Some ideas for efficient implementation of algorithms for polynomial matrix computations

SageFlintDays, University of Warwick, December 18, 2011

Arne Storjohann
University of Waterloo

Introduction: integers versus univariate polynomials

Integers:

- well suited ring for a binary computer
  $\rightarrow$ array of limbs of 64 bits each

- primes are quite dense
  $\rightarrow$ prime number theorem: there are about $N/\ln N$ primes $\leq N$
  $\rightarrow$ more than 10^{15} 64-bit primes
  $\rightarrow$ enough for all practical purposes

Polynomials $\mathsf{K}[x]$:

- dense, sparse, supersparse (lacunary)?

- what is the coefficient field K?
  $\rightarrow$ $\mathsf{K} = \mathbb{Z}/(p)$, $\mathsf{K} = \mathrm{GF}(2)$, $\mathsf{K} = \mathbb{Q}$
  $\rightarrow$ K an extension field of one of the above

- and then... what about
  multivariate? coeffcient ring not a field?

The core problem: Polynomial matrix multiplication

Representation:

- matrices of polynomials:

$$F = \begin{bmatrix} x+1 & 4x^2+3x \\ 4x^2+3x+1 & 4x^2+6x+2 \end{bmatrix}$$

- polynomials with matrix coefficients:

$$F = \begin{bmatrix} 0 & 4 \\ 4 & 4 \end{bmatrix} x^2 + \begin{bmatrix} 1 & 3 \\ 3 & 6 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}$$

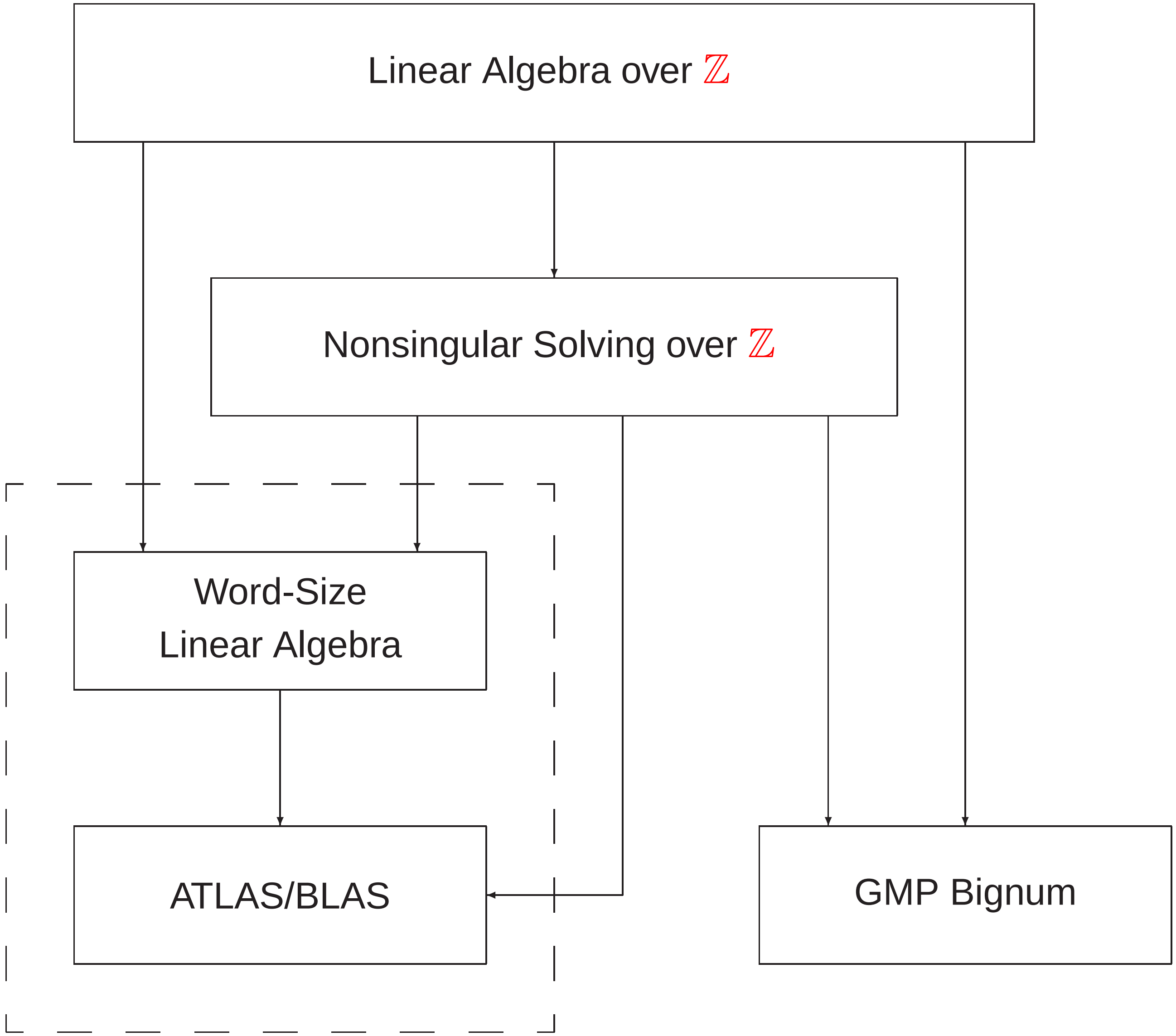
- compact:

$$F = \left[\begin{array}{cc|cc|cc} 0 & 4 & 1 & 3 & 1 & 0 \\ 4 & 4 & 3 & 6 & 1 & 2 \end{array}\right]$$

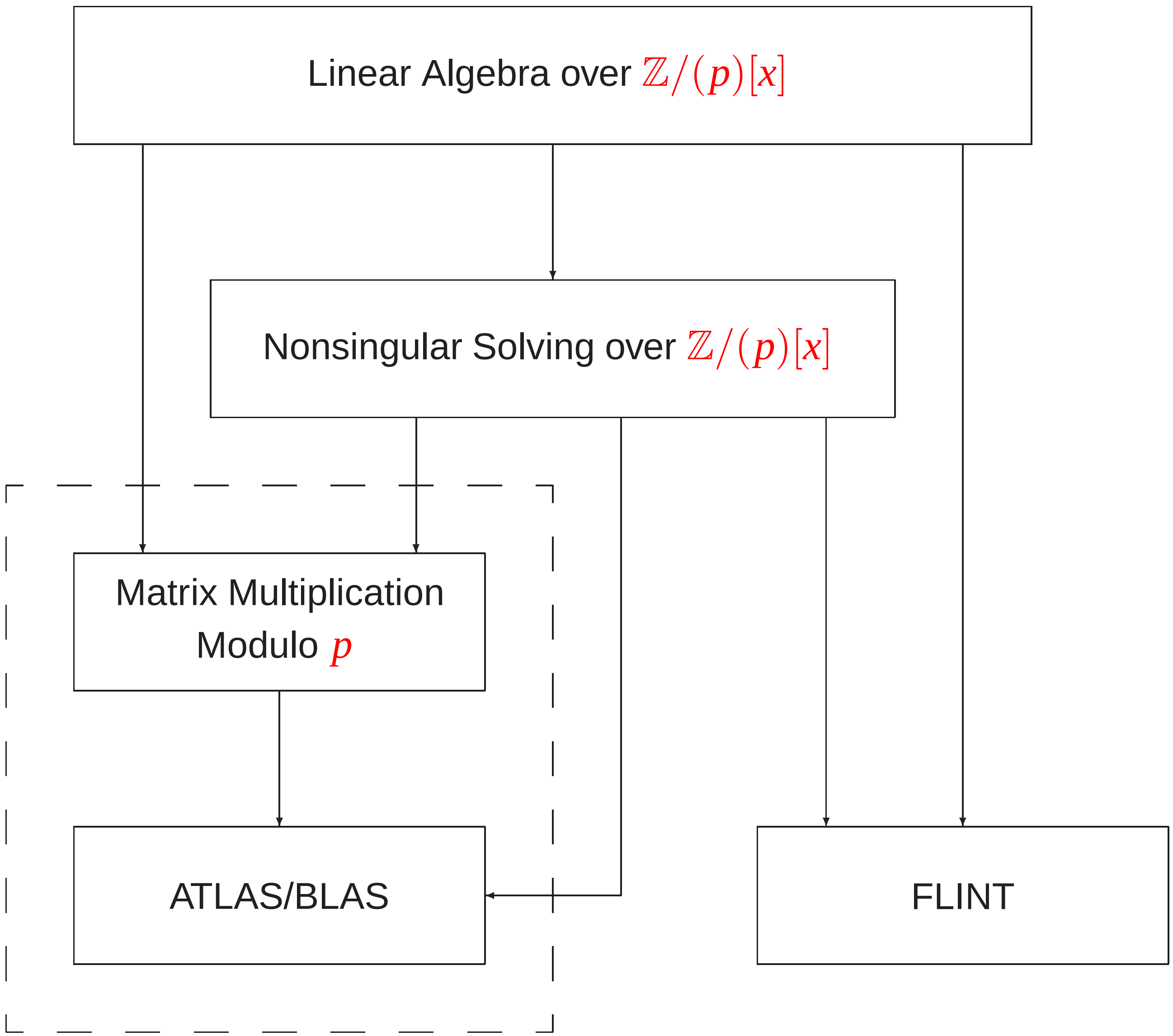
Methods:

1. triply nested for loop using polynomial multiplication (easiest)
2. via integer matrix multiplication: bit packing (FLINT [Frederik J.])
3. evaluation and interpolation (asymtoticaly fastest)
   $\rightarrow$ in-place truncated FFT [Harvey and Roche, 2010]?

Organization of the Integer Matrix Library: IML



Organization of the Polynomial Matrix Library: PML?



Example of linear solving: $\mathsf{K} = \mathbb{Z}/(7)$, $n = 5$, $d = 2$

Input:

$$A = \begin{bmatrix} 6x^2+6x+4 & 2x^2+2x+5 & 3x^2+2x+1 & 5x+3 & 2x^2+6x+6 \\ 4x^2+4x & 2x^2+5x+3 & 5x+4 & 3x^2+5x & 4x^2+x+1 \\ 6x^2+2 & 2x^2+x+6 & 4x^2+2x+2 & 4x^2+5x+1 & x^2+x \\ 3x^2+x+4 & 5x+6 & 4x^2+2x+1 & 2x^2+6x+2 & 3x^2+x \\ x^2+2x+6 & 2x^2+5x+5 & 4x^2+4x & 6 & x^2+x+3 \end{bmatrix} \quad b = \begin{bmatrix} x^2+x+2 \\ 5x^2+x+2 \\ 3x^2+3x+5 \\ x^2+5x+3 \\ 2x+2 \end{bmatrix}$$

Output:

$$v := A^{-1}b = \begin{bmatrix} \frac{4x^9+2x^8+5x^7+6x^6+x^5+3x^4+6x^3+2x+2}{4x^{10}+x^9+3x^8+4x^7+4x^6+2x^5+6x^4+6x^3+2x^2+5x+3} \\ \frac{6x^{10}+2x^9+x^7+3x^6+x^5+6x^4+3x^3+6x^2+2x+1}{4x^{10}+x^9+3x^8+4x^7+4x^6+2x^5+6x^4+6x^3+2x^2+5x+3} \\ \frac{x^{10}+5x^9+6x^8+3x^6+5x^5+3x^4+5x^3+x^2+3x+5}{4x^{10}+x^9+3x^8+4x^7+4x^6+2x^5+6x^4+6x^3+2x^2+5x+3} \\ \frac{3x^{10}+3x^9+6x^8+6x^7+5x^6+5x^4+6x^3+x^2+x+2}{4x^{10}+x^9+3x^8+4x^7+4x^6+2x^5+6x^4+6x^3+2x^2+5x+3} \\ \frac{5x^{10}+4x^9+5x^8+2x^7+x^6+2x^5+5x^4+5x^3+3x^2+6x+4}{4x^{10}+x^9+3x^8+4x^7+4x^6+2x^5+6x^4+6x^3+2x^2+5x+3} \end{bmatrix}$$

Outline of Y-adic lifting for system solving

1. Radix expansion of solution: $Y = x^2$

$$\frac{5x^2+6x+3}{x^2+4x+3} \equiv (1+3x)+(2+x)Y+(1+5x)Y^2 \pmod{Y^3}$$

2. Radix conversion:

$$(1+3x)+(2+x)Y+(1+5x)Y^2 = 1+3x+2x^2+x^3+x^4+5x^5$$

3. Rational function reconstruction:

$$\frac{5x^2+6x+3}{x^2+4x+3} \equiv 1+3x+2x^2+x^3+x^4+5x^5 \pmod{x^5}$$

Nonsingular rational system solving over $\mathsf{K}[x]$ via lifting

Input: $A \in \mathsf{K}[x]^{n\times n}$ and $b \in \mathsf{K}[x]^{n\times 1}$ of degree d

Compute: $A^{-1}b \in \mathsf{K}(x)$

Method:

1. Choose $Y \in \mathsf{K}[x]$ such that $\gcd(Y, \det A) = 1$
   Set $k = \lceil 2nd / \deg Y \rceil$
2. Compute $B = \mathrm{Rem}(A^{-1}, Y)$
3. $r := b$
   for $i = 0$ **to** $k-1$ **do**
   $v_i := \mathrm{Rem}(Br, Y)$ # $\deg v_i < \deg Y$
   $r := (r - Av_i)/Y$ # $\deg r < d$
4. Reconstruct $A^{-1}b$ from $A^{-1}b \equiv v_0 + v_1 Y + \cdots + v_{k-1}Y^{k-1} \bmod Y^k$

$\rightarrow$ what should degree of Y be?

$\rightarrow$ what should factorization of Y be?

Lifting using a “lifting basis”

1. Select of modulus:
   $Y = (x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_{\deg Y})$
   $Z = (x-\beta_1)(x-\beta_2)\cdots(x-\beta_d)$ with $\gcd(Z,Y)=1$

2. Initialization:
   old: $B := \mathrm{Rem}(A^{-1}, Y)$
   new: $(B_1,\ldots,B_{\deg Y}) := ((A\,|_{x=\alpha_1})^{-1},\ldots,(A\,|_{x=\alpha_{\deg Y}})^{-1})$
   $(C_1,\ldots,C_d) := (A\,|_{x=\beta_1},\ldots,A\,|_{x=\beta_d})$

3(a). Lifting step:
   old: $v_i := \mathrm{Rem}(Br, Y)$
   new: $((v_i)\,|_{x=\alpha_1},\ldots,(v_i)\,|_{x=\alpha_{\deg Y}}) := (B_1 r\,|_{x=\alpha_1},\ldots,B_{\deg Y} r_{x=\alpha_{\deg Y}})$

3(b). Residue update:
   old: $r := (r - Av_i)/Y$
   new: $r\,|_{x=\beta_j} := (r\,|_{x=\beta_j} - C_j(v_i)\,|_{x=\beta_j})\mathrm{Rem}(Y^{-1}, x-\beta_j)$ for $j=1,\ldots,d$

Lifting using a "lifting basis": main work

2. Initialization:
   $\to$ $\deg Y$ matrix inversions over K

3(a). Lifting steps:
   $\to$ total $2nd$ matrix$\times$vector products over K

3(b). Residue updates:
   $\to$ total $2nd \times (d/\deg Y)$ matrix$\times$vector products over K

4. Reconstruct solution using interpolation $+$ radix conversion $+$ rational function reconstruction

Key optimizations:

- Choose $\deg Y$ to balance costs of phases 2 and 3.
  $\to$ can be automatically tuned
- Reduce the $2nd$ using vector rational function reconstruction.
  $\to$ decrease $2nd$ but increase cost of step 4
  [Olesh & Storjohann, 2007]

Linearization for polynomial lattice basis reduction

First consider Euclidean algorithm over $\mathbb{Z}/(7)[x]$

$$\begin{bmatrix} 4x^3+6x^2+5x+6 \\ 4x^3+2x^2+3x+5 \end{bmatrix} \to \begin{bmatrix} 4x^3+6x^2+5x+6 \\ 3x^2+5x+6 \end{bmatrix} \to \begin{bmatrix} 4x^2+4x+6 \\ 3x^2+5x+6 \end{bmatrix} \to \cdots \to \begin{bmatrix} x+6 \\ 0 \end{bmatrix}$$

Same idea works for lattice reduction of matrices
(Note: $[d] \equiv$a polynomial of degree d)

$$\overset{A}{\begin{bmatrix} [13] & [13] & [12] & [12] \\ [13] & [13] & [12] & [12] \\ [13] & [13] & [13] & [12] \\ [13] & [12] & [12] & [12] \end{bmatrix}} \to \begin{bmatrix} [13] & [13] & [12] & [12] \\ [13] & [12] & [12] & [12] \\ [13] & [13] & [13] & [12] \\ [13] & [12] & [12] & [12] \end{bmatrix} \to \cdots \to \overset{R}{\begin{bmatrix} [1] & [1] & [1] & [1] \\ [2] & [1] & [1] & [0] \\ [1] & [2] & [2] & [0] \\ [1] & [4] & [1] & [0] \end{bmatrix}}$$

<u>Question:</u> How to represent rows of the work matrix?

Linearization for lattice reduction of matrices

First consider Euclidean algorithm over $\mathbb{Z}/(7)[x]$

$$\begin{bmatrix} 4x^3+6x^2+5x+6 \\ 4x^3+2x^2+3x+5 \end{bmatrix} \to \begin{bmatrix} 4x^3+6x^2+5x+6 \\ 3x^2+5x+6 \end{bmatrix} \to \begin{bmatrix} 4x^2+4x+6 \\ 3x^2+5x+6 \end{bmatrix} \to \cdots \to \begin{bmatrix} x+6 \\ 0 \end{bmatrix}$$

Same idea works for lattice reduction of matrices
(Note: $[d]$ $\equiv$a polynomial of degree d)

$$\overset{A}{\begin{bmatrix} [13] & [13] & [12] & [12] \\ [13] & [13] & [12] & [12] \\ [13] & [13] & [13] & [12] \\ [13] & [12] & [12] & [12] \end{bmatrix}} \to \begin{bmatrix} [13] & [13] & [12] & [12] \\ [13] & [12] & [12] & [12] \\ [13] & [13] & [13] & [12] \\ [13] & [12] & [12] & [12] \end{bmatrix} \to \cdots \to \overset{R}{\begin{bmatrix} [1] & [1] & [1] & [1] \\ [2] & [1] & [1] & [0] \\ [1] & [2] & [2] & [0] \\ [1] & [4] & [1] & [0] \end{bmatrix}}$$

<u>Question:</u> How to represent rows of the work matrix?
<u>Idea:</u> also used in earlier version of FLINT

$$\left[\begin{array}{c|c} 3x^2+5x+1 & 4x+2 \\ \hline 4x^2+2 & 6x+2 \end{array}\right] \to \left[\begin{array}{ccc|ccc} 3 & 5 & 1 & 0 & 4 & 2 \\ \hline 4 & 0 & 2 & 0 & 6 & 2 \end{array}\right]$$

From left equivalence to similarity

Recall companion matrix: 1×1 matrix of degree 4

$$\left[x^4+5x^3+6x^2+74x+72\right] \longleftrightarrow xI_4 - \begin{bmatrix} & & & -72 \\ 1 & & & -74 \\ & 1 & & -6 \\ & & 1 & -5 \end{bmatrix}$$

Same idea works for matrices (sometimes): 2×2 matrix of degree 3

$$\overset{A}{\left[\begin{array}{c|c} x^3+2x^2+6x+6 & 4x^2+4x \\ \hline 2x^2+5x+3 & x^3+5x+4 \end{array}\right]} \longleftrightarrow xI_6 - \overset{C}{\left[\begin{array}{cc|cc|cc} & & & & -6 & 0 \\ & & & & -3 & -4 \\ \hline 1 & & & & -6 & -4 \\ & 1 & & & -5 & -5 \\ \hline & & 1 & & -2 & -4 \\ & & & 1 & -2 & 0 \end{array}\right]}$$

<u>Idea:</u> Compute $\det A$ by computing Frobenius form of C in time $O((nd)^3)$

From left equivalence to similarity: general case

Input:

$$A = \begin{bmatrix} 5x^2+4x+1 & x+1 \\ 5x+1 & 2x+1 \end{bmatrix}$$

1. Random shift:

$$B = A\mid_{x=x-2} = \begin{bmatrix} 5x^2+5x+6 & x+6 \\ 5x+5 & 2x+4 \end{bmatrix}$$

2. Revert:

$$C = x^2 B\mid_{x=1/x} = \begin{bmatrix} 6x^2+5x+5 & 6x^2+x \\ 5x^2+5x & 4x^2+2x \end{bmatrix}$$

3. Normalize:

$$D = \begin{bmatrix} 6 & 6 \\ 5 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 6x^2+5x+5 & 6x^2+x \\ 5x^2+5x & 4x^2+2x \end{bmatrix} = \begin{bmatrix} x^2+4x+6 & 6x \\ 5x+3 & x^2 \end{bmatrix}$$

Partial linearization

- Cost of many algorithm highly sensitive to $\deg A$
- What if some entries in A have large degree?
- Examples: $[t] \equiv$ a polynomial of degree t

$$A = \begin{bmatrix} [0] & & & [5] & [18] \\ & [0] & & [5] & [18] \\ & & [0] & [5] & [18] \\ & & & [6] & [18] \\ & & & & [19] \end{bmatrix} \qquad A = \begin{bmatrix} [19] & [1] & [5] & [3] & [19] \\ [4] & [6] & [3] & [6] & [0] \\ [0] & [0] & [0] & [0] & [0] \\ [17] & [6] & [0] & [0] & [0] \\ [19] & [0] & [0] & [0] & [0] \end{bmatrix}$$

Partial column linearization

$$
\begin{array}{c}
A \\
\left[\begin{array}{ccccc}
[0] & & & [5] & [18] \\
 & [0] & & [5] & [18] \\
 & & [0] & [5] & [18] \\
 & & & [6] & [18] \\
 & & & & [19]
\end{array}\right]
\end{array}
\longleftrightarrow
\begin{array}{c}
C \\
\left[\begin{array}{ccccc||c|ccc}
[0] & & & [4] & [4] & [0] & [4] & [4] & [3] \\
 & [0] & & [4] & [4] & [0] & [4] & [4] & [3] \\
 & & [0] & [4] & [4] & [0] & [4] & [4] & [3] \\
 & & & [4] & [4] & [1] & [4] & [4] & [3] \\
 & & & & [4] & & [4] & [4] & [4] \\
\hline\hline
 & & & -x^5 & & 1 & & & \\
 & & & & -x^5 & & 1 & & \\
 & & & & & & -x^5 & 1 & \\
 & & & & & & & -x^5 & 1
\end{array}\right]
\end{array}.
$$

- $\deg C =$ average column degree of A
- dimension of C is less than $2\times$ dimension of A
- no computation required, only rewriting
- $\det C = \det A$, $\text{nullity}(C) = \text{nullity}(A)$, $C^{-1} = \left[\begin{array}{c|c} A^{-1} & * \\ \hline * & * \end{array}\right]$

Partial row and column linearization

$$A = \begin{bmatrix} [19] & [1] & [5] & [3] & [19] \\ [4] & [6] & [3] & [6] & [0] \\ [0] & [0] & [0] & [0] & [0] \\ [17] & [6] & [0] & [0] & [0] \\ [19] & [0] & [0] & [0] & [0] \end{bmatrix}$$

$$C = \left[\begin{array}{ccccccccc||cccc}
[4] & [1] & [4] & [3] & [4] & [4] & [4] & [4] & & -x^5 & & & \\
[4] & [4] & [3] & [4] & [0] & & & & [1] & & & & -x^5 \\
[0] & [0] & [0] & [0] & [0] & & & & & & & & \\
[4] & [4] & [0] & [0] & [0] & [4] & [4] & [2] & [1] & & & & \\
[4] & [0] & [0] & [0] & [0] & [4] & [4] & [4] & & & & & \\
-x^5 & & & & & 1 & & & & & & & \\
& & & & & -x^5 & 1 & & & & & & \\
& & & & & & -x^5 & 1 & & & & & \\
& -x^5 & & & & & & & 1 & & & & \\
\hline\hline
& & [0] & & [4] & & & & & 1 & -x^5 & & \\
& & & & [4] & & & & & & 1 & -x^5 & \\
& & & & [4] & & & & & & & 1 & \\
\hline
& & & [1] & & & & & & & & & 1
\end{array}\right].$$

- dimension of C is $< 3 \times (19+6+0+0+0)/5 = 15$