, system solving
- Rank, determinant
- Inversion $O(\tilde{m}^3 d)$

Univariate specific operations
- Truncated inverse, QuoRem
- Hermite-Padé approximation
- Vector rational interpolation
- Syzygies / modular equations

Transformation to normal forms
- Triangularization: Hermite form
- Row reduction: Popov form
- Diagonalization: Smith form
introduction

polynomial matrices: reminder and motivation

reductions of most problems to polynomial matrix multiplication
matrix $m \times m$ of degree $d$ of “average” degree $\frac{D}{m}$
$\rightarrow O(\tilde{m}^\omega d)$
$\rightarrow O(\tilde{m}^\omega \frac{D}{m})$

classical matrix operations
- multiplication
- kernel, system solving
- rank, determinant
- inversion $O(\tilde{m}^3 d)$

univariate specific operations
- truncated inverse, QuoRem
- Hermite-Padé approximation
- vector rational interpolation
- syzygies / modular equations

transformation to normal forms
- triangularization: Hermite form
- row reduction: Popov form
- diagonalization: Smith form
outline

introduction

- rational approximation and interpolation
- the vector case
- pol. matrices: reminders and motivation

shifted reduced forms

fast algorithms

applications
shifted reduced forms

reducedness: examples and properties

notation:

let $A \in \mathbb{K}[X]^{m \times n}$ with no zero row,
define $d = (d_1, \ldots, d_m) = \text{rdeg}(A)$

and $X^d = \begin{bmatrix} X^{d_1} \\ \vdots \\ X^{d_m} \end{bmatrix} \in \mathbb{K}[X]^{m \times m}$

definition: (row-wise) leading matrix

the leading matrix of $A$ is the unique matrix $\text{lm}(A) \in \mathbb{K}^{m \times n}$
such that $A = X^d \text{lm}(A) + R$ with $\text{rdeg}(R) < d$ entry-wise

equivalently, $X^{-d}A = \text{lm}(A) + \text{terms of strictly negative degree}$
shifted reduced forms

reducedness: examples and properties

notation:

let \( A \in \mathbb{K}[X]^{m \times n} \) with no zero row,
define \( d = (d_1, \ldots, d_m) = \text{rdeg}(A) \)
and \( X^d = \begin{bmatrix} X^{d_1} & & \\ & \ddots & \\ & & X^{d_m} \end{bmatrix} \in \mathbb{K}[X]^{m \times m} \)

definition: (row-wise) leading matrix

the leading matrix of \( A \) is the unique matrix \( \text{lm}(A) \in \mathbb{K}^{m \times n} \) such that \( A = X^d \text{lm}(A) + R \) with \( \text{rdeg}(R) < d \) entry-wise

equivalently, \( X^{-d} A = \text{lm}(A) + \) terms of strictly negative degree

definition: (row-wise) reduced matrix

\( A \in \mathbb{K}[X]^{m \times n} \) is said to be reduced if \( \text{lm}(A) \) has full row rank
shifted reduced forms

reducedness: examples and properties

consider the following matrices, with $\mathbb{K} = \mathbb{F}_7$:

$$A_1 = \begin{bmatrix} 3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\ 5 & 5X^2 + 3X + 1 & 5X + 3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 3X + 1 & 4X + 3 & 5X + 5 \\ 0 & 4X^2 + 6X & 5 \\ 4X^2 + 5X + 2 & 5 & 6X^2 + 1 \end{bmatrix}$$

$A_3 = \text{transpose of } A_1$

$A_4 = \text{transpose of } A_2$

answer the following, for $i \in \{1, 2, 3, 4\}$:
1. what is $\text{rdeg}(A_i)$?
2. what is $\text{lm}(A_i)$?
3. is $A_i$ reduced?
let $A \in \mathbb{K}[X]^{m \times n}$ with $m \leq n$, the following are equivalent:

(i) $A$ is reduced (i.e. $\text{Im}(A)$ has full rank)
polynomial matrices in reduced form

reducedness: examples and properties

let $A \in \mathbb{K}[X]^{m \times n}$ with $m \leq n$,
the following are equivalent:

(i) $A$ is reduced (i.e. $\text{Im}(A)$ has full rank)

(ii) for any vector $u = [u_1 \ 1 \ u_2] \in \mathbb{K}[X]^{1 \times m}$ with 1 at index $i$,
$\text{rdeg}(uA) \geq \text{rdeg}(A_{i,*})$
polynomial matrices in reduced form

reducedness: examples and properties

let $A \in \mathbb{K}[X]^{m \times n}$ with $m \leq n$, the following are equivalent:

(i) $A$ is reduced (i.e. $\text{Im}(A)$ has full rank)

(ii) for any vector $u = [u_1 \ 1 \ u_2] \in \mathbb{K}[X]^{1 \times m}$ with 1 at index $i$, $rdeg(uA) \geq rdeg(A_{i,*})$

(iii) predictable degree: for any vector $u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m}$, $rdeg(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + rdeg(A_{i,*}))$
polynomial matrices in reduced form

reducedness: examples and properties

Let \( A \in \mathbb{K}[X]^{m \times n} \) with \( m \leq n \),

the following are equivalent:

(i) \( A \) is reduced (i.e. \( \text{Im}(A) \) has full rank)

(ii) for any vector \( u = [u_1 \ 1 \ u_2] \in \mathbb{K}[X]^{1 \times m} \) with 1 at index \( i \),
\( \text{rdeg}(uA) \geq \text{rdeg}(A_{i,*}) \)

(iii) predictable degree: for any vector \( u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m} \),
\( \text{rdeg}(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + \text{rdeg}(A_{i,*})) \)

(iv) degree minimality: \( \text{rdeg}(A) \precsim \text{rdeg}(UA) \) holds for any nonsingular matrix \( U \in \mathbb{K}[X]^{m \times m} \), where \( \precsim \) sorts the tuples in nondecreasing order and then uses lexicographic comparison.
polynomial matrices in reduced form

reducedness: examples and properties

let \( A \in \mathbb{K}[X]^{m \times n} \) with \( m \leq n \),
the following are equivalent:

(i) \( A \) is reduced (i.e. \( \text{Im}(A) \) has full rank)

(ii) for any vector \( u = [u_1 \ 1 \ u_2] \in \mathbb{K}[X]^{1 \times m} \) with 1 at index \( i \),
\( r\deg(uA) \geq r\deg(A_{i,*}) \)

(iii) predictable degree: for any vector \( u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m} \),
\( r\deg(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + r\deg(A_{i,*})) \)

(iv) degree minimality: \( r\deg(A) \preceq r\deg(UA) \) holds for any nonsingular matrix \( U \in \mathbb{K}[X]^{m \times m} \), where \( \preceq \) sorts the tuples in nondecreasing order and then uses lexicographic comparison

(v) predictable determinantal degree: \( \deg \det(A) = |r\deg(A)| \)
(only when \( m = n \))
shifted reduced forms

reducedness: examples and properties

recall the matrix, with \( K = \mathbb{F}_7 \),

\[
A = \begin{bmatrix}
3X + 1 & 4X + 3 & 5X + 5 \\
0 & 4X^2 + 6X & 5 \\
4X^2 + 5X + 2 & 5 & 6X^2 + 1
\end{bmatrix}
\]

1. what is \( \text{deg det}(A) \)?

2. what is \( \text{rdeg}([4X^2 + 1 \ 2X \ 4X + 5]A) \)?

3. is it possible to find a matrix

\[
P = \begin{bmatrix}
p_{00} & p_{01} & p_{02} \\
p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

whose rank is 2, whose degree is 1, and which is a left-multiple of \( A \)?
shifted reduced forms

reducedness: examples and properties

recall the matrix, with $K = F_7$,
$$A = \begin{bmatrix} 3X + 1 & 4X + 3 & 5X + 5 \\ 0 & 4X^2 + 6X & 5 \\ 4X^2 + 5X + 2 & 5 & 6X^2 + 1 \end{bmatrix}$$

1. what is $\text{deg det}(A)$?

2. what is $\text{rdeg}([4X^2 + 1 \ 2X \ 4X + 5] A)$?

3. is it possible to find a matrix
$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \end{bmatrix}$$
whose rank is 2, whose degree is 1, and which is a left-multiple of $A$?

find a row vector $u$ of degree 1 such that $uA$ has degree 2, where
$$A = \begin{bmatrix} 3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\ 5 & 5X^2 + 3X + 1 & 5X + 3 \end{bmatrix}$$
shifted reduced forms

shifted forms and degree constraints

keeping our problem in mind:

- input: $f_i$'s and $\alpha_i$'s and degree constraints $d_1, \ldots, d_m \in \mathbb{Z}_{>0}$
- output: a solution $p$ satisfying the constraints $c\deg(p) < (d_1, \ldots, d_m)$

**obstacle:**
computing a reduced basis of solutions ignores the constraints

**exercice:** suppose we have a reduced basis $P \in \mathbb{K}[X]^{m \times m}$ of solutions

- think of particular constraints $(d_1, \ldots, d_m)$ that can be handled via $P$
- give constraints $(d_1, \ldots, d_m)$ for which $P$ is “typically” not satisfactory
shifted reduced forms

shifted forms and degree constraints

keeping our problem in mind:

► input: $f_i$'s and $\alpha_i$'s and degree constraints $d_1, \ldots, d_m \in \mathbb{Z}_{>0}$
► output: a solution $p$ satisfying the constraints $\text{cdeg}(p) < (d_1, \ldots, d_m)$

obstacle:
computing a reduced basis of solutions ignores the constraints

exercice: suppose we have a reduced basis $P \in K[X]^{m \times m}$ of solutions

► think of particular constraints $(d_1, \ldots, d_m)$ that can be handled via $P$
► give constraints $(d_1, \ldots, d_m)$ for which $P$ is “typically” not satisfactory

solution: compute $P$ in shifted reduced form
shifted reduced forms

shifted forms and degree constraints

\[
A = \begin{bmatrix}
3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\
5 & 5X^2 + 3X + 1 & 5X + 3 \\
3X^3 + X^2 + 5X + 3 & 6X + 5 & 2X + 1
\end{bmatrix}
\]

using elementary row operations, transform \( A \) into...

**Hermite form**

\[
H = \begin{bmatrix}
X^6 + 6X^4 + X^3 + X + 4 & 0 & 0 \\
5X^5 + 5X^4 + 6X^3 + 2X^2 + 6X + 3 & X & 0 \\
3X^4 + 5X^3 + 4X^2 + 6X + 1 & 5 & 1
\end{bmatrix}
\]

**Popov form**

\[
P = \begin{bmatrix}
X^3 + 5X^2 + 4X + 1 & 2X + 4 & 3X + 5 \\
1 & X^2 + 2X + 3 & X + 2 \\
3X + 2 & 4X & X^2
\end{bmatrix}
\]
shifted reduced forms

shifted forms and degree constraints

nonsingular $A \in K[X]^{m \times m}$

elementary row transformations

Hermite form [Hermite, 1851]
- triangular
- column normalized

$\begin{bmatrix}
16 & 0 \\
15 & 0 \\
15 & 0 \\
15 & 0 \\
\end{bmatrix}$

$\begin{bmatrix}
4 & 7 \\
3 & 7 \\
1 & 5 & 3 \\
3 & 6 & 1 & 2 \\
\end{bmatrix}$

Hermite form

Popov form [Popov, 1972]
- row reduced/distinct pivots
- column normalized

Beckermann-Labahn-Villard, 1999; Mulders-Storjohann, 2003
shifted reduced forms

shifted forms and degree constraints

nonsingular $A \in K[X]^{m \times m}$

elementary row transformations

Hermite form \cite{Hermite, 1851}
- triangular
- column normalized

\[
\begin{bmatrix}
16 & 0 \\
15 & 0 \\
15 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
4 & 7 \\
3 & 1 \\
3 & 6
\end{bmatrix}
\]

Popov form \cite{Popov, 1972}
- row reduced/distinct pivots
- column normalized

\[
\begin{bmatrix}
4 & 3 & 3 & 3 \\
3 & 4 & 3 & 3 \\
3 & 3 & 4 & 3 \\
3 & 3 & 3 & 4
\end{bmatrix}
\quad
\begin{bmatrix}
7 & 0 & 1 & 5 \\
0 & 1 & 0 \\
2 \\
6 & 0 & 1 & 6
\end{bmatrix}
\]
shifted reduced forms

shifted forms and degree constraints

nonsingular $A \in \mathbb{K}[X]^{m \times m}$

elementary row transformations

Hermite form [Hermite, 1851]
- triangular
- column normalized

Popov form [Popov, 1972]
- row reduced/distinct pivots
- column normalized

$\mathbb{K}[X]$-module $S \subset \mathbb{K}[X]^{1 \times m}$ of rank $m$
shifted reduced forms

shifted forms and degree constraints

nonsingular $A \in \mathbb{K}[X]^{m \times m}$

elementary row transformations

Hermite form [Hermite, 1851]
- triangular
- column normalized

<table>
<thead>
<tr>
<th>16</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

| 4 | 7 |
| 3 | 3 |
| 1 | 1 |
| 3 | 2 |

Popov form [Popov, 1972]
- row reduced/distinct pivots
- column normalized

| 4 | 3 | 3 | 3 |
| 4 | 3 | 3 | 3 |
| 3 | 3 | 4 | 3 |
| 3 | 3 | 3 | 4 |

| 7 | 0 | 1 | 5 |
| 0 | 1 | 0 |
| 2 |
| 6 | 0 | 1 | 6 |

invariant: $D = \deg(\det(A)) = 4 + 7 + 3 + 2 = 7 + 1 + 2 + 6$

- average column degree is $\frac{D}{m}$
- size of object is $mD + m^2 = m^2\left(\frac{D}{m} + 1\right)$
shifted reduced forms

shifted forms and degree constraints

nonsingular \( A \in K[X]^{m \times m} \)

elementary row transformations

**Hermite form** [Hermite, 1851]
- triangular
- column normalized

**Popov form** [Popov, 1972]
- row reduced/distinct pivots
- column normalized

\[
\begin{bmatrix}
16 & 0 \\
15 & 0 \\
15 & 0 \\
\end{bmatrix}
\quad \begin{bmatrix}
4 & 3 & 7 \\
3 & 1 & 5 & 3 \\
3 & 6 & 1 & 2 \\
\end{bmatrix}
\quad \begin{bmatrix}
4 & 3 & 3 & 3 \\
3 & 4 & 3 & 3 \\
3 & 3 & 4 & 3 \\
3 & 3 & 3 & 4 \\
\end{bmatrix}
\quad \begin{bmatrix}
7 & 0 & 1 & 5 \\
0 & 1 & 0 & 0 \\
2 & 0 & 1 & 6 \\
\end{bmatrix}
\]

[Beckermann-Labahn-Villard, 1999; Mulders-Storjohann, 2003]

**shifted reduced form:**
*arbitrary* degree constraints + *no* column column normalization

\( \approx \) minimal, non-reduced, \(\prec\)-Gröbner basis
shifted reduced forms

shift: integer tuple $s = (s_1, \ldots, s_m)$ acting as column weights
→ connects Popov and Hermite forms

$\begin{align*}
\text{Popov} & : s = (0, 0, 0, 0) \\
\begin{bmatrix} 4 & 3 & 3 \end{bmatrix} & \begin{bmatrix} 7 & 0 & 1 \end{bmatrix} \\
\begin{bmatrix} 3 & 4 & 3 \\ 3 & 3 & 4 \\ 3 & 3 & 3 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \\
\end{align*}$

$\begin{align*}
\text{s-Popov} & : s = (0, 2, 4, 6) \\
\begin{bmatrix} 7 & 4 & 2 \end{bmatrix} & \begin{bmatrix} 8 & 5 \end{bmatrix} \\
\begin{bmatrix} 6 & 5 & 2 \\ 6 & 4 & 3 \\ 6 & 4 & 2 \end{bmatrix} & \begin{bmatrix} 7 & 6 \\ 2 & 0 \end{bmatrix} \\
\end{align*}$

$\begin{align*}
\text{Hermite} & : s = (0, D, 2D, 3D) \\
\begin{bmatrix} 16 & 0 \end{bmatrix} & \begin{bmatrix} 4 \\ 3 \end{bmatrix} \\
\begin{bmatrix} 15 & 0 \\ 15 & 0 \end{bmatrix} & \begin{bmatrix} 4 & 7 \\ 1 & 5 \end{bmatrix} \\
\end{align*}$

- normal form, average column degree $D/m$
- shifted reduced form: same without normalization
- shifts arise naturally in algorithms (approximants, kernel, \ldots)
shifted row degree of a polynomial matrix
= the list of the maximum shifted degree in each of its rows

for \( A = (a_{i,j}) \in \mathbb{K}[X]^{m \times n} \), and \( s = (s_1, \ldots, s_n) \in \mathbb{Z}^n \),

\[
\text{rdeg}_s(A) = (\text{rdeg}_s(A_{1,*}), \ldots, \text{rdeg}_s(A_{m,*}))
\]

\[
= \left( \max_{1 \leq j \leq n} (\deg(A_{1,j}) + s_j), \ldots, \max_{1 \leq j \leq n} (\deg(A_{m,j}) + s_j) \right) \in \mathbb{Z}^m
\]

example: for the matrix \( A = \begin{bmatrix} 3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\ 5 & 5X^2 + 3X + 1 & 5X + 3 \end{bmatrix} \),

describe \( \text{rdeg}_{(0,0,0)}(A) \), \( \text{rdeg}_{(0,1,2)}(A) \), and \( \text{rdeg}_{(-1,-3,-2)}(A) \)
shifted row degree of a polynomial matrix
= the list of the maximum shifted degree in each of its rows

for $A = (a_{i,j}) \in \mathbb{K}[X]^{m \times n}$, and $s = (s_1, \ldots, s_n) \in \mathbb{Z}^n$,

$$rdeg_s(A) = (rdeg_s(A_{1,*}), \ldots, rdeg_s(A_{m,*}))$$

$$= \left( \max_{1 \leq j \leq n} (\deg(A_{1,j}) + s_j), \ldots, \max_{1 \leq j \leq n} (\deg(A_{m,j}) + s_j) \right) \in \mathbb{Z}^m$$

example: for the matrix $A = \begin{bmatrix} 3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\ 5 & 5X^2 + 3X + 1 & 5X + 3 \end{bmatrix}$,

describe $rdeg_{(0,0,0)}(A)$, $rdeg_{(0,1,2)}(A)$, and $rdeg_{(-1,-3,-2)}(A)$

- $rdeg_s(A) = rdeg(AX^s)$
- $rdeg_s(A)$ only depends on the degrees in $A$
- $rdeg_{s+(c,\ldots,c)}(A) = rdeg_s(A) + c$
shifted reduced forms

shifted forms and degree constraints

notation:

let $A \in \mathbb{K}[X]^{m \times n}$ with no zero row, and $s \in \mathbb{Z}^n$, define $d = (d_1, \ldots, d_m) = \text{rdeg}_s(A)$

and $X^d = \begin{bmatrix} X^{d_1} & \cdots & \cdots & X^{d_m} \end{bmatrix} \in \mathbb{K}[X, X^{-1}]^{m \times m}$

definition: $s$-leading matrix / $s$-reduced matrix

assuming $s \geq 0$,

- the $s$-leading matrix of $A$ is $\text{lm}_s(A) = \text{lm}(AX^s) \in \mathbb{K}^{m \times n}$
- $A \in \mathbb{K}[X]^{m \times n}$ is reduced if $\text{lm}_s(A)$ has full row rank
shifted forms and degree constraints

**notation:**

Let \( A \in \mathbb{K}[X]^{m \times n} \) with no zero row, and \( s \in \mathbb{Z}^n \), define

\[
d = (d_1, \ldots, d_m) = \text{rdeg}_s(A)
\]

and

\[
X^d = \begin{bmatrix}
X^{d_1} \\
\vdots \\
X^{d_m}
\end{bmatrix} \in \mathbb{K}[X, X^{-1}]^{m \times m}
\]

**definition: s-leading matrix / s-reduced matrix**

Assuming \( s \geq 0 \),

- The **s-leading matrix** of \( A \) is \( \text{Im}_s(A) = \text{Im}(AX^s) \in \mathbb{K}^{m \times n} \)
- \( A \in \mathbb{K}[X]^{m \times n} \) is **reduced** if \( \text{Im}_s(A) \) has full row rank

- These notions are invariant under \( s \to s + (c, \ldots, c) \)
- They coincide with the non-shifted case when \( s = (0, \ldots, 0) \)
- \( X^{-d}AX^s = \text{Im}_s(A) + \text{terms of strictly negative degree} \)
shifted reduced forms

shifted forms and degree constraints

exercise: for each of the matrices below, and each shift \( \mathbf{s} \),
1. give the \( \mathbf{s} \)-leading matrix
2. deduce whether the matrix is \( \mathbf{s} \)-reduced

\[
\mathbf{A} = \begin{bmatrix}
3X + 4 & X^3 + 4X + 1 & 4X^2 + 3 \\
5 & 5X^2 + 3X + 1 & 5X + 3 \\
3X^3 + X^2 + 5X + 3 & 6X + 5 & 2X + 1
\end{bmatrix}
\]

\[
\mathbf{H} = \begin{bmatrix}
X^6 + 6X^4 + X^3 + X + 4 & 0 & 0 \\
5X^5 + 5X^4 + 6X^3 + 2X^2 + 6X + 3 & X & 0 \\
3X^4 + 5X^3 + 4X^2 + 6X + 1 & 5 & 1
\end{bmatrix}
\]

\[
\mathbf{P} = \begin{bmatrix}
X^3 + 5X^2 + 4X + 1 & 2X + 4 & 3X + 5 \\
1 & X^2 + 2X + 3 & X + 2 \\
3X + 2 & 4X & X^2
\end{bmatrix}
\]

\( \mathbf{s} = (0, 0, 0), \quad \mathbf{s} = (0, 5, 6), \quad \mathbf{s} = (-3, -2, -2) \)
shifted reduced forms

shifted forms and degree constraints

the characterizations generalize to the \( s \)-shifted case, using \( s \)-row degrees and \( s \)-leading matrices where appropriate

(proofs: direct reductions, with: \( A \) is \( s \)-reduced \( \iff \) \( AX^s \) is reduced)

for example recall the predictable degree property:

\( A \) is reduced if and only if for any \( u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m} \),

\[
\text{rdeg}(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + \text{rdeg}(A_{i,*}))
\]
shifted reduced forms

shifted forms and degree constraints

the characterizations generalize to the \( s \)-shifted case, using \( s \)-row degrees and \( s \)-leading matrices where appropriate

(proofs: direct reductions, with: \( A \) is \( s \)-reduced \( \iff \) \( AX^s \) is reduced)

for example recall the predictable degree property:

\( A \) is reduced if and only if for any \( u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m} \),

\[
\text{rdeg}(uA) = \max_{1 \leq i \leq m} (\text{deg}(u_i) + \text{rdeg}(A_{i,*}))
\]

\( \implies \)

- this means \( \text{rdeg}(uA) = \text{rdeg}_t(u) \) where \( t = \text{rdeg}(A) \)

- i.e. \( \text{rdeg}(uA) = \text{rdeg}(uX^{\text{rdeg}(A)}) \), “no surprising cancellation”

- proof: let \( \delta = \text{rdeg}_t(u) \), our goal is to show \( \text{rdeg}(uA) = \delta \) terms of \( X^{-\delta}uA \) have degree \( \leq 0 \), and \( X^{-\delta}uA = (X^{-\delta}uX^t)(X^{-t}A) \);

the term of degree 0 is \( \text{Im}_t(u)\text{Im}(A) \), it is nonzero since \( \text{Im}(A) \) has full rank and \( \text{Im}_t(u) \neq 0 \) (the case \( u = 0 \) is trivial)
shifted reduced forms

shifted forms and degree constraints

the characterizations generalize to the $s$-shifted case, using $s$-row degrees and $s$-leading matrices where appropriate

(proofs: direct reductions, with: $A$ is $s$-reduced $\iff AX^s$ is reduced)

for example recall the predictable degree property:

$A$ is reduced if and only if for any $u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m}$,

$$rdeg(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + rdeg(A_{i,*}))$$

$A$ is $s$-reduced if and only if for any $u = [u_1 \cdots u_m] \in \mathbb{K}[X]^{1 \times m}$,

$$rdeg_s(uA) = \max_{1 \leq i \leq m} (\deg(u_i) + rdeg_s(A_{i,*}))$$

this means $rdeg_s(uA) = rdeg_t(u)$, where $t = rdeg_s(A)$
shifted reduced forms

shifted forms and degree constraints

the characterizations generalize to the \( s \)-shifted case, using \( s \)-row degrees and \( s \)-leading matrices where appropriate

(proofs: direct reductions, with: \( A \) is \( s \)-reduced \( \iff \) \( AX^s \) is reduced)

for example recall the predictable degree property:

\[ A \text{ is reduced if and only if for any } u = [u_1 \cdots u_m] \in K[X]^{1 \times m}, \]
\[ \text{rdeg}(uA) = \max_{1 \leq i \leq m}(\deg(u_i) + \text{rdeg}(A_{i,*})) \]

\[ A \text{ is } s\text{-reduced if and only if for any } u = [u_1 \cdots u_m] \in K[X]^{1 \times m}, \]
\[ \text{rdeg}_s(uA) = \max_{1 \leq i \leq m}(\deg(u_i) + \text{rdeg}_s(A_{i,*})) \]

this means \( \text{rdeg}_s(uA) = \text{rdeg}_t(u) \), where \( t = \text{rdeg}_s(A) \)

- \( s \)-reduced forms provide vectors of minimal \( s \)-degree in the module
- satisfying degree constraints \( (d_1, \ldots, d_m) \Rightarrow \) taking \( s = (-d_1, \ldots, -d_m) \)
- indeed \( \text{cdeg}([p_1 \cdots p_m]) < (d_1, \ldots, d_m) \)
  if and only if \( \text{rdeg}(-d_1, \ldots, -d_m)([p_1 \cdots p_m]) < 0 \)
shifted reduced forms

stability under multiplication

algorithms based on polynomial matrix multiplication


▷ compute a first basis $P_1$ for a subproblem
▷ update the input instance to get the second subproblem
▷ compute a second basis $P_2$ for this second subproblem
▷ the output basis of solutions is $P_2P_1$

we want $P_2P_1$ to be reduced:
1. is it implied by “$P_1$ reduced and $P_2$ reduced”?
2. any idea of how to fix this?
algorithms based on polynomial matrix multiplication

▶ compute a first basis $P_1$ for a subproblem
▶ update the input instance to get the second subproblem
▶ compute a second basis $P_2$ for this second subproblem
▶ the output basis of solutions is $P_2P_1$

we want $P_2P_1$ to be reduced:
1. is it implied by “$P_1$ reduced and $P_2$ reduced”? 
2. any idea of how to fix this?

we want $P_2P_1$ to be reduced
**Theorem:** implied by “$P_1$ is reduced and $P_2$ is $t$-reduced”
where $t = rdeg(P_1)$
shifted reduced forms

stability under multiplication

algorithms based on polynomial matrix multiplication

▶ compute a first basis $P_1$ for a subproblem
▶ update the input instance to get the second subproblem
▶ compute a second basis $P_2$ for this second subproblem
▶ the output basis of solutions is $P_2P_1$

we want $P_2P_1$ to be reduced:
1. is it implied by “$P_1$ reduced and $P_2$ reduced”? 
2. any idea of how to fix this?

we want $P_2P_1$ to be $s$-reduced

**Theorem:** implied by “$P_1$ is $s$-reduced and $P_2$ is $t$-reduced”
where $t = \text{rdeg}_s(P_1)$
shifted reduced forms

stability under multiplication

Let $\mathcal{M} \subseteq \mathcal{M}_1$ be two $K[X]$-submodules of $K[X]^m$ of rank $m$, let $P_1 \in K[X]^{m \times m}$ be a basis of $\mathcal{M}_1$, let $s \in \mathbb{Z}^m$ and $t = \text{rdeg}_s(P_1)$,

- the rank of the module $\mathcal{M}_2 = \{ \lambda \in K[X]^{1 \times m} | \lambda P_1 \in \mathcal{M} \}$ is $m$
- and for any basis $P_2 \in K[X]^{m \times m}$ of $\mathcal{M}_2$,
- the product $P_2 P_1$ is a basis of $\mathcal{M}$
- if $P_1$ is $s$-reduced and $P_2$ is $t$-reduced,
then $P_2 P_1$ is $s$-reduced
shifted reduced forms

stability under multiplication

Let $\mathcal{M} \subseteq \mathcal{M}_1$ be two $\mathbb{K}[X]$-submodules of $\mathbb{K}[X]^m$ of rank $m$, let $P_1 \in \mathbb{K}[X]^{m \times m}$ be a basis of $\mathcal{M}_1$, let $s \in \mathbb{Z}^m$ and $t = \text{rdeg}_s(P_1)$,

- the rank of the module $\mathcal{M}_2 = \{\lambda \in \mathbb{K}[X]^{1 \times m} \mid \lambda P_1 \in \mathcal{M}\}$ is $m$ and for any basis $P_2 \in \mathbb{K}[X]^{m \times m}$ of $\mathcal{M}_2$, the product $P_2 P_1$ is a basis of $\mathcal{M}$
- if $P_1$ is $s$-reduced and $P_2$ is $t$-reduced, then $P_2 P_1$ is $s$-reduced

Let $A \in \mathbb{K}[X]^{m \times m}$ denote the adjugate of $P_1$. Then, we have $AP_1 = \det(P_1)I_m$. Thus, $pAP_1 = \det(P_1)p \in \mathcal{M}$ for all $p \in \mathcal{M}$, and therefore $\mathcal{M}A \subseteq \mathcal{M}_2$. Now, the nonsingularity of $A$ ensures that $\mathcal{M}A$ has rank $m$; this implies that $\mathcal{M}_2$ has rank $m$ as well (see e.g. [Dummit-Foote 2004, Sec. 12.1, Thm. 4]). The matrix $P_2 P_1$ is nonsingular since $\det(P_2 P_1) \neq 0$. Now let $p \in \mathcal{M}$; we want to prove that $p$ is a $\mathbb{K}[X]$-linear combination of the rows of $P_2 P_1$. First, $p \in \mathcal{M}_1$, so there exists $\lambda \in \mathbb{K}[X]^{1 \times m}$ such that $p = \lambda P_1$. But then $\lambda \in \mathcal{M}_2$, and thus there exists $\mu \in \mathbb{K}[X]^{1 \times m}$ such that $\lambda = \mu P_2$. This yields the combination $p = \mu P_2 P_1$. 
Let $M \subseteq M_1$ be two $K[X]$-submodules of $K[X]^m$ of rank $m$, let $P_1 \in K[X]^{m \times m}$ be a basis of $M_1$, let $s \in \mathbb{Z}^m$ and $t = \text{rdeg}_s(P_1)$,

- the rank of the module $M_2 = \{ \lambda \in K[X]^{1 \times m} \mid \lambda P_1 \in M \}$ is $m$ and for any basis $P_2 \in K[X]^{m \times m}$ of $M_2$,
  - the product $P_2 P_1$ is a basis of $M$, if $P_1$ is $s$-reduced and $P_2$ is $t$-reduced,
  - then $P_2 P_1$ is $s$-reduced.

Let $d = \text{rdeg}_t(P_2)$; we have $d = \text{rdeg}_s(P_2 P_1)$ by the predictable degree property. Using $X^{-d} P_2 P_1 X^s = X^{-d} P_2 X^t X^{-t} P_1 X^s$, we obtain that $\text{lm}_s(P_2 P_1) = \text{lm}_t(P_2) \text{lm}_s(P_1)$. By assumption, $\text{lm}_t(P_2)$ and $\text{lm}_s(P_1)$ are invertible, and therefore $\text{lm}_s(P_2 P_1)$ is invertible as well; thus $P_2 P_1$ is $s$-reduced.
<table>
<thead>
<tr>
<th>outline</th>
<th>introduction</th>
<th>shifted reduced forms</th>
<th>fast algorithms</th>
<th>applications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▶ rational approximation and interpolation</td>
<td>▶ reducedness: examples and properties</td>
<td>▶ iterative algorithm and output size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ the vector case</td>
<td>▶ shifted forms and degree constraints</td>
<td>▶ base case: modulus of degree 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ pol. matrices: reminders and motivation</td>
<td>▶ stability under multiplication</td>
<td>▶ recursion: residual and basis multiplication</td>
<td></td>
</tr>
</tbody>
</table>
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

input: vector $F = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$, points $\alpha_1, \ldots, \alpha_d \in \mathbb{K}$, shift $s = (s_1, \ldots, s_m) \in \mathbb{Z}^m$

1. $P = \begin{bmatrix} -p_1 \\ \vdots \\ -p_m \end{bmatrix}$ = identity matrix in $\mathbb{K}[X]^{m \times m}$

2. for $i$ from 1 to $d$:
   a. evaluate updated vector $\begin{bmatrix} (p_1 \cdot F)(\alpha_i) \\ \vdots \\ (p_m \cdot F)(\alpha_i) \end{bmatrix} = (P \cdot F)(\alpha_i)$
   b. choose pivot $\pi$ with smallest $s_\pi$ such that $(p_\pi \cdot F)(\alpha_i) \neq 0$
      update pivot shift $s_\pi = s_\pi + 1$
   c. eliminate:
      /* after this, $\forall j \neq \pi$, $(p_j \cdot F)(\alpha_i) = 0$ */
      for $j \neq \pi$ do $p_j \leftarrow p_j - \frac{(p_j \cdot F)(\alpha_i)}{(p_\pi \cdot F)(\alpha_i)} p_\pi$; $p_\pi \leftarrow (X - \alpha_i)p_\pi$

after $i$ iterations: $P$ is an $s$-reduced basis of solutions for $(\alpha_1, \ldots, \alpha_i)$
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( F = [1 \ L \ L^2 \ L^3]^T \)

iteration: \( i = 1 \) \quad point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift \( [0 \ 2 \ 4 \ 6] \)

basis

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
80 & 73 & 73 & 35 & 66 & 46 & 91 & 64 & 36 \\
95 & 91 & 91 & 61 & 88 & 79 & 36 & 22 & 50 \\
34 & 47 & 47 & 1 & 85 & 45 & 75 & 50 & \end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \)  \( m = 4 \)  \( s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( F = [1 \ L \ L^2 \ L^3]^T \)

iteration: \( i = 1 \)  point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
80 & 73 & 73 & 35 & 66 & 46 & 91 & 64 & \\
95 & 91 & 91 & 61 & 88 & 79 & 36 & 22 & \\
34 & 47 & 47 & 1 & 85 & 45 & 75 & 50 & \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \quad \text{base field } \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \(\mathbf{F} = [1 \quad 1 \quad L^2 \quad L^3]^T\)

iteration: \(i = 1\) \quad point: \(24, 31, 15, 32, 83, 27, 20, 59\)

shift \([0 \quad 2 \quad 4 \quad 6]\)

basis

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
17 & 1 & 0 & 0 \\
2 & 0 & 1 & 0 \\
63 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 90 & 90 & 52 & 83 & 63 & 11 & 81 & 0 \\
0 & 93 & 93 & 63 & 90 & 81 & 38 & 24 & 0 \\
0 & 13 & 13 & 64 & 51 & 11 & 41 & 16 & 0
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: $d = 8$, $m = 4$, $s = (0, 2, 4, 6)$, base field $\mathbb{F}_{97}$

input: $(24, 31, 15, 32, 83, 27, 20, 59)$ and $\mathbf{F} = [1 \ L \ L^2 \ L^3]^T$

iteration: $i = 1$  
point: $24, 31, 15, 32, 83, 27, 20, 59$

shift

$$\begin{bmatrix}
X + 73 \\
17 \\
2 \\
63
\end{bmatrix}$$

basis

$$\begin{bmatrix}
0 & 2 & 4 & 6 \\
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}$$

values

$$\begin{bmatrix}
0 & 7 & 88 & 8 & 59 & 3 & 93 & 35 \\
0 & 90 & 90 & 52 & 83 & 63 & 11 & 81 \\
0 & 93 & 93 & 63 & 90 & 81 & 38 & 24 \\
0 & 13 & 13 & 64 & 51 & 11 & 41 & 16
\end{bmatrix}$$
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: $d = 8$  $m = 4$  $s = (0, 2, 4, 6)$, base field $\mathbb{F}_{97}$

input: $(24, 31, 15, 32, 83, 27, 20, 59)$ and $F = [1 \ L \ L^2 \ L^3]^{T}$

iteration: $i = 2$  point: $24, 31, 15, 32, 83, 27, 20, 59$

shift

$$\begin{bmatrix} X + 73 & 0 & 0 & 0 \\ 17 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 63 & 0 & 0 & 1 \end{bmatrix}$$

basis

$$\begin{bmatrix} 0 & 7 & 88 & 8 & 59 & 3 & 93 & 35 \\ 0 & 90 & 90 & 52 & 83 & 63 & 11 & 81 \\ 0 & 93 & 93 & 63 & 90 & 81 & 38 & 24 \\ 0 & 13 & 13 & 64 & 51 & 11 & 41 & 16 \end{bmatrix}$$
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( F = [1 \quad L \quad L^2 \quad L^3]^T \)

iteration: \( i = 2 \)  
point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift: \( [1 \quad 2 \quad 4 \quad 6] \)

basis:
\[
\begin{bmatrix}
X + 73 & 0 & 0 & 0 \\
X + 90 & 1 & 0 & 0 \\
56X + 16 & 0 & 1 & 0 \\
12X + 66 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values:
\[
\begin{bmatrix}
0 & 7 & 88 & 8 & 59 & 3 & 93 & 35 \\
0 & 0 & 81 & 60 & 45 & 66 & 7 & 19 \\
0 & 0 & 74 & 26 & 96 & 55 & 8 & 44 \\
0 & 0 & 2 & 63 & 80 & 47 & 90 & 48 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \( (24, 31, 15, 32, 83, 27, 20, 59) \) and \( F = [1 \ L \ L^2 \ L^3]^T \)

iteration: \( i = 2 \) \quad point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift \( [2 \ 2 \ 4 \ 6] \)

basis
\[
\begin{bmatrix}
X^2 + 42X + 65 & 0 & 0 & 0 \\
X + 90 & 1 & 0 & 0 \\
56X + 16 & 0 & 1 & 0 \\
12X + 66 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values
\[
\begin{bmatrix}
0 & 0 & 47 & 8 & 61 & 85 & 44 & 10 \\
0 & 0 & 81 & 60 & 45 & 66 & 7 & 19 \\
0 & 0 & 74 & 26 & 96 & 55 & 8 & 44 \\
0 & 0 & 2 & 63 & 80 & 47 & 90 & 48 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( \mathbf{F} = [1 \ L \ L^2 \ L^3]^T \)

iteration: \( i = 3 \)  \quad point: 24, 31, 15, 32, 83, 27, 20, 59

shift

\[
\begin{bmatrix}
X^2 + 42X + 65 & 0 & 0 & 0 \\
X + 90 & 1 & 0 & 0 \\
56X + 16 & 0 & 1 & 0 \\
12X + 66 & 0 & 0 & 1 \\
\end{bmatrix}
\]

basis

\[
\begin{bmatrix}
0 & 0 & 47 & 8 & 61 & 85 & 44 & 10 \\
0 & 0 & 81 & 60 & 45 & 66 & 7 & 19 \\
0 & 0 & 74 & 26 & 96 & 55 & 8 & 44 \\
0 & 0 & 2 & 63 & 80 & 47 & 90 & 48 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \(d = 8\), \(m = 4\), \(s = (0, 2, 4, 6)\), base field \(\mathbb{F}_{97}\)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \(\mathbf{F} = [1 \quad \mathbf{L} \quad \mathbf{L}^2 \quad \mathbf{L}^3]^T\)

iteration: \(i = 3\)  
point: 24, 31, 15, 32, 83, 27, 20, 59

shift \([3 \quad 2 \quad 4 \quad 6]\)

\[
\begin{align*}
\begin{bmatrix}
X^3 + 27X^2 + 17X + 92 & 0 & 0 & 0 \\
54X^2 + 38X + 11 & 1 & 0 & 0 \\
17X^2 + 91X + 54 & 0 & 1 & 0 \\
66X^2 + 68X + 88 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

basis

\[
\begin{bmatrix}
0 & 0 & 0 & 39 & 74 & 50 & 26 & 52 \\
0 & 0 & 0 & 7 & 41 & 0 & 55 & 74 \\
0 & 0 & 0 & 65 & 66 & 45 & 77 & 20 \\
0 & 0 & 0 & 9 & 32 & 31 & 84 & 29
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: $d = 8$, $m = 4$, $s = (0, 2, 4, 6)$, base field $\mathbb{F}_{97}$

input: $(24, 31, 15, 32, 83, 27, 20, 59)$ and $F = [1 \ L \ L^2 \ L^3]^T$

iteration: $i = 4$  
point: $24, 31, 15, 32, 83, 27, 20, 59$

shift

$[3 \ 2 \ 4 \ 6]$  

basis

\[
\begin{bmatrix}
X^3 + 27X^2 + 17X + 92 & 0 & 0 & 0 \\
54X^2 + 38X + 11 & 1 & 0 & 0 \\
17X^2 + 91X + 54 & 0 & 1 & 0 \\
66X^2 + 68X + 88 & 0 & 0 & 1 \\
\end{bmatrix}
\]

values

\[
\begin{bmatrix}
0 & 0 & 0 & 39 & 74 & 50 & 26 & 52 \\
0 & 0 & 0 & 7 & 41 & 0 & 55 & 74 \\
0 & 0 & 0 & 65 & 66 & 45 & 77 & 20 \\
0 & 0 & 0 & 9 & 32 & 31 & 84 & 29 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \), \( m = 4 \), \( s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( F = [1 \quad L \quad L^2 \quad L^3]^T \)

iteration: \( i = 4 \) \quad point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

\[
\begin{align*}
\text{shift} & \quad \begin{bmatrix} 3 & 3 & 4 & 6 \end{bmatrix} \\
\text{basis} & \begin{bmatrix}
X^3 + 31X^2 + 27X + 3 & 36 & 0 & 0 \\
54X^3 + 56X^2 + 56X + 36 & X + 65 & 0 & 0 \\
56X^2 + 43X + 35 & 60 & 1 & 0 \\
52X^2 + 33X + 60 & 68 & 0 & 1 \\
\end{bmatrix} \\
\text{values} & \begin{bmatrix}
0 & 0 & 0 & 0 & 95 & 50 & 66 & 0 \\
0 & 0 & 0 & 0 & 54 & 0 & 19 & 58 \\
0 & 0 & 0 & 0 & 4 & 45 & 79 & 95 \\
0 & 0 & 0 & 0 & 7 & 31 & 41 & 17 \\
\end{bmatrix}
\end{align*}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \(\mathbf{F} = [1 \ L \ L^2 \ L^3]^T\)

iteration: \(i = 5\)  \quad point: 24, 31, 15, 32, 83, 27, 20, 59

\[
\begin{bmatrix}
X^4 + 45X^3 + 73X^2 + 90X + 42 & 36X + 19 & 0 & 0 \\
81X^3 + 20X^2 + 9X + 20 & X + 67 & 0 & 0 \\
2X^3 + 21X^2 + 41 & 35 & 1 & 0 \\
52X^3 + 15X^2 + 79X + 22 & 0 & 0 & 1
\end{bmatrix}
\]

shift \([4 \ 3 \ 4 \ 6]\)

basis

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 13 & 13 & 0 \\
0 & 0 & 0 & 0 & 0 & 89 & 55 & 58 \\
0 & 0 & 0 & 0 & 0 & 48 & 17 & 95 \\
0 & 0 & 0 & 0 & 0 & 12 & 78 & 17
\end{bmatrix}
\]

values
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( \mathbf{F} = \begin{bmatrix} 1 & L & L^2 & L^3 \end{bmatrix}^T \)

iteration: \( i = 6 \)  
point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift \( [4 \quad 4 \quad 4 \quad 6] \)

\[
\begin{bmatrix}
X^4 + 19X^3 + 57X^2 + 44X + 26 & \quad \quad 74X + 43 & \quad \quad 0 & \quad \quad 0 \\
81X^4 + 64X^3 + 51X^2 + 68X + 42 & \quad \quad X^2 + 40X + 34 & \quad \quad 0 & \quad \quad 0 \\
3X^3 + 44X^2 + 54X + 64 & \quad \quad 6X + 49 & \quad \quad 1 & \quad \quad 0 \\
28X^3 + 45X^2 + 44X + 52 & \quad \quad 50X + 52 & \quad \quad 0 & \quad \quad 1 \\
\end{bmatrix}
\]

basis

values

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 66 & 70 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 13 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 56 & 55 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 7 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) \), base field \( \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( \mathbf{F} = \begin{bmatrix} 1 & L & L^2 & L^3 \end{bmatrix}^T \)

iteration: \( i = 7 \)  
point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift \[ \begin{bmatrix} 5 & 4 & 4 & 6 \end{bmatrix} \]

basis
\[
\begin{bmatrix}
X^5 + 96X^4 + 65X^3 + 68X^2 + 19X + 62 & 74X^2 + 18X + 13 & 0 & 0 \\
6X^4 + 94X^3 + 44X^2 + 66X + 32 & X^2 + 19X + 10 & 0 & 0 \\
55X^4 + 78X^3 + 75X^2 + 49X + 39 & 2X + 86 & 1 & 0 \\
13X^4 + 81X^3 + 10X^2 + 34X + 2 & 42X + 29 & 0 & 1 \\
\end{bmatrix}
\]

values
\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 25 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 44 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm [van Barel-Bultheel / Beckermann-Labahn]

parameters: \( d = 8 \quad m = 4 \quad s = (0, 2, 4, 6) , \quad \text{base field } \mathbb{F}_{97} \)

input: \((24, 31, 15, 32, 83, 27, 20, 59)\) and \( \mathbf{F} = [1 \quad L \quad L^2 \quad L^3]^T \)

iteration: \( i = 8 \) \quad point: \( 24, 31, 15, 32, 83, 27, 20, 59 \)

shift

\[
\begin{bmatrix}
5 & 5 & 4 & 6
\end{bmatrix}
\]

basis

\[
\begin{bmatrix}
x^5 + 12x^4 + 10x^3 + 34x^2 + 65x + 2 & 60x^2 + 43x + 67 & 0 & 0 \\
6x^5 + 31x^4 + 27x^3 + 89x^2 + 18x + 52 & x^3 + 57x^2 + 53x + 89 & 0 & 0 \\
2x^4 + 56x^3 + 42x^2 + 48x + 15 & 72x^2 + 12x + 30 & 1 & 0 \\
40x^4 + 19x^3 + 14x^2 + 40x + 49 & 53x^2 + 79x + 74 & 0 & 1
\end{bmatrix}
\]

values

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
fast algorithms

base case: modulus of degree 1

modular vector equation

input:
- vector $\mathbf{F} = [f_1 \cdots f_m]^T \in K[X]^{m \times 1}$ of degree $< d$
- field elements $(\alpha_1, \ldots, \alpha_d) \in K^d$
- shift $s = (s_1, \ldots, s_m) \in \mathbb{Z}^m$

output:
matrix $\mathbf{P} \in K[X]^{m \times m}$ such that
- $\mathbf{PF} = 0 \mod \prod_{1 \leq i \leq d}(X - \alpha_i)$
- $\mathbf{P}$ generates all vectors $\mathbf{p}$ such that $\mathbf{pF} = 0 \mod \prod_{1 \leq i \leq d}(X - \alpha_i)$
- $\mathbf{P}$ is $s$-reduced

notation: $I(\alpha, \mathbf{F}) = \{ \mathbf{p} \in K[X]^{1 \times m} | \mathbf{pF} = 0 \mod \prod_{1 \leq i \leq d}(X - \alpha_i) \}$
fast algorithms

base case: modulus of degree 1

modular vector reconstruction: base case

input:
- vector $\mathbf{F} = [f_1 \cdots f_m]^T \in K[X]^{m \times 1}$ of degree $< 1$
- field element $\alpha \in K$
- shift $\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{Z}^m$

output:
matrix $\mathbf{P} \in K[X]^{m \times m}$ such that
- $\mathbf{PF} = 0 \mod (X - \alpha)$
- $\mathbf{P}$ generates all vectors $\mathbf{p}$ such that $\mathbf{pF} = 0 \mod (X - \alpha)$
- $\mathbf{P}$ is $s$-reduced
modular vector reconstruction: base case

input:
- vector \( \mathbf{F} = [f_1 \cdots f_m]^\top \in \mathbb{K}[X]^{m \times 1} \) of degree < 1
- field element \( \alpha \in \mathbb{K} \)
- shift \( \mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{Z}^m \)

output:
- matrix \( \mathbf{P} \in \mathbb{K}[X]^{m \times m} \) such that
  - \( \mathbf{PF} = 0 \) mod \( (X - \alpha) \)
  - \( \mathbf{P} \) generates all vectors \( \mathbf{p} \) such that \( \mathbf{pF} = 0 \) mod \( (X - \alpha) \)
  - \( \mathbf{P} \) is \( s \)-reduced

\( \mathbf{F} \in \mathbb{K}^{m \times 1} \)
modular vector reconstruction: base case

iterative algorithm:  \[ P = \begin{bmatrix} I_{\pi - 1} & \lambda_1 & 0 \\ 0 & X - \alpha & 0 \\ 0 & \lambda_2 & I_{m - \pi} \end{bmatrix} \]

where
- \( \pi \) minimizes \( s_\pi \) among indices such that \( (p_\pi F)(\alpha_i) \neq 0 \)
- the vectors \( \lambda_1 \in \mathbb{K}^{(\pi - 1) \times 1} \) and \( \lambda_2 \in \mathbb{K}^{(m - \pi) \times 1} \) are constant
fast algorithms

base case: modulus of degree 1

modular vector reconstruction: base case

iterative algorithm: \( P = \begin{bmatrix} I_{\pi-1} & \lambda_1 & 0 \\ 0 & X - \alpha & 0 \\ 0 & \lambda_2 & I_{m-\pi} \end{bmatrix} \)

where

- \( \pi \) minimizes \( s_\pi \) among indices such that \( (p_{\pi}F)(\alpha_i) \neq 0 \)
- the vectors \( \lambda_1 \in \mathbb{K}^{(\pi-1) \times 1} \) and \( \lambda_2 \in \mathbb{K}^{(m-\pi) \times 1} \) are constant

iterative algorithm:

- \( P = \) identity matrix in \( \mathbb{K}[X]^{m \times m} \)
- for \( i \) from 1 to \( d \):
  a. from the evaluation \( F(\alpha_i) \), find \( P_i \) as above
  b. update shift \( s_\pi \leftarrow s_\pi + 1 \)
  c. update \( P \leftarrow P_iP \) as well as \( F \leftarrow P_iF \mod \prod_{i+1 \leq j \leq d} (X - \alpha_j) \)

called residual vector
fast algorithms

base case: modulus of degree 1

modular vector reconstruction: base case

iterative algorithm:

\[
P = \begin{bmatrix}
I_{\pi-1} & \lambda_1 & 0 \\
0 & X - \alpha & 0 \\
0 & \lambda_2 & I_{m-\pi}
\end{bmatrix}
\]

where

- \(\pi\) minimizes \(s_{\pi}\) among indices such that \((p_\pi F)(\alpha_i) \neq 0\)
- the vectors \(\lambda_1 \in K^{(\pi-1) \times 1}\) and \(\lambda_2 \in K^{(m-\pi) \times 1}\) are constant

**complexity** \(O(m^2d^2)\):

- iteration with \(d\) steps
- each step: evaluation of \(F\) + multiplications \(P_i F\) and \(P_i P\)
- at any stage \(F\) has degree \(<d\) and size \(m \times 1\)
- at any stage \(P\) has degree \(\leq d\) and size \(m \times m\)

normalizing at each step + refined analysis yields \(O(md^2)\)
modular vector reconstruction: base case

iterative algorithm: 

\[
P = \begin{bmatrix} 
I_{\pi-1} & \lambda_1 & 0 \\
0 & X - \alpha & 0 \\
0 & \lambda_2 & I_{m-\pi} 
\end{bmatrix}
\]

where

- \( \pi \) minimizes \( s_\pi \) among indices such that \((p_\pi F)(\alpha_i) \neq 0\)
- the vectors \( \lambda_1 \in \mathbb{K}^{(\pi-1) \times 1} \) and \( \lambda_2 \in \mathbb{K}^{(m-\pi) \times 1} \) are constant

**correctness:**

- the main task is to prove the base case with \( P_i \)
- then, direct consequence of the “basis multiplication theorem”
fast algorithms

iterative algorithm – complexity aspects

▶ input size: \( md + d \) elements from \( K \)
  . \( md \) coefficients of \( F \), assumed reduced modulo \( M(X) \)
  . \( d \) points \( \alpha_1, \ldots, \alpha_d \)

▶ output size: \( \leq m^2(d + 1) \) elements from \( K \)
  . \( m \times m \) matrix \( P \) of degree at most \( i \) at step \( i \)

is this output size bound tight?
fast algorithms

iterative algorithm – complexity aspects

- **input size:** \( md + d \) elements from \( K \)
  . \( md \) coefficients of \( F \), assumed reduced modulo \( M(X) \)
  . \( d \) points \( \alpha_1, \ldots, \alpha_d \)

- **output size:** \( \leq m^2(d + 1) \) elements from \( K \)
  . \( m \times m \) matrix \( P \) of degree at most \( i \) at step \( i \)

is this output size bound tight?

- one can prove \( \deg(\det(P)) \leq d \)
  . \( P \) is a basis of \( J(\alpha, F) \), which is the kernel of \( K[X]^m \rightarrow K[X]/\langle M(X) \rangle, p \mapsto pF \)
  . \( K[X]^m/J(\alpha, F) \) has \( K \)-dimension at most \( \dim_K(K[X]/\langle M(X) \rangle) = d \)

- **normalized bases** have average column degree \( \leq d \), and size \( \leq m(d + 1) \)

- yet the bound \( \Theta(m^2(d + 1)) \) is tight for this algorithm
  . normalizing at each step is feasible for the iterative version
  . but is much harder to incorporate in fast divide and conquer versions
parameters: $K = F_{97}$, $m = 4$, $\alpha = 0$, $d = 128$, $s = (0, \ldots, 0)$

choose random polynomial $R(X)$ of degree $< 128$

$$F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} R \\ R + XR \\ XR + X^2R \\ X^2R + X^3R \end{bmatrix}$$

- approximants are $p$ such that $pF = 0 \mod X^{128}$
- $F$ has small vectors in its left kernel
  $\Rightarrow$ reduced approximant basis has unbalanced row degrees $(1, 1, 1, 125)$
- will help to build an example with output size $\Omega(m^2d)$
### running the iterative algorithm:

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>( (0, 0, 0, 0) )</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>( R )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>( R + XR )</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>( XR + X^2R )</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>( X^2R + X^3R )</td>
</tr>
</tbody>
</table>

\[ P \]
Running the iterative algorithm:

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>(0, 0, 0, 0)</td>
<td>(1, 0, 0, 0)</td>
</tr>
<tr>
<td>f₁</td>
<td>R</td>
<td>XR</td>
</tr>
<tr>
<td>f₂</td>
<td>R + XR</td>
<td>XR</td>
</tr>
<tr>
<td>f₃</td>
<td>XR + X²R</td>
<td>XR + X²R</td>
</tr>
<tr>
<td>f₄</td>
<td>X²R + X³R</td>
<td>X²R + X³R</td>
</tr>
</tbody>
</table>

P = \[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
fast algorithms

iterative algorithm – complexity aspects

running the iterative algorithm:

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>(0, 0, 0, 0)</td>
<td>(1, 0, 0, 0)</td>
<td>(1, 1, 0, 0)</td>
</tr>
<tr>
<td>f₁</td>
<td>R</td>
<td>XR</td>
<td>0</td>
</tr>
<tr>
<td>f₂</td>
<td>R + XR</td>
<td>XR</td>
<td>X²R</td>
</tr>
<tr>
<td>f₃</td>
<td>XR + X²R</td>
<td>XR + X²R</td>
<td>X²R</td>
</tr>
<tr>
<td>f₄</td>
<td>X²R + X³R</td>
<td>X²R + X³R</td>
<td>X²R + X³R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

degrees and "pivots" in final basis

P:

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 0 & 0
\end{bmatrix}
\]
### fast algorithms

#### iterative algorithm – complexity aspects

**running the iterative algorithm:**

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>((0, 0, 0, 0))</td>
<td>((1, 0, 0, 0))</td>
<td>((1, 1, 0, 0))</td>
<td>((1, 1, 1, 0))</td>
</tr>
<tr>
<td>f_1</td>
<td>R</td>
<td>XR</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_2</td>
<td>R + XR</td>
<td>XR + X^2R</td>
<td>X^2R</td>
<td>X^3R</td>
</tr>
<tr>
<td>f_3</td>
<td>XR + X^2R</td>
<td>XR + X^2R</td>
<td>X^2R</td>
<td>X^3R</td>
</tr>
<tr>
<td>f_4</td>
<td>X^2R + X^3R</td>
<td>X^2R + X^3R</td>
<td>X^2R + X^3R</td>
<td>X^3R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>1 0 0 (0)</th>
<th>1 0 0 (0)</th>
<th>1 0 0 (1)</th>
<th>0 0 0 0 (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 0 (0)</td>
<td>0 0 0 (0)</td>
<td>0 0 0 0 (0)</td>
<td>0 0 0 0 0 (0)</td>
</tr>
</tbody>
</table>
### Fast Algorithms

#### Iterative Algorithm – Complexity Aspects

---

**Running the Iterative Algorithm:**

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>((0, 0, 0, 0))</td>
<td>((1, 0, 0, 0))</td>
<td>((1, 1, 0, 0))</td>
<td>((1, 1, 1, 0))</td>
<td>...</td>
</tr>
<tr>
<td>f_1</td>
<td>R</td>
<td>XR</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_2</td>
<td>R + XR</td>
<td>XR</td>
<td>(X^2R)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f_3</td>
<td>XR + (X^2R)</td>
<td>XR + (X^2R)</td>
<td>(X^2R)</td>
<td>(X^3R)</td>
<td>0</td>
</tr>
<tr>
<td>f_4</td>
<td>(X^2R + X^3R)</td>
<td>(X^2R + X^3R)</td>
<td>(X^2R + X^3R)</td>
<td>(X^3R)</td>
<td>(X^4R)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
1 & 0 \\
0 & 0
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 1 & 0
\end{bmatrix}
\] |
| ... |

**Degrees and "Pivots" in Final Basis:**

<table>
<thead>
<tr>
<th>P</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 1 & 0
\end{bmatrix}
\] |
| \[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 1 & 1
\end{bmatrix}
\] |
| ... |
### Fast Algorithms

#### Iterative Algorithm – Complexity Aspects

**Running the Iterative Algorithm:**

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>$$(0, 0, 0, 0)$$</td>
<td>$$(1, 0, 0, 0)$$</td>
<td>$$(1, 1, 0, 0)$$</td>
<td>$$(1, 1, 1, 0)$$</td>
<td>…</td>
</tr>
<tr>
<td>$f_1$</td>
<td>$R$</td>
<td>$XR$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$R + XR$</td>
<td>$XR$</td>
<td>$X^2R$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$XR + X^2R$</td>
<td>$XR + X^2R$</td>
<td>$X^2R$</td>
<td>$X^3R$</td>
<td>0</td>
</tr>
<tr>
<td>$f_4$</td>
<td>$X^2R + X^3R$</td>
<td>$X^2R + X^3R$</td>
<td>$X^2R + X^3R$</td>
<td>$X^3R$</td>
<td>$X^4R$</td>
</tr>
<tr>
<td>$\mathbf{P}$</td>
<td>$\begin{bmatrix} 1 &amp; 0 \ 0 &amp; 1 \ 0 &amp; 0 \ 0 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 1 \ 1 &amp; 1 \ 0 &amp; 1 \ 0 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 0 \ 1 &amp; 1 \ 0 &amp; 1 \ 0 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 0 \ 1 &amp; 1 \ 1 &amp; 1 \ 0 &amp; 1 \end{bmatrix}$</td>
<td>…</td>
</tr>
</tbody>
</table>

Degrees and “pivots” in final basis $\mathbf{P}$:

$$\begin{bmatrix} 1 & 0 & 125 \\ 1 & 1 & 125 \\ 1 & 1 & 125 \\ 1 & 1 & 125 \end{bmatrix}$$
fast algorithms

iterative algorithm – complexity aspects

parameters: $m = 8$, $d = 128$, $s = (0, 0, 0, 0, d, d, d, d)$

input $F$: same $f_1, f_2, f_3, f_4$  /  random $f_5, f_6, f_7, f_8$

$i = 4$

$$
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
$$
fast algorithms

iterative algorithm – complexity aspects

parameters: \( m = 8, \ d = 128, \ s = (0, 0, 0, 0, d, d, d, d) \)

input \( F \): same \( f_1, f_2, f_3, f_4 \) / random \( f_5, f_6, f_7, f_8 \)

\[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
125 & 125 & 125 & 125 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0
\end{bmatrix}
\]

▶ 1/4 of the entries have degree \( \approx d \): size \( \Theta(m^2 d) \)

▶ remark: complexity of iterative algorithm is \( O(m^2 d^2) \)

→ improved to \( O(md^2) \) via normalization

opinions on a “reasonable” target cost for fast algorithms?
fast algorithms

iterative algorithm – complexity aspects

parameters: \( m = 8, \ d = 128, \ s = (0, 0, 0, 0, d, d, d, d) \)

input \( F \): same \( f_1, f_2, f_3, f_4 \) / random \( f_5, f_6, f_7, f_8 \)

\[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
125 & 125 & 125 & 125 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
124 & 124 & 124 & 124 & 0 \\
\end{bmatrix}
\]

▶ 1/4 of the entries have degree \( \approx d \): size \( \Theta(m^2d) \)

▶ remark: complexity of iterative algorithm is \( O(m^2d^2) \)
  → improved to \( O(md^2) \) via normalization

▶ opinions on a “reasonable” target cost for fast algorithms?
divide and conquer algorithm:

input: $F, (\alpha_1, \ldots, \alpha_d), s$  |  output: $P$

- if $d = 1$, use the base case algorithm to find $P$ and return
- otherwise:
  a. $M_1 \leftarrow (X - \alpha_1) \cdots (X - \alpha_{\lfloor d/2 \rfloor})$; $M_2 \leftarrow (X - \alpha_{\lceil d/2 \rceil}) \cdots (X - \alpha_d)$
  b. $P_1 \leftarrow$ call the algorithm on $F \text{ rem } M_1, (\alpha_1, \ldots, \alpha_{\lfloor d/2 \rfloor}), s$
  c. updated shift: $t \leftarrow \text{rdeg}_s(P_1)$
  d. residual: $G \leftarrow \frac{1}{M_1} P_1 F$
  e. $P_2 \leftarrow$ call the algorithm on $G \text{ rem } M_2, (\alpha_{\lceil d/2 \rceil}, \ldots, \alpha_d), t$
  f. return the product $P_2P_1$
**fast algorithms**

recursion: residual and basis multiplication

**divide and conquer algorithm:**

input: $F, (\alpha_1, \ldots, \alpha_d), s$  
output: $P$

- if $d = 1$, use the base case algorithm to find $P$ and return
- otherwise:
  a. $M_1 \leftarrow (X - \alpha_1) \cdots (X - \alpha_{\lfloor d/2 \rfloor})$; $M_2 \leftarrow (X - \alpha_{\lceil d/2 \rceil}) \cdots (X - \alpha_d)$
  b. $P_1 \leftarrow$ call the algorithm on $F \text{ rem } M_1, (\alpha_1, \ldots, \alpha_{\lfloor d/2 \rfloor}), s$
  c. updated shift: $t \leftarrow \text{rdeg}_s(P_1)$
  d. residual: $G \leftarrow \frac{1}{M_1} P_1 F$
  e. $P_2 \leftarrow$ call the algorithm on $G \text{ rem } M_2, (\alpha_{\lceil d/2 \rceil}, \ldots, \alpha_d), t$
  f. return the product $P_2 P_1$

**correctness:**

- correctness of base case
- then, direct consequence of the “basis multiplication theorem”
- about the residual: $\{p \mid pP_1 F = 0 \text{ mod } M\} = \{p \mid pG = 0 \text{ mod } M_2\}$
divide and conquer algorithm:

input: \( F, (\alpha_1, \ldots, \alpha_d), s \)  
output: \( P \)

- if \( d = 1 \), use the base case algorithm to find \( P \) and return
- otherwise:
  a. \( M_1 \leftarrow (X - \alpha_1) \cdots (X - \alpha_{\lfloor d/2 \rfloor}) \);  
     \( M_2 \leftarrow (X - \alpha_{\lceil d/2 \rceil}) \cdots (X - \alpha_d) \)
  b. \( P_1 \leftarrow \) call the algorithm on \( F \) rem \( M_1, (\alpha_1, \ldots, \alpha_{\lfloor d/2 \rfloor}) \), \( s \)
  c. updated shift: \( t \leftarrow \text{rdeg}_s(P_1) \)
  d. residual: \( G \leftarrow \frac{1}{M_1} P_1 F \)
  e. \( P_2 \leftarrow \) call the algorithm on \( G \) rem \( M_2, (\alpha_{\lceil d/2 \rceil}, \ldots, \alpha_d) \), \( t \)
  f. return the product \( P_2 P_1 \)

**complexity** \( O(m^\omega M(d) \log(d)) \):
- if \( \omega = 2 \), quasi-linear in worst-case output size
- most expensive step in the recursion is the product \( P_2 P_1 \)
- equation \( C(m, d) = C(m, \lfloor d/2 \rfloor) + C(m, \lceil d/2 \rceil) + O(m^\omega M(d)) \)
fast algorithms

recursion: residual and basis multiplication

complexity of each step:

- residual $G \leftarrow \frac{1}{M_1} P_1 F$  \quad $O(m^2 M(d))$
- $F$ rem $M_1$ and $G$ rem $M_2$  \quad $O(m M(d))$
- product $P_2 P_1$  \quad $O(m^\omega M(d))$
- two recursive calls  \quad $2C(m, \lfloor d/2 \rfloor)$

input: $\text{deg}(F) < d$  \quad output: $\text{deg}(P) \leq d$
fast algorithms

recursion: residual and basis multiplication

input: \( \deg(F) < d \)  
output: \( \deg(P) \leq d \)

complexity of each step:

- residual \( G \leftarrow \frac{1}{M_1} P_1 F \)  
  \( O(m^2 M(d)) \)
- \( F \) rem \( M_1 \) and \( G \) rem \( M_2 \)  
  \( O(mM(d)) \)
- product \( P_2 P_1 \)  
  \( O(m^\omega M(d)) \)
- two recursive calls  
  \( 2C(m, \lceil d/2 \rceil) \)

\[
C(m, d) = C(m, \lceil d/2 \rceil) + C(m, \lceil d/2 \rceil) + O(m^\omega M(d))
\]

d base cases, each one costs . . . ???
fast algorithms

recursion: residual and basis multiplication

input: $\deg(F) < d$

output: $\deg(P) \leq d$

complexity of each step:

- residual $G \leftarrow \frac{1}{M_1}P_1F$ \hspace{1cm} $O(m^2M(d))$
- $F \text{ rem } M_1$ and $G \text{ rem } M_2$ \hspace{1cm} $O(mM(d))$
- product $P_2P_1$ \hspace{1cm} $O(m^\omega M(d))$
- two recursive calls \hspace{1cm} $2C(m, \lceil d/2 \rceil)$

\[
C(m, d) = C(m, \lfloor d/2 \rfloor) + C(m, \lceil d/2 \rceil) + O(m^\omega M(d))
\]

d base cases, each one costs $O(m)$

\[
\Rightarrow \quad O(m^\omega M(d) \log(d))
\]

unrolling: $m^\omega (M(d) + 2M(\frac{d}{2}) + 4M(\frac{d}{4}) + \cdots + \frac{d}{2}M(2)) + dm$
### Fast Algorithms

**Recursion: Residual and Basis Multiplication**

<table>
<thead>
<tr>
<th>Complexity of Each Step</th>
<th>Output: (\text{deg}(P) \leq d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Residual: (G \leftarrow \frac{1}{M_1}P_1F)</td>
<td>(O(m^2M(d)))</td>
</tr>
<tr>
<td>▶ (F) rem (M_1) and (G) rem (M_2)</td>
<td>(O(mM(d)))</td>
</tr>
<tr>
<td>▶ Product: (P_2P_1)</td>
<td>(O(m^\omega M(d)))</td>
</tr>
<tr>
<td>▶ Two Recursive Calls</td>
<td>(2\mathbb{C}(m, \lfloor d/2 \rfloor))</td>
</tr>
</tbody>
</table>

**Input:** \(\text{deg}(F') < d\)

**Output:** \(\text{deg}(P) \approx \left\lceil \frac{d}{m} \right\rceil\)

- \(s = 0\) and generic \(F\):
  - \(O(m^\omega M(\left\lceil \frac{d}{m} \right\rceil))\) unchanged

**Unrolling:** \(m^\omega (M(d) + 2M(\frac{d}{2}) + 4M(\frac{d}{4}) + \cdots + \frac{d}{2}M(2)) + dm\)
fast algorithms

recursion: residual and basis multiplication

input: $\deg(F) < d$

output: $\deg(P) \leq d$

output: $\deg(P) \approx \lceil \frac{d}{m} \rceil$

complexity of each step:

- residual $G \leftarrow \frac{1}{M_1} P_1 F$
  
  $O(m^2 M(d))$

- $F \text{ rem } M_1$ and $G \text{ rem } M_2$
  
  $O(mM(d))$

- product $P_2 P_1$
  
  $O(m^\omega M(d))$

- two recursive calls
  
  $2C(m, \lceil d/2 \rceil)$

\[
\begin{align*}
C(m, d) &= C(m, \lfloor d/2 \rfloor) + C(m, \lceil d/2 \rceil) + O(m^\omega M(d)) \\
\text{d base cases, each one costs } O(m) \\
\Rightarrow O(m^\omega M(d) \log(d)) & \quad O(m^\omega M(\lceil \frac{d}{m} \rceil) \log(\lceil \frac{d}{m} \rceil))
\end{align*}
\]

unrolling: $m^\omega M(d) + 2M(\frac{d}{2}) + 4M(\frac{d}{4}) + \cdots + \frac{d}{2} M(2) + d m$

\[s = 0 \text{ and generic } F:
\]

$O(m^\omega M(\lceil \frac{d}{m} \rceil))$
unchanged

$O(m^\omega M(\lceil \frac{d}{m} \rceil))$
unchanged

- partial linearization
- base case for $d \approx m$, costs $O(m^\omega)$
fast algorithms

recursion: residual and basis multiplication

state of the art:

- recursive algorithm: from [Beckermann-Labahn 1994] (for Hermite-Padé)
  it also works for $F \in \mathbb{K}[X]^{m \times n}$ with $n > 1$

- [Giorgi-Jeannerod-Villard 2003] achieved $O(m^\omega M(d) \log(d))$
  for $F \mod X^d$, with $n \geq 1$ and $n \in O(m)$

- for $s = 0$ and generic $F$: $O^\sim(m^\omega \lceil \frac{nd}{m} \rceil)$ is [folklore]
fast algorithms

recursion: residual and basis multiplication

state of the art:

- **recursive** algorithm: from [Beckermann-Labahn 1994] (for Hermite-Padé)
  it also works for $F \in \mathbb{K}[X]^{m \times n}$ with $n > 1$

- [Giorgi-Jeannerod-Villard 2003] achieved $O(m^\omega M(d) \log(d))$
  for $F \mod X^d$, with $n \geq 1$ and $n \in O(m)$

- for $s = 0$ and generic $F$: $O^\sim(m^\omega \lceil \frac{nd}{m} \rceil)$ is [folklore]

- more recently: $O^\sim(m^{\omega-1}nd)$ for $F \mod X^d$
  [Storjohann 2006] [Zhou-Labahn 2012] [Jeannerod-Neiger-Villard 2020]

$\Rightarrow$ any $s$, no genericity assumption, returns the canonical basis “$s$-Popov”
fast algorithms

recursion: residual and basis multiplication

state of the art:

- recursive algorithm: from [Beckermann-Labahn 1994] (for Hermite-Padé)
  it also works for $F \in \mathbb{K}[X]^{m \times n}$ with $n > 1$

- [Giorgi-Jeannerod-Villard 2003] achieved $O(m^\omega M(d) \log(d))$
  for $F \mod X^d$, with $n \geq 1$ and $n \in O(m)$

- for $s = 0$ and generic $F$: $O^\sim(m^\omega \left\lceil \frac{nd}{m} \right\rceil)$ is [folklore]

- more recently: $O^\sim(m^{\omega - 1}nd)$ for $F \mod X^d$
  [Storjohann 2006] [Zhou-Labahn 2012] [Jeannerod-Neiger-Villard 2020]
  $\leadsto$ any $s$, no genericity assumption, returns the canonical basis “s-Popov”

- $F \mod M$ and general modular matrix equations in similar complexity
  [Beckermann-Labahn 1997] [Jeannerod-Neiger-Schost-Villard 2017] [Neiger-Vu 2017]
  $\leadsto$ any $s$, no genericity assumption, returns the canonical “s-Popov” basis
for $F \in \mathbb{K}[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

$\Rightarrow$ basis $K \in \mathbb{K}[X]^{(m-r) \times m}$
for $F \in K[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in K[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $K[X]$-module
- It has rank $m - r$, where $r$ is the rank of $F$

$\Rightarrow$ basis $K \in K[X]^{(m-r) \times m}$

Kernel basis for a constant matrix?

Input matrix $F$

$$\begin{bmatrix}
5 & 6 \\
6 & 1 \\
2 & 6 \\
5 & 2 \\
5 & 6
\end{bmatrix}$$
for $F \in \mathbb{K}[X]^{m\times n}$, its left kernel is

$$\mathcal{K}(F) = \{p \in \mathbb{K}[X]^{1\times m} \mid pF = 0\}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

⇒ basis $K \in \mathbb{K}[X]^{(m-r)\times m}$

kernel basis for a constant matrix? → usual nullspace

input matrix $F$

$$\begin{bmatrix} 5 & 6 \\ 6 & 1 \end{bmatrix}$$

kernel basis $K$

$$\begin{bmatrix} 5 & 6 & 1 & 0 & 0 \\ 0 & 5 & 0 & 1 & 0 \\ 0 & 0 & 3 & 2 & 1 \end{bmatrix}$$
for $F \in K[X]^{m \times n}$, its **left kernel** is

$$\mathcal{K}(F) = \{ p \in K[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $K[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

$\Rightarrow$ basis $K \in K[X]^{(m-r) \times m}$

**kernel basis of the following matrix over $F_2$?**

$$\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
x^2 & x^2 + x + 1 & x^2 + x \\
x^2 + 1 & x^2 & x^2 + x + 1 \\
x^2 & x^2 + x & x^2
\end{bmatrix}$$
for $F \in \mathbb{K}[X]^{m \times n}$, its **left kernel** is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

$\Rightarrow$ basis $K \in \mathbb{K}[X]^{(m-r) \times m}$

Kernel basis of the following matrix over $\mathbb{F}_2$?

input matrix $F$

$$\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}$$

Kernel basis $K$

$$\begin{bmatrix}
x^2 + 1 & x^2 + x + 1 & x^2 + x & 1 & 0 & 0 \\
x^2 & x^2 + x + 1 & x^2 + x & 0 & 1 & 0 \\
x^2 & x^2 + x & x^2 & 0 & 0 & 1 \\
\end{bmatrix}$$
for $F \in K[X]^{m \times n}$, its **left kernel** is

$$\mathcal{K}(F) = \{ p \in K[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $K[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

⇒ **basis** $K \in K[X]^{(m - r) \times m}$

kernel basis of the following block matrix with $G$ nonsingular?

$$\begin{bmatrix} G \\ H \end{bmatrix} \in K[X]^{(n + m) \times n}$$
For $F \in \mathbb{K}[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- It has rank $m - r$, where $r$ is the rank of $F$

⇒ Basis $K \in \mathbb{K}[X]^{(m-r) \times m}$

Kernel basis of the following block matrix with $G$ nonsingular?

Kernel basis $K$

... is left multiple of $[-HG^{-1} \ I_m]$

... $\det(G) [-HG^{-1} \ I_m]$ is left multiple of it

Input matrix $F$

$$\begin{bmatrix} G \\ H \end{bmatrix} \in \mathbb{K}[X]^{(n+m) \times n}$$
for $F \in \mathbb{K}[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} \mid pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

⇒ basis $K \in \mathbb{K}[X]^{(m-r) \times m}$

kernel basis of the following $4 \times 1$ vector with $R \in \mathbb{K}[X] \setminus \{0\}$?

input matrix $F$

$$
\begin{bmatrix}
R \\
R + XR \\
XR + X^2R \\
X^2R + X^3R
\end{bmatrix}
$$
for $F \in \mathbb{K}[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} | pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

$\Rightarrow$ basis $K \in \mathbb{K}[X]^{(m-r) \times m}$

Kernel basis of the following $4 \times 1$ vector with $R \in \mathbb{K}[X] \setminus \{0\}$?

Kernel basis $K$

$$\begin{bmatrix} 1 + X & -1 & \vdots & \vdots \\ 0 & 1 + X & -1 & \vdots \\ 0 & 0 & 1 + X & -1 \end{bmatrix}$$

Input matrix $F$

$$\begin{bmatrix} R \\ R + XR \\ XR + X^2R \\ X^2R + X^3R \end{bmatrix}$$
for $F \in \mathbb{K}[X]^{m \times n}$, its left kernel is

$$\mathcal{K}(F) = \{ p \in \mathbb{K}[X]^{1 \times m} | pF = 0 \}$$

- $\mathcal{K}(F)$ is a $\mathbb{K}[X]$-module
- it has rank $m - r$, where $r$ is the rank of $F$

⇒ basis $K \in \mathbb{K}[X]^{(m-r) \times m}$

inclusion $\mathcal{K}(F) \subset \mathcal{I}(M, F) = \{ p \in \mathbb{K}[X]^{1 \times m} | pF = 0 \mod M \}$

⇒ recover kernel via interpolation with suitable choices of $M$
input:
▶ matrix $\mathbf{F} \in \mathbb{K}[X]^{m \times n}$
▶ $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(\mathbf{F})$ of degree $\leq \delta$

algorithm via interpolation at sufficiently many points

▶ $d \leftarrow \delta + \deg(\mathbf{F}) + 1$
▶ $\alpha \leftarrow$ choose some $(\alpha_1, \ldots, \alpha_d)$ in $\mathbb{K}^d$ (not necessarily distinct)
▶ $\mathbf{P} \in \mathbb{K}[X]^{m \times m} \leftarrow$ reduced basis of $\mathcal{I}(\alpha, \mathbf{F})$
▶ $\mathbf{K} \in \mathbb{K}[X]^{k \times m} \leftarrow$ rows of $\mathbf{P}$ which have degree $\leq \delta$
applications

minimal kernel bases and linear systems

input:
- matrix $\mathbf{F} \in \mathbb{K}[X]^{m \times n}$
- $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(\mathbf{F})$ of degree $\leq \delta$

algorithm via interpolation at sufficiently many points

- $d \leftarrow \delta + \deg(\mathbf{F}) + 1$
- $\alpha \leftarrow$ choose some $(\alpha_1, \ldots, \alpha_d)$ in $\mathbb{K}^d$ (not necessarily distinct)
- $\mathbf{P} \in \mathbb{K}[X]^{m \times m} \leftarrow$ reduced basis of $I(\alpha, \mathbf{F})$
- $\mathbf{K} \in \mathbb{K}[X]^{k \times m} \leftarrow$ rows of $\mathbf{P}$ which have degree $\leq \delta$

$\Rightarrow \mathbf{K}$ is a reduced basis of $\mathcal{K}(\mathbf{F})$
$\Rightarrow$ complexity $O(m^\omega M(\lceil \frac{nd}{m} \rceil) \log(\lceil \frac{nd}{m} \rceil))$
input:
- matrix $\mathbf{F} \in \mathbb{K}[X]^{m \times n}$
- $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(\mathbf{F})$ of degree $\leq \delta$

algorithm via interpolation at sufficiently many points

- $d \leftarrow \delta + \deg(\mathbf{F}) + 1$
- $\alpha \leftarrow$ choose some $(\alpha_1, \ldots, \alpha_d)$ in $\mathbb{K}^d$ (not necessarily distinct)
- $\mathbf{P} \in \mathbb{K}[X]^{m \times m} \leftarrow$ reduced basis of $I(\alpha, \mathbf{F})$
- $\mathbf{K} \in \mathbb{K}[X]^{k \times m} \leftarrow$ rows of $\mathbf{P}$ which have degree $\leq \delta$

$\Rightarrow \mathbf{K}$ is a reduced basis of $\mathcal{K}(\mathbf{F})$
$\Rightarrow$ complexity $O(m^\omega M(\lceil \frac{nd}{m} \rceil) \log(\lceil \frac{nd}{m} \rceil))$

how to find the degree bound $\delta$?
Knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(F)$ of degree $\leq \delta$

- take $d \leftarrow \delta + \deg(F) + 1$ and some $\alpha \leftarrow (\alpha_1, \ldots, \alpha_d)$ in $\mathbb{K}^d$
- $P \in \mathbb{K}[X]^{m \times m}$ reduced basis of $J(\alpha, F)$
- $K \in \mathbb{K}[X]^{k \times m}$ rows of $P$ which have degree $\leq \delta$

$\Rightarrow K$ is a reduced basis of $\mathcal{K}(F)$
applications

minimal kernel bases and linear systems

knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(F)$ of degree $\leq \delta$

- take $d \leftarrow \delta + \deg(F) + 1$ and some $\alpha \leftarrow (\alpha_1, \ldots, \alpha_d)$ in $K^d$
- $P \in K[X]^{m \times m}$ reduced basis of $I(\alpha, F)$
- $K \in K[X]^{k \times m}$ rows of $P$ which have degree $\leq \delta$

$\Rightarrow K$ is a reduced basis of $\mathcal{K}(F)$

proof:
$\Rightarrow K$ is reduced by construction

. $K$ satisfies $KF = 0 \mod (X - \alpha_1) \cdots (X - \alpha_d)$
. and $\deg(K) \leq \delta$, hence $\deg(KF) \leq \delta + \deg(F) < d$
$\Rightarrow KF = 0$, i.e. the rows of $K$ are in $\mathcal{K}(F)$

. let $B \in K[X]^{(m-r) \times m}$ be a basis of $\mathcal{K}(F)$ of degree $\leq \delta$
. then $B = UP$ for some $U$
. by the predictable degree property, in fact $B = VK$
$\Rightarrow$ any vector in $\mathcal{K}(F)$ is generated by $K$
knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(\mathbf{F})$ of degree $\leq \delta$

**how to find the degree bound $\delta$?**

- A specific bound may be known from the context e.g. gcd, “row bases”

- A general bound is $\delta = n \deg(\mathbf{F})$

- Yields complexity $O^\sim(m^\omega \lceil \frac{n^2 \deg(\mathbf{F})}{m} \rceil)$

**proof:**

up to row and column permutation, $\mathbf{F} = [\mathbf{G} \ast \mathbf{H}]$ with $\mathbf{G} \in \mathbb{K}[\mathbf{X}]_{r \times r}$ nonsingular

then, $\mathbb{K}(\mathbf{F}) = \mathbb{K}([\mathbf{G} \mathbf{H}])$

the matrix $[\mathbf{G} \mathbf{H} - \mathbf{G}^{-1} \frac{\det(\mathbf{G})}{\det(\mathbf{G})} \mathbf{I}_m - r]$ has polynomial entries, it has rank $m - r$ and its rows are in $\mathbb{K}(\mathbf{F})$, it has degree $\leq \max(\deg \det(\mathbf{G}), \deg(\mathbf{H}) + (r - 1) \deg(\mathbf{G})) \leq r \deg(\mathbf{F})$

by degree minimality of reduced matrices, any reduced basis of $\mathbb{K}(\mathbf{F})$ must have degree $\leq r \deg(\mathbf{F})$

**rules of thumb, generically:**

"quantity of information is preserved" + "degrees in reduced basis are uniform" $\Rightarrow (m - r) m \deg(\mathbb{K}) \approx mn \deg(\mathbb{F})$ $\Leftrightarrow \deg(\mathbb{K}) \approx n m - r \deg(\mathbb{F})$, which is $\leq n m - n \deg(\mathbb{F})$
Knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(F)$ of degree $\leq \delta$,

**how to find the degree bound $\delta$?**

- A specific bound may be known from the context, e.g., gcd, “row bases”
- A general bound is $\delta = n \deg(F)$
- Yields complexity $O^\sim(m^\omega \left\lceil \frac{n^2 \deg(F)}{m} \right\rceil)$

**Proof:**

Complexity $O^\sim(m^\omega \left\lceil \frac{nd}{m} \right\rceil)$

With $d = \delta + \deg(F) + 1 = (n + 1) \deg(F) + 1$
Knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(\mathbf{F})$ of degree $\leq \delta$

**how to find the degree bound $\delta$?**

A specific bound may be known from the context e.g. gcd, “row bases”

- A general bound is $\delta = n \deg(\mathbf{F})$
- Yields complexity $O^\sim(m^\omega \left\lfloor \frac{n^2 \deg(\mathbf{F})}{m} \right\rfloor)$

**proof:**

Up to row and column permutation, $\mathbf{F} = [\mathbf{G} \ast \mathbf{H}]$

With $\mathbf{G} \in \mathbb{K}[X]^{r \times r}$ nonsingular

Then, $\mathcal{K}(\mathbf{F}) = \mathcal{K}([\mathbf{G} \ast \mathbf{H}])$

The matrix $[-\mathbf{H}(\det(\mathbf{G})\mathbf{G}^{-1}) \quad \det(\mathbf{G})\mathbf{I}_{m-r}]$ has polynomial entries,

It has rank $m-r$ and its rows are in $\mathcal{K}(\mathbf{F})$,

It has degree $\leq \max(\deg \det(\mathbf{G}), \deg(\mathbf{H}) + (r-1)\deg(\mathbf{G})) \leq r\deg(\mathbf{F})$

By degree minimality of reduced matrices,

Any reduced basis of $\mathcal{K}(\mathbf{F})$ must have degree $\leq r\deg(\mathbf{F})$
knowing $\delta \in \mathbb{Z}_{>0}$ such that there exists a basis of $\mathcal{K}(F)$ of degree $\leq \delta$

**how to find the degree bound $\delta$?**

A specific bound may be known from the context, e.g. gcd, “row bases”

- A general bound is $\delta = n \deg(F)$
- Yields complexity $O(\sim m^\omega \left\lfloor \frac{n^2 \deg(F)}{m} \right\rfloor)$

- Rules of thumb, generically:
  - “Quantity of information is preserved”
  - “Degrees in reduced basis are uniform”

$$\leadsto (m-r)m \deg(K) \approx mn \deg(F)$$

$$\Leftrightarrow \deg(K) \approx \frac{n}{m-r} \deg(F), \text{ which is } \leq \frac{n}{m-n} \deg(F)$$

Example: if $F$ is $m \times \frac{m}{2}$, generically $\deg(K) = \deg(F)$

$\Rightarrow d = 2 \deg(F) + 1$ and complexity $O(\sim m^\omega \deg(F))$
applications

minimal kernel bases and linear systems

breakthrough [Zhou-Labahn-Storjohann 2012]

- complexity $O^\sim(m^\omega \lceil \frac{n \deg(F)}{m} \rceil)$ without assumption
- computes $s$-reduced basis of $\mathcal{K}(F)$ for $s = r\deg(F)$

- $n$ large: divide and conquer on $n$, via residual + basis multiplication
  $\leadsto$ partial linearization for multiplying matrices with weakly unbalanced degrees
- $n$ small: use fast approximation/interpolation algorithms
  $\leadsto$ well-chosen $d$ yields at least half the kernel efficiently
applications

minimal kernel bases and linear systems

breakthrough [Zhou-Labahn-Storjohann 2012]

- complexity $O^\sim(m^\omega \lceil \frac{n \deg(F)}{m} \rceil)$ without assumption
- computes $s$-reduced basis of $\mathcal{K}(F)$ for $s = \text{rdeg}(F)$

- $n$ large: divide and conquer on $n$, via residual + basis multiplication
- $\rightsquigarrow$ partial linearization for multiplying matrices with weakly unbalanced degrees
- $n$ small: use fast approximation/interpolation algorithms
- $\rightsquigarrow$ well-chosen $d$ yields at least half the kernel efficiently

if $n > \frac{m}{2}$:

$K_1 \leftarrow$ recursive call on first $\frac{n}{2}$ columns of $F$, and shift $s$

$G \leftarrow$ multiply $K_1 \cdot F^*, \frac{n}{2}..n$ (last $\frac{n}{2}$ columns of $F$)

$K_2 \leftarrow$ recursive call on $G$, and shift $t = \text{rdeg}_s(K_1)$

return $K_2K_1$
minimal kernel bases and linear systems

breakthrough [Zhou-Labahn-Storjohann 2012]

- complexity $O^\sim(m^\omega \left\lceil \frac{n \deg(F)}{m} \right\rceil)$ without assumption
- computes $s$-reduced basis of $\mathcal{K}(F)$ for $s = r\deg(F)$

$n$ large: divide and conquer on $n$, via residual + basis multiplication
$\rightsquigarrow$ partial linearization for multiplying matrices with weakly unbalanced degrees

$n$ small: use fast approximation/interpolation algorithms
$\rightsquigarrow$ well-chosen $d$ yields at least half the kernel efficiently

if $n \leq \frac{m}{2}$:

$\delta \leftarrow$ degree of kernel basis expected generically

$d \leftarrow \delta + \deg(F) + 1$ and take some $\alpha \leftarrow (\alpha_1, \ldots, \alpha_d)$ in $\mathbb{K}^d$

$P \in \mathbb{K}[X]^{m \times m} \leftarrow$ $s$-reduced basis of $J(\alpha, F)$

$K_1, Q \leftarrow$ rows of $P$ which are in $\mathcal{K}(F)$ / which are not in $\mathcal{K}(F)$

$K_2 \leftarrow$ recursive call on $\frac{1}{(X-\alpha_1)\cdots(X-\alpha_d)}QF$, return $[K_1 \ K_2]$
linear system solving:
given $A \in \mathbb{K}[X]^{m \times m}$ nonsingular and $v \in \mathbb{K}[X]^{1 \times m}$
find $u \in \mathbb{K}[X]^{1 \times m}$ and $g \in \mathbb{K}[X]$ such that

$$uA = gv$$

and $g$ has minimal degree.

- the equation has a solution: $u = gvA^{-1}$ with $g = \det(A)$
- but there is often no polynomial solution with $g = 1$
- target complexity? (recall that $\det(A)A^{-1}$ can have degree $\approx m \deg(A)$)
- propose an algorithm based on a kernel computation
linear system solving:
given $A \in K[X]^{m \times m}$ nonsingular and $v \in K[X]^{1 \times m}$
find $u \in K[X]^{1 \times m}$ and $g \in K[X]$ such that
\[ uA = gv \quad \text{and} \quad g \text{ has minimal degree.} \]

- the equation has a solution: $u = g v A^{-1}$ with $g = \det(A)$
- but there is often no polynomial solution with $g = 1$
- target complexity? (recall that $\det(A) A^{-1}$ can have degree $\approx m \deg(A)$)
- propose an algorithm based on a kernel computation

compute $[u \ g] \in K[X]^{1 \times (m+1)}$ kernel basis of $F = \begin{bmatrix} A \\ -v \end{bmatrix} \in K[X]^{(m+1) \times m}$

- using the shift $s = (r\deg(A), \deg(v))$
- complexity $O^\sim(m^\omega \max(\deg(A), \deg(v)))$
- $u, g$ is a solution to the equation $uA = gv$
- minimality of $\deg(g)$ follows from basis of $\mathcal{K}(F)$
applications

fast gcd and extended gcd

**gcd**
input: f and g univariate polynomials in $\mathbb{K}[X]$  
output: $h = \gcd(f, g)$

**xgcd**
input: f and g univariate polynomials in $\mathbb{K}[X]$  
output: $(u, v, h)$ where $h = \gcd(f, g) = uf + vg$
applications

fast gcd and extended gcd

**gcd**

input: \( f \) and \( g \) univariate polynomials in \( \mathbb{K}[X] \)

output: \( h = \gcd(f, g) \)

**xgcd**

input: \( f \) and \( g \) univariate polynomials in \( \mathbb{K}[X] \)

output: \((u, v, h)\) where \( h = \gcd(f, g) = uf + vg \)

**some notation:**

. polynomials \( \bar{f} = f/h \) and \( \bar{g} = g/h \)
. \( m = \deg(f) \) and \( n = \deg(g) \)
. \( \ell = \deg(h) \)

\( \leadsto \) then \( \deg(\bar{f}) = m - \ell \) and \( \deg(\bar{g}) = n - \ell \)

\( \bar{f} \) and \( \bar{g} \) are coprime
we assume \( m, n > 0 \)
hence \( \ell \leq \min(m, n) \)

earlier in the course:

**claim:** gcd and xgcd are solved in \( O(M(d) \log(d)) \)

where \( d = \max(m, n) \)
applications

fast gcd and extended gcd

input: f and g univariate polynomials in \( K[X] \)
output: \( h = \gcd(f, g) \)

some notation:
  . polynomials \( \bar{f} = f/h \) and \( \bar{g} = g/h \) \( \bar{f} \) and \( \bar{g} \) are coprime
  . \( m = \deg(f) \) and \( n = \deg(g) \) we assume \( m, n > 0 \)

result: gcd is solved in \( O(M(\max(m, n)) \log(\max(m, n))) \)
Applications

Fast GCD and Extended GCD

Input: \( f \) and \( g \) univariate polynomials in \( K[X] \).
Output: \( h = \gcd(f, g) \).

Some notation:
- Polynomials \( \bar{f} = f/h \) and \( \bar{g} = g/h \). \( \bar{f} \) and \( \bar{g} \) are coprime.
- \( m = \deg(f) \) and \( n = \deg(g) \). We assume \( m, n > 0 \).

Result: GCD is solved in \( O(M(\max(m, n))) \log(\max(m, n))) \).

Lemma: \([\bar{g} \bar{f}]\) is a basis of the left kernel of \([f \ g]\).

Proof:
This kernel has rank 1 (\( f \) and \( g \) are nonzero).

Let \([a \ b]\) be a basis of it; all other bases are \([ca \ cb]\) for some \( c \in K \setminus \{0\}\).

Since \([\bar{g} \bar{f}]\)\([f \ g]\) = \(-\frac{g}{h}f + \frac{f}{h}g = 0\), we get \([\bar{g} \bar{f}]\) = \([\lambda a \ \lambda b]\) for some \( \lambda \in K[X] \setminus \{0\}\).

Then \( \lambda \) divides \( \bar{f} \) and \( \bar{g} \), so \( \lambda \) is a nonzero constant.
fast gcd and extended gcd

input: \( f \) and \( g \) univariate polynomials in \( \mathbb{K}[X] \)

output: \( h = \gcd(f, g) \)

some notation:

- polynomials \( \bar{f} = f/h \) and \( \bar{g} = g/h \) \( \bar{f} \) and \( \bar{g} \) are coprime
- \( m = \deg(f) \) and \( n = \deg(g) \) we assume \( m, n > 0 \)

result: gcd is solved in \( O(M(\max(m, n)) \log(\max(m, n))) \)

lemma: \( [-\bar{g} \quad \bar{f}] \) is a basis of the left kernel of \( [f]_g \)

algorithm: kernel basis via interpolation at sufficiently many points

- the input matrix \( F = [f]_g \) has degree \( \max(m, n) \)
- the sought kernel basis has degree at most \( \delta = \max(m, n) \)

\[
\begin{align*}
1. & \text{pick } \delta + \deg(F) + 1 = 2\delta + 1 \text{ points } \alpha \in \mathbb{K}^{2\delta + 1} & \text{O}(1) \\
\Rightarrow & \text{find } [-\bar{g} \quad \bar{f}] \text{ via a reduced basis of } J(\alpha, [f]_g) & O(M(\delta) \log(\delta)) \\
2. & \text{deduce } h = g/\bar{g} & O(M(\delta))
\end{align*}
\]
applications

fast gcd and extended gcd

\textbf{xgcd} input: \( f \) and \( g \) univariate polynomials in \( K[X] \)
output: \((u, v, h)\) where \( h = \gcd(f, g) = uf + vg \)

\textbf{some notation:}
\begin{itemize}
  \item polynomials \( \bar{f} = f/h \) and \( \bar{g} = g/h \) \( \bar{f} \) and \( \bar{g} \) are coprime
  \item \( m = \deg(f) \), \( n = \deg(g) \), \( \ell = \deg(h) \)
  \item \( \deg(\bar{f}) = m - \ell \) and \( \deg(\bar{g}) = n - \ell \)
\end{itemize}
\( m, n > 0, \ell \leq \min(m, n) \)
some notation:

. polynomials $\bar{f} = f/h$ and $\bar{g} = g/h$
. $m = \deg(f)$, $n = \deg(g)$, $\ell = \deg(h)$

$\Rightarrow$ $\deg(\bar{f}) = m - \ell$ and $\deg(\bar{g}) = n - \ell$

lemma:

. there exists a unique $(u, v)$ in $\mathbb{K}[X]^2$ such that

\[
\begin{cases}
uf + vg = h, \\
\deg(u) < n - \ell \quad \text{and} \quad \deg(v) < m - \ell.
\end{cases}
\]

. for this $(u, v) \in \mathbb{K}[X]^2$ one has

\[
\begin{bmatrix}
u & v \\
-\bar{g} & \bar{f}
\end{bmatrix}
\begin{bmatrix}
f \\
g
\end{bmatrix} =
\begin{bmatrix}
h \\
0
\end{bmatrix},
\]

and the leftmost matrix in this identity is unimodular.
applications

fast gcd and extended gcd

input: $f$ and $g$ univariate polynomials in $\mathbb{K}[X]$ output: $(u, v, h)$ where $h = \gcd(f, g) = uf + vg$

some notation:
- polynomials $\bar{f} = f/h$ and $\bar{g} = g/h$ $\bar{f}$ and $\bar{g}$ are coprime
- $m = \deg(f)$, $n = \deg(g)$, $\ell = \deg(h)$ $m, n > 0$, $\ell \leq \min(m, n)$
- $\Rightarrow \deg(\bar{f}) = m - \ell$ and $\deg(\bar{g}) = n - \ell$

theorem:
- defining $R = \begin{bmatrix} \rev(u, n - \ell - 1) & \rev(v, m - \ell - 1) \\ -\rev(\bar{g}, n - \ell) & \rev(f, m - \ell) \end{bmatrix} \in \mathbb{K}[X]^{2 \times 2}$,
- one has: $R \begin{bmatrix} \rev(f, m) \\ \rev(g, n) \end{bmatrix} = \begin{bmatrix} x^{m+n-2\ell-1} \rev(h, \ell) \\ 0 \end{bmatrix}$
- the matrix $R$ is a $(-n, -m)$-reduced basis of $J(0, \begin{bmatrix} \rev(f, m) \\ \rev(g, n) \end{bmatrix})$

$$= \left\{ [p, q] \in \mathbb{K}[X]^{1 \times 2} \mid [p, q] \begin{bmatrix} \rev(f, m) \\ \rev(g, n) \end{bmatrix} = 0 \mod x^{m+n-2\ell-1} \right\}$$
Applications

Fast GCD and Extended GCD

**xgcd**

Input: \( f \) and \( g \) univariate polynomials in \( \mathbb{K}[X] \)

Output: \((u, v, h)\) where \( h = \text{gcd}(f, g) = uf + vg\)

Some notation:

- Polynomials \( \tilde{f} = f/h \) and \( \tilde{g} = g/h \) \( \tilde{f} \) and \( \tilde{g} \) are coprime
- \( m = \deg(f) \), \( n = \deg(g) \), \( \ell = \deg(h) \)
- \( \deg(\tilde{f}) = m - \ell \) and \( \deg(\tilde{g}) = n - \ell \)

**Theorem:**

- Defining \( R = \begin{bmatrix} \text{rev}(u, n - \ell - 1) & \text{rev}(v, m - \ell - 1) \\ -\text{rev}(\tilde{g}, n - \ell) & \text{rev}(\tilde{f}, m - \ell) \end{bmatrix} \in \mathbb{K}[X]^{2\times 2}, \)
- One has: \( R \begin{bmatrix} \text{rev}(f, m) \\ \text{rev}(g, n) \end{bmatrix} = \begin{bmatrix} x^{m+n-2\ell-1} \text{rev}(h, \ell) \\ 0 \end{bmatrix} \)

\( \ell \) is unknown!

- The matrix \( R \) is a \((-n, -m)\)-reduced basis of \( J(0, [\text{rev}(f, m) \text{rev}(g, n)]) \)

\[ \begin{cases} [p \ q] \in \mathbb{K}[X]^{1\times 2} & [p \ q] \begin{bmatrix} \text{rev}(f, m) \\ \text{rev}(g, n) \end{bmatrix} = 0 \mod x^{m+n-2\ell-1} \end{cases} \]
**applications**

**fast gcd and extended gcd**

**xgcd**

input: \( f \) and \( g \) univariate polynomials in \( \mathbb{K}[X] \)

output: \((u, v, h)\) where \( h = \gcd(f, g) = uf + vg \)

**some notation:**
- polynomials \( \tilde{f} = f/h \) and \( \tilde{g} = g/h \) \( \tilde{f} \) and \( \tilde{g} \) are coprime
- \( m = \deg(f), \ n = \deg(g), \ \ell = \deg(h) \) \( m, n > 0, \ \ell \leq \min(m, n) \)
- \( \nabla \) \( \deg(f) = m - \ell \) and \( \deg(\tilde{g}) = n - \ell \)

**corollary:** \( \text{xgcd in } O(M(d) \log(d)) \)

for any \( d \geq n + m - 2\ell - 1 \)

let \( e = d - (n + m - 2\ell - 1) \)

e.g. \( d = n + m + 1 \)

hence \( e = 2\ell \)

then \( \begin{bmatrix} x^e & 0 \\ 0 & 1 \end{bmatrix} R = \begin{bmatrix} x^e \ \text{rev}(u, n - \ell - 1) & x^e \ \text{rev}(v, m - \ell - 1) \\ - \text{rev}(\tilde{g}, n - \ell) & \text{rev}(\tilde{f}, m - \ell) \end{bmatrix} \)

is a \((-n, -m)\)-reduced basis of

\[
\begin{bmatrix} p & q \end{bmatrix} \in \mathbb{K}[X]^{1 \times 2} \quad \begin{bmatrix} p & q \end{bmatrix} \begin{bmatrix} \text{rev}(f, m) \\ \text{rev}(g, n) \end{bmatrix} = 0 \mod x^d
\]


a row basis of a matrix \( F \in \mathbb{K}[X]^{m \times n} \) is a basis of its \( \mathbb{K}[X] \)-row space \( \{ pF \mid p \in \mathbb{K}[X]^{1 \times m} \} \) represented as \( R \in \mathbb{K}[X]^{r \times n} \), where \( r \) is the rank of \( F \) \( \implies F = UR \) for some \( U \in \mathbb{K}[X]^{m \times r} \)
a row basis of a matrix \( F \in K[X]^{m \times n} \) is a basis of its \( K[X] \)-row space

\[ \{ pF \mid p \in K[X]^{1 \times m} \} \]

\( \rightsquigarrow \) represented as \( R \in K[X]^{r \times n} \), where \( r \) is the rank of \( F \)

\( \rightsquigarrow F = UR \) for some \( U \in K[X]^{m \times r} \)

examples:

- row basis for \( F \in K[X]^{m \times m} \) nonsingular?
- row basis of \( \begin{bmatrix} f \\ g \end{bmatrix} \) for \( f, g \) coprime polynomials?
- \( K \in K[X]^{(m-r) \times m} \) a left kernel basis of \( F \in K[X]^{m \times n} \)
- row basis of \( K \)? column basis of \( K \)?
a row basis of a matrix \( F \in \mathbb{K}[X]^{m \times n} \) is a basis of its \( \mathbb{K}[X] \)-row space 

\[ \Rightarrow \text{represented as } R \in \mathbb{K}[X]^{r \times n}, \text{ where } r \text{ is the rank of } F \]

\( \Rightarrow F = UR \) for some \( U \in \mathbb{K}[X]^{m \times r} \)

equations:

- row basis for \( F \in \mathbb{K}[X]^{m \times m} \) nonsingular? \( R = F \)
- row basis of \( \begin{bmatrix} f \\ g \end{bmatrix} \) for \( f, g \) coprime polynomials?
- \( K \in \mathbb{K}[X]^{(m-r) \times m} \) a left kernel basis of \( F \in \mathbb{K}[X]^{m \times n} \)
- row basis of \( K \)? column basis of \( K \)?
a row basis of a matrix $F \in \mathbb{K}[X]^{m \times n}$ is a basis of its $\mathbb{K}[X]$-row space represented as $R \in \mathbb{K}[X]^{r \times n}$, where $r$ is the rank of $F$.

$F = UR$ for some $U \in \mathbb{K}[X]^{m \times r}$.

**Examples:**
- row basis for $F \in \mathbb{K}[X]^{m \times m}$ nonsingular? $R = F$
- row basis of $\begin{bmatrix} f \\ g \end{bmatrix}$ for $f, g$ coprime polynomials? $R = [1]$
- $K \in \mathbb{K}[X]^{(m-r) \times m}$ a left kernel basis of $F \in \mathbb{K}[X]^{m \times n}$ row basis of $K$? column basis of $K$?
a row basis of a matrix $F \in \mathbb{K}[X]^{m \times n}$ is a basis of its $\mathbb{K}[X]$-row space

$\leadsto$ represented as $R \in \mathbb{K}[X]^{r \times n}$, where $r$ is the rank of $F$

$\leadsto F = UR$ for some $U \in \mathbb{K}[X]^{m \times r}$

examples:

- row basis for $F \in \mathbb{K}[X]^{m \times m}$ nonsingular? $R = F$

- row basis of $\begin{bmatrix} f \\ g \end{bmatrix}$ for $f, g$ coprime polynomials? $R = [1]$

- $K \in \mathbb{K}[X]^{(m-r) \times m}$ a left kernel basis of $F \in \mathbb{K}[X]^{m \times n}$
  row basis of $K$? column basis of $K$? $R = K$ and $C = I_{m-r}$

$K$ has full rank so $C$ is $(m-r) \times (m-r)$ nonsingular
and by definition $K = CK$ for some $\bar{K}$
so $KF = 0 \Rightarrow \bar{K}F = 0$, hence $\bar{K} = VK$
from $K = CVK$, with $K$ having full row rank, we deduce $CV = I_{m-r}$
a row basis of a matrix $F \in \mathbb{K}[X]^{m \times n}$ is a basis of its $\mathbb{K}[X]$-row space

$\mapsto$ represented as $R \in \mathbb{K}[X]^{r \times n}$, where $r$ is the rank of $F$

$\mapsto F = UR$ for some $U \in \mathbb{K}[X]^{m \times r}$

applications:
- compute an $s$-reduced basis of the row space
- verify that a matrix is a kernel basis
- triangularization: Hermite normal form and determinant
a row basis of a matrix $F \in \mathbb{K}[X]^{m \times n}$ is a basis of its $\mathbb{K}[X]$-row space

\[ \{ pF \mid p \in \mathbb{K}[X]^{1 \times m} \} \]

\( \leadsto \) represented as $R \in \mathbb{K}[X]^{r \times n}$, where $r$ is the rank of $F$

\( \leadsto F = UR \) for some $U \in \mathbb{K}[X]^{m \times r}$

applications:
- compute an $s$-reduced basis of the row space
- verify that a matrix is a kernel basis
- triangularization: Hermite normal form and determinant

algorithm:
- $K \leftarrow$ left kernel basis for $F$
- $G \leftarrow$ right kernel basis for $K$
- $R \leftarrow$ matrix such that $F = GR$

complexity $O^\sim(mn^{\omega - 1} \deg(F))$, assuming $m \geq n$ [Zhou-Labahn, 2013]
triangularization of $m \times m$ matrix $A$ using $\frac{m}{2} \times \frac{m}{2}$ blocks

\[
\begin{bmatrix}
K_1 & K_2 \\
A_3 & A_4
\end{bmatrix}
\begin{bmatrix}
A_1 \\
A_2
\end{bmatrix}
= 
\begin{bmatrix}
R & * \\
0 & B
\end{bmatrix}
\]

kernel basis of $\begin{bmatrix} A_1 \\ A_3 \end{bmatrix}$

$K_1 A_2 + K_2 A_4$

row basis of $\begin{bmatrix} A_1 \\ A_3 \end{bmatrix}$

applications

perspectives — triangularization

Triangularization of $m \times m$ matrix $A$ using $\frac{m}{2} \times \frac{m}{2}$ blocks

\[
\begin{bmatrix}
K_1 & K_2
\end{bmatrix}
\begin{bmatrix}
A_1 & A_2 \\
A_3 & A_4
\end{bmatrix} =
\begin{bmatrix}
R & * \\
0 & B
\end{bmatrix}
\]

Main property: $\begin{bmatrix}
* & * \\
K_1 & K_2
\end{bmatrix}$ is unimodular

- Hermite form of $A = \text{Hermite form of } \begin{bmatrix}
R & * \\
0 & B
\end{bmatrix}$
- $\det(A) = \det(R) \det(B)$

Hermite normal form and determinant in $O^\sim(m^\omega \deg(A))$

given a **sparse** matrix $A \in \mathbb{K}^{n \times n}$:

- solve a linear system $Au = v$
- compute the **minimal polynomial** of $A$

. sparse means that $A$ has a large proportion of zero entries
. goal: exploit sparsity to do better than exponent $\omega$

block Wiedemann approach, for block dimension $m$:
1. choose random blocking matrices $U, V \in \mathbb{K}^{n \times m}$
2. compute **linearly recurrent sequence of matrices** in $\mathbb{K}^{m \times m}$
   
   $U^T V, U^T AV, \ldots, U^T A^k V, \ldots$
3. find polynomial matrix generator $P \in \mathbb{K}[X]^{m \times m}$ of this sequence
given a **sparse** matrix $A \in \mathbb{K}^{n \times n}$:

- solve a linear system $Au = v$
- compute the **minimal polynomial** of $A$

+ sparse means that $A$ has a large proportion of zero entries
+ goal: exploit sparsity to do better than exponent $\omega$

---


block Wiedemann approach, for block dimension $m$:

1. choose random blocking matrices $U, V \in \mathbb{K}^{n \times m}$
2. compute **linearly recurrent sequence of matrices** in $\mathbb{K}^{m \times m}$
   
   $U^T V, U^T AV, \ldots, U^T A^k V, \ldots$
3. find polynomial matrix generator $P \in \mathbb{K}[X]^{m \times m}$ of this sequence

+ generically, $d = 2 \frac{n}{m} - 1$ terms of the sequence are sufficient
+ step 3 is **matrix-Padé approx.**, in $O^\sim(m^\omega d) = O^\sim(m^\omega - 1 n)$
+ often, $m$ is taken as the **number of threads** available for parallel computation of the matrix sequence
<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>introduction</strong></td>
<td>• rational approximation and interpolation</td>
</tr>
<tr>
<td></td>
<td>• the vector case</td>
</tr>
<tr>
<td></td>
<td>• pol. matrices: reminders and motivation</td>
</tr>
<tr>
<td><strong>shifted reduced forms</strong></td>
<td>• reducedness: examples and properties</td>
</tr>
<tr>
<td></td>
<td>• shifted forms and degree constraints</td>
</tr>
<tr>
<td></td>
<td>• stability under multiplication</td>
</tr>
<tr>
<td><strong>fast algorithms</strong></td>
<td>• iterative algorithm and output size</td>
</tr>
<tr>
<td></td>
<td>• base case: modulus of degree 1</td>
</tr>
<tr>
<td></td>
<td>• recursion: residual and basis multiplication</td>
</tr>
<tr>
<td><strong>applications</strong></td>
<td>• minimal kernel bases and linear systems</td>
</tr>
<tr>
<td></td>
<td>• fast gcd and extended gcd</td>
</tr>
<tr>
<td></td>
<td>• perspectives</td>
</tr>
</tbody>
</table>