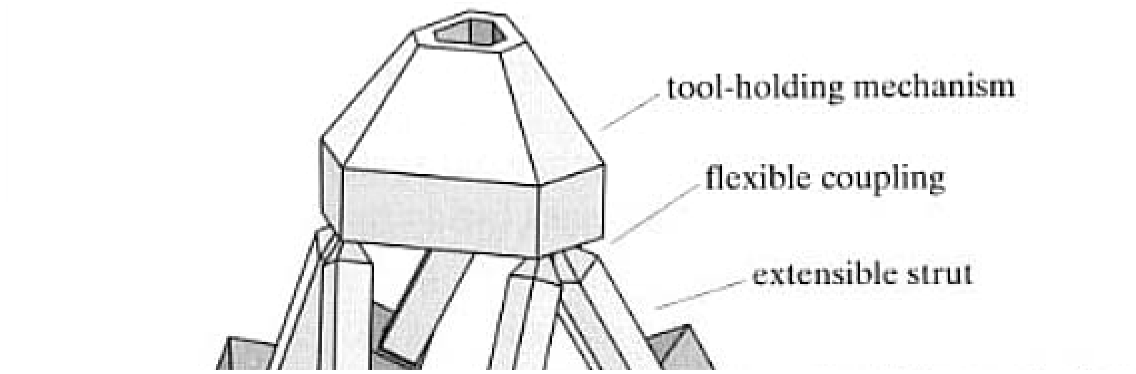
Extracting Sparse Resultant Matrices from Dixon Resultant Formulation

Arthur D. Chtcherba and Deepak Kapur

University of New Mexico
Albuquerque, NM 87131, USA

Stewart Platform Problem



$$f_1 = x^T x - \alpha_1 |q|$$

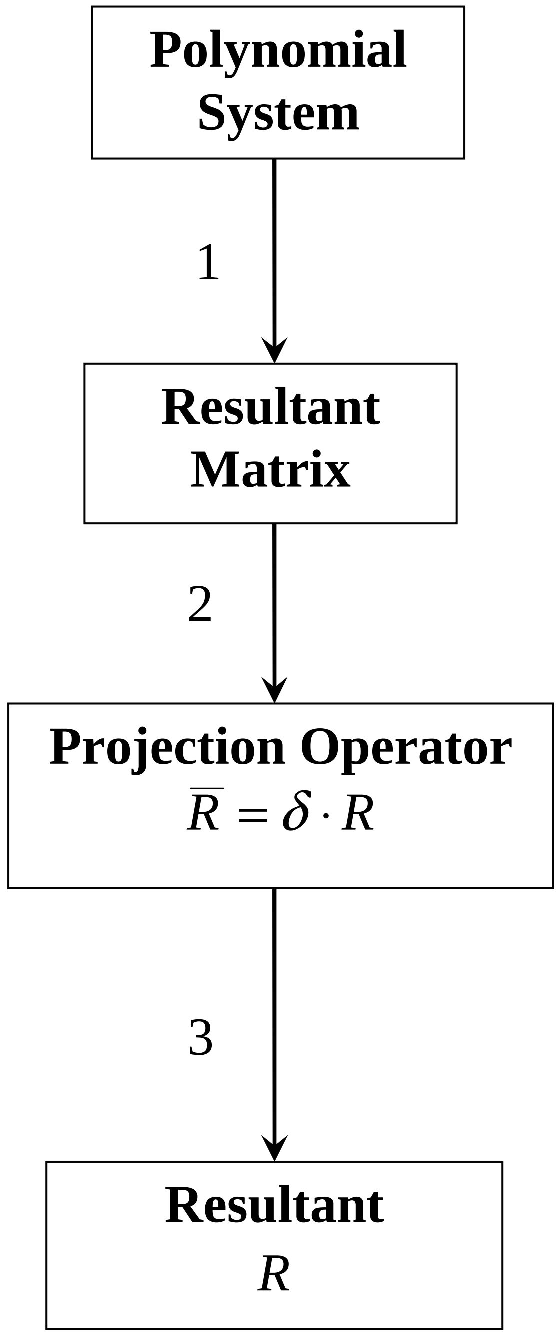
$$f_i = b_i^T xq - a_i^T qx - \left(qb_i q^*\right)^T a_i - \alpha_i |\, q \,| \qquad i = 2,\ldots,6$$

$$f_7 = x_0 - x_1 q_1 - x_2 q_2 - x_3 q_3$$

$x = (x_0, x_1, x_2, x_3)$ $\qquad$ $q = (1, q_1, q_2, q_3)$

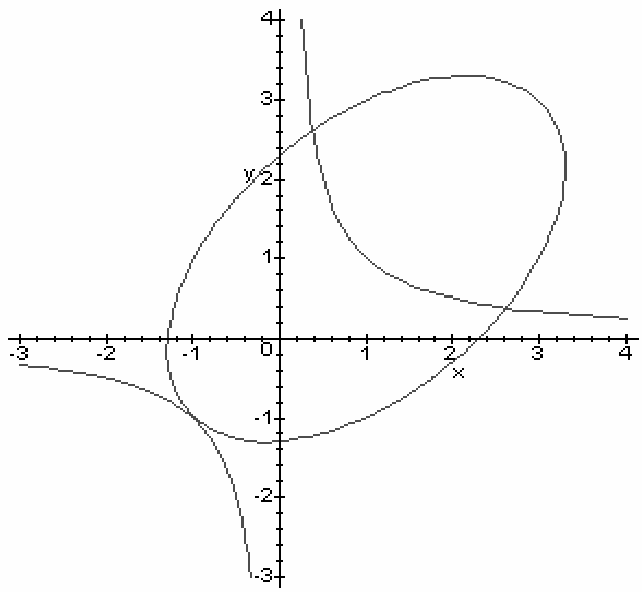
Hidden Variable	Resultant Degree	Dixon Projection	Dixon Sparse Projection
q_1	40	103	?
x_3	40	40	?

Computing Resultant



Resultants and solutions of a Polynomial system (Example)

$$f := x^2 + y^2 - xy - x - y - 3$$
$$g := xy - 1$$



Sylvester Resultant

$M := \text{sylvester}(f, g, x);$

$$M = \begin{bmatrix} 1 & -1-y & y^2 - y - 3 \\ y & -1 & 0 \\ 0 & y & -1 \end{bmatrix}$$

$\text{factor}(\det M);$

$$\left(2y - 3 - \sqrt{5}\right)\left(2y - 3 + \sqrt{5}\right)(1 + y)^2$$

Solutions: (Solve for *y*, and then extract solutions for *x*)

1. $y = -1$ $\qquad x = -1$
2. $y = \dfrac{3 \pm \sqrt{5}}{2}$ $\qquad x = \dfrac{3 \mp \sqrt{5}}{2}$

Dixon Resultant Formulation

$M := \text{DixonMatrix}\left(f,\ g,\ u_0 + u_1 x + u_2 y\right);;$

$$M = \begin{bmatrix} 0 & u_1 & u_2 & u_0 \\ -u_2 & 0 & -u_0 & -u_1 \\ -u_1 & u_0 & u_1 - u_2 & u_2 + 4u_1 - u_0 \\ -u_0 & u_2 & u_0 - u_1 - 4u_2 & u_1 - u_2 \end{bmatrix}$$

factor(det M);

$$\left(\left(3+\sqrt{5}\right)u_2 + \left(3-\sqrt{5}\right)u_1 + 2u_0\right)$$
$$\left(\left(3-\sqrt{5}\right)u_2 + \left(3+\sqrt{5}\right)u_1 + 2u_0\right)\left(u_2 + u_1 - u_0\right)^2$$

Solutions: (Coefficients of u_1, u_2)

- $x = \dfrac{3-\sqrt{5}}{2}$ $\qquad y = \dfrac{3+\sqrt{5}}{2}$
- $x = \dfrac{3+\sqrt{5}}{2}$ $\qquad y = \dfrac{3-\sqrt{5}}{2}$
- $x = -1$ $\qquad y = -1$

Condition that circle, hyperbola and line intersect at common point

$$f := (x-s)^2 + (y-t)^2 - r^2$$
$$g := q\,xy + p$$
$$h := a\,x + b\,y + c$$

Projection Operator

$q^2 (q^2 c^4 + b^4 p^2 + a^4 p^2 + 2\, q^2 b c^3 t + 2\, b^2 a^2 p^2 + 2\, q b^3 a s^2 p - 8\, q b^2 a^2 s\, t\, p + 2\, q\, b\, a^3 t^2 p + q^2 b^2 a^2 s^4 + 2\, q^2 b^2 a^2 s^2 t^2 - 2\, q^2 b^2 a^2 s^2 r^2 + 2\, q^2 b\, a^2 s^2 t\, c - 2\, q\, b\, a^3 s^2 p + q^2 b^2 a^2 t^4 - 2\, q^2 b^2 a^2 t^2 r^2 + 2\, q^2 b\, a^2 t^3 c + q^2 b^2 a^2 r^4 - 2\, q^2 b\, a^2 r^2 t\, c + 2\, q\, b\, a^3 r^2 p + 2\, q^2 b^2 a\, s^3 c + 2\, q^2 b^2 a\, s\, c\, t^2 - 2\, q^2 b^2 a\, s\, c\, r^2 + 4\, q^2 b\, a\, s\, c^2 t - 6\, q\, b\, a^2 s\, c\, p - 2\, q\, b^3 a\, p\, t^2 + 2\, q\, b^3 a\, p\, r^2 - 6\, q\, b^2 a\, p\, t\, c + 2\, q\, b^3 c\, s\, p + q^2 b^2 c^2 s^2 + q^2 b^2 c^2 t^2 - q^2 b^2 c^2 r^2 - 4\, q\, b\, c^2 p\, a + q^2 c^2 a^2 s^2 + q^2 c^2 a^2 t^2 - q^2 c^2 a^2 r^2 + 2\, q^2 c^3 a\, s + 2\, q\, c\, a^3 p\, t)$

Extraeneous factor

$$q^2$$

Constructing Resultant Matrix

Major Resultant Matrix Methologies

♦ **Macaulay** (Macaulay 1902)
Based on total degrees
Does not use sparsness of polynomial system
Matrices are big, and usually are singular
Projective Variety

♦ **Newton**
(Sturmfels, Zelevinski, Emiris, Canny, 1994)
Based on Mixed volumes of Newton Polytopes
Explicitly uses sparsness of polynomial system
Matrices are much smaller than Macaulay
Toric Variety

♦ **Dixon** (Dixon 1908), (Kapur and Saxena 1994)
Based on Bezout method for one variable
Implicitly uses sparsness
Small Resultant matrices,
(by a factor of *n* than Newton Sparse)
More complicated matrix entries
Affine Variety

Sparse Resultant Matrices

for every f_i compute set

$$X_i = \{\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_{m_i}}\}$$

then let

$$X_i f_i \equiv \{\mathbf{x}^{\alpha} f_i \mid \mathbf{x}^{\alpha} \in X_i\}$$

In matrix notation it can be written as

$$\begin{pmatrix} X_0 f_0 = \mathbf{M}_0 \times Y_0 \\ \vdots \\ X_i f_i = \mathbf{M}_i \times Y_i \\ \vdots \\ X_n f_n = \mathbf{M}_n \times Y_n \end{pmatrix} = \mathbf{M} \times Y$$

must have

$$Size(\mathbf{M}) \geq \deg R = \sum_{i=0}^{n} \#\text{ of roots of } F_i$$

Example:

$$f_0 = x^2 + y^2 - xy - x - y - 3$$
$$f_1 = xy - 1$$
$$f_2 = ax + by + c$$

Multiplier sets:

$$X_0 = \{1, y\} \qquad X_1 = \{1, x, y\} \qquad X_2 = \{1, x, y, y^2\}$$

Matrix:

	x^2y	x^2	xy^2	xy	x	y^3	y^2	y	1
f_0	0	**1**	0	**-1**	**-1**	0	**1**	**-1**	**-3**
$y f_0$	**1**	0	**-1**	**-1**	0	**1**	**-1**	**-3**	0
f_1	0	0	0	**1**	0	0	0	0	**-1**
$x f_1$	**1**	0	0	0	**-1**	0	0	0	0
$y f_1$	0	0	**1**	0	0	0	0	**-1**	0
f_2	0	0	0	0	***a***	0	0	***b***	***c***
$x f_2$	0	***a***	0	***b***	***c***	0	0	0	0
$y f_2$	0	0	0	***a***	0	0	***b***	***c***	0
$y^2 f_2$	0	0	***a***	0	0	***b***	***c***	0	0

Determinant (Projection operator):

$$\left(2c + \left(3 - \sqrt{5}\right)a + \left(3 + \sqrt{5}\right)b\right)\left(2c + \left(3 + \sqrt{5}\right)a + \left(3 - \sqrt{5}\right)b\right)$$
$$(c - a - b)^2$$

Existing Methods for computing Newton Matrices

Mixed Subdivision based (Canny & Emiris 1993)

- Compute mixed subdivision of Minskowski sum of Newton polytopes.
- Read off monomial multipliers from each subdivision.

Computation of mixed subdivision is #P-complete
Random Algorithm

Incremental (Emiris 1994)

- Choose random direction vector
- Pick closest monomials in that direction as multipliers

Random Algorithm (highly depends on random vector).
Uses Linear programming to find nearest monomials.

Main Result

- ♦ Kapur and Saxena have shown how **Dixon Resultant Formulation** *implicitly* exploits sparse structure of a given polynomial system.

- ♦ Existing Newton Sparse Matrix construction methods *have to know a priori number of roots* or at least an upper bound.

Is it possible to construct such matrices without that knowledge, as Dixon method does ?

The answer is **yes**.

The proposed algorithm is based on Dixon Construction is:

- ♦ Simple
- ♦ Does not need to know number of roots
- ♦ Faster/more efficient than subdivision and incremental
- ♦ Better at producing smaller matrices

Dixon Resultant Formulation

Dixon polynomial

$$\theta(f_0, f_1, \ldots, f_n) = \prod_{i=1}^{n} \frac{1}{\bar{x}_i - x_i} \begin{vmatrix} f_0(x_1,\ldots,x_n) & f_1(x_1,\ldots,x_n) & \cdots & f_n(x_1,\ldots,x_n) \\ f_0(\bar{x}_1,\ldots,x_n) & f_1(\bar{x}_1,\ldots,x_n) & \cdots & f_n(\bar{x}_1,\ldots,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_0(\bar{x}_1,\ldots,\bar{x}_n) & f_1(\bar{x}_1,\ldots,\bar{x}_n) & \cdots & f_n(\bar{x}_1,\ldots,x_n) \end{vmatrix}$$

Dixon Matrix

$$\theta = \sum_{\alpha,\beta} p_{\alpha,\beta} \mathbf{x}^{\alpha} \bar{\mathbf{x}}^{\beta}$$

$$= (\bar{\mathbf{x}}^{\beta_1} \ldots \bar{\mathbf{x}}^{\beta_k}) \begin{vmatrix} p_{\alpha_1\beta_1} & p_{\alpha_2\beta_1} & \cdots & p_{\alpha_l\beta_1} \\ p_{\alpha_1\beta_2} & p_{\alpha_2\beta_2} & \cdots & p_{\alpha_l\beta_2} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\alpha_1\beta_k} & p_{\alpha_2\beta_k} & \cdots & p_{\alpha_l\beta_k} \end{vmatrix} \begin{pmatrix} \mathbf{x}^{\alpha_1} \\ \mathbf{x}^{\alpha_2} \\ \vdots \\ \mathbf{x}^{\alpha_l} \end{pmatrix} =$$

$$= \bar{X}\,\Theta\,X$$

Since whenever $F \equiv 0$ has a solution, $\theta = 0$ no matter what $\bar{x}_1,\ldots,\bar{x}_n$ are, it follows that

$$\Theta\,X = 0$$

Derivation of Dixon Sparse Matrix

$$\theta(f_0, \ldots, f_n) = \sum_{i=1}^{n} \theta_i f_i$$

where $\theta_i = \theta(f_0, \ldots, f_{i-1}, 1, f_{i+1}, \ldots, f_n)$.

Let

$$X_i := \{\mathbf{x}^\alpha \mid \mathbf{x}^\alpha \in \theta_i\}$$

Then construct sparse Dixon matrix using X_i.

$$\mathbf{M} \times Y = \begin{pmatrix} X_0 f_0 \\ \vdots \\ X_i f_i \\ \vdots \\ X_n f_n \end{pmatrix}$$

It can be established that

$$\Theta = \mathbf{T} \times \mathbf{M}$$

hence $\mathbf{M}$ retains properties of the **resultant** matrix.

Complexity

To construct Sparse matrix one needs to compute for each f_i a set of monomial multipliers X_i

For Dixon Sparse matrix it means to compute

$$\theta_i = \prod_{i=1}^{n} \frac{1}{\bar{x}_i - x_i} \left| f_0, \cdots, f_{i-1}, 1, f_{i+1}, \cdots, f_n \right|$$

Assuming number of terms in $F \leq m$
this amounts to expanding $\binom{m}{n}$ determinants of size $n \times n$.

Hence total complexity is

$$T_A = n \binom{m}{n} n! = \frac{n\, m!}{(m-n)!} = O\left(n\, m^n\right)$$

i.e. construction time is polynomial in number of terms for fixed dimension.

Complexity

Subdivision method complexity is given by

$$T_S = O\left(Size(M) d^{9.5} m^{6.5} \log^2 k \log^2 \frac{1}{\varepsilon_l \varepsilon_\delta} \right)$$

where

$Size(M)$ - size of resultant matrix

k - max degree of polynomials in the

ε_l - probability of failure to pick generic lifting vector

ε_δ - probability of perturbation failure

Incremental method complexity is given by

$$T_I = O^*\left(e^{3d} m^{5.5} (\deg R)^3 \right) + O^*\left(d^{7.5} m^{2d+5.5} \right)$$

Benchmark Examples:

#	Problem	*d*	Bezout Bound	Mixed Volume	deg R_T	Mixed Vol	deg R_A
1	Emiris Example	2	8,6,12	4,3,4	11	4,3,4	11
2	Cyclic Roots n = 3	2	4,2,2	2,2,2	6	2,2,2	6
3	Cyclic Roots n = 4	3	18,9,6,6	5,6,5,4	20	5,6,5,5	21
4	Cyclic Roots n = 5	4	96,48,32, 24,24	16,18,18, 16,14	82	16,18,18, 16,18	86
5	Cyclic Roots n = 6	5	600,300,200, 150,120,120	46,58,56, 58,46,26	290	46,58,56, 58,46,52	316
6	Sum of Squares n = 2	2	2,2,4	2,2,4	8	2,2,4	8
7	Sum of Squares n = 3	3	4,4,4,8	4,4,4,8	20	4,4,4,8	20
8	Sum of Squares n = 4	4	8,8,8,8,16	8,8,8,8,16	48	8,8,8,8,16	48
9	Sum of Squares n = 5	5	16,16,16, 16,16,32	16,16,16, 16,16,32	112	16,16,16, 16,16,32	112
10	Sum of Squares n = 6	6	32,32,32,32, 32,32,64	32,32,32,32, 32,32,64	256	32,32,32,32, 32,32,64	256
11	Max Volume of Tetrahedron	4	24,24,24,24,16	9,9,9,9,8	44	9,9,9,9,8	44
12	Distance of Intersection of conics from origin	2	4,4,4	4,4,4	12	4,4,4	12
13	Condition of perpendicular intersection of conic and a circle	2	4,4,4	4,4,4	12	4,4,4	12
14	Imlicitization of a sphere	2	16,8,8	8,4,4	16	12,4,4	20
15	Implicitization of bicubic surface	2	18,18,9	18,18,9	45	18,18,9	45
16	Implicitization of bicubic dense surface	2	36,36,36	18,18,18	54	18,18,18	54
17	Cubic surface implicitization	2	9,9,9	9,9,9	27	9,9,9	27
18	Equilibrium of Lerentz System	3	4,4,8,4	3,4,5,3	15	3,4,5,3	15
19	Camera Motion from point matches	5	32,32,32, 32,32,32	6,6,6,6,6,6	36	6,6,6,6,6,6	36
20	Conformal Analysis of cyclic molecules	3	16,16,16,64	12,12,12,16	52	12,12,12,16	52
21	Stewart Platform q1 is parameter	6	64,64,64,64, 64,64,64	22,32,32,32, 32,32,32	214	22,32,32,32, 32,32,32	214
22	Stewart Platform x3 is parameter	6	64,64,64,64, 64,64,64	42,52,52,52, 52,52,52	354	42,52,52,52, 52,52,52	354

Empirical Results

#	Subdivision Matrix				Incremental Matrix				Sparse Dixon Matrix			
	Size	Rank	# Mult	Time	Size	Rank	# Mult	Time	Size	Rank	# Mult	Time
1	14 × 14	14	4,4,6	10.29	12×12	12	5,3,4	12.41	12 × 12	12	5,3,4	0.08
2	6 × 6	6	2,2,2	3.18	8 × 7	7	3,2,3	4.04	6 × 6	6	2,2,2	0.05
3	25 × 25	25	5,7,8,5	10.90					20 × 22	18	4,6,4,6	0.01
4	144 × 144	144	15,26,33, 32,38	187.35					90 × 92	90	16,18,18, 16,22	0.04
5									408 × 412	402	57,68,72, 68,57,86	0.75
6	9 × 9	9	2,3,4	3.39	8 × 8	8	2,2,4	6.61	8 × 8	8	2,2,4	0.03
7	20 × 20	20	4,4,4,8	12.08	20×20	20	4,4,48	72.13	20 × 20	20	4,4,4,8	0.06
8	48 × 48	48	8,8,8,8,16	35.13	48×48	48	8,8,8,8,1 6	762.23	48 × 48	48	8,8,8,8,16	0.18
9	136×136	136	16,16,16, 40,16,32	103.08					112 × 112	112	16,16,16,16,16, 32	0.08
10	280×280	280	32,32,48,40, 32,32,64	577.11					256 × 256	256	32,32,32,32,32, 32,64	0.39
11	102 × 102	102	9,18,20, 26,29	166.64					60 × 63	60	11,11,12,16,10	0.42
12	14 × 14	14	4,5,5	7.95	16×15	15	6,6,4	17.06	15 × 14	14	5,5,5	0.04
13	21 × 21	21	4,7,10	14.41	16×15	15	6,6,4	14.06	13 × 14	12	5,3,5	0.04
14	24 × 24	24	8,8,8	14.88	16×16	16	8,4,4	20.38	20 × 20	20	12,4,4	0.16
15	64 × 64	64	18,25,21	78.93	110×77	77	41,41,28	74.89	45 × 45	45	18,18,9	0.28
16	72 × 72	72	18,27,27	154.24					54 × 54	54	18,18,18	0.75
17	38 × 38	38	9,15,14	26.53	39×33	33	12,15,12	27.76	28 × 28	28	9,10,9	0.09
18	23 × 23	23	3,5,7,8	19.98	19×17	17	4,4,6,5	134.62	16 × 16	16	3,5,5,3	0.07
19									36 × 36	36	6,6,6,6,6,6	0.92
20	108×108	108	12,16,24,56	112.01					86 × 84	81	15,21,18,32	0.07
21									659 × 454	437	27,105,105,105, 105,105,107	9.54
22									1328×924		74,207,207,207, 207,207,219	25.75

Univariate Case Example

$$f_0 = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0$$
$$f_1 = b_n x^n + b_{n-1} x^{n-1} + \cdots b_1 x + b_0$$

Classical Sylvester matrix

$$R = \begin{matrix} x^{n-1} f_0 \\ x^{n-2} f_0 \\ \vdots \\ f_0 \\ x^{m-1} f_1 \\ x^{m-2} f_1 \\ \vdots \\ f_0 \end{matrix} \begin{vmatrix} a_m & a_{m-1} & \cdots & a_0 & & & \\ & a_m & a_{m-1} & \cdots & a_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & a_m & a_{m-1} & \cdots & a_0 \\ b_n & b_{n-1} & \cdots & b_0 & & & \\ & b_n & b_{n-1} & \cdots & b_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & b_n & b_{n-1} & \cdots & b_0 \end{vmatrix}$$

Note

$$\theta_0 = \theta(1, f_1) = \frac{1}{\bar{x} - x} \begin{vmatrix} 1 & f_1(x) \\ 1 & f_1(\bar{x}) \end{vmatrix} = \frac{f_1(\bar{x}) - f(x)}{\bar{x} - x} =$$

$$= b_1 + b_2 \frac{\bar{x}^2 - x^2}{\bar{x} - x} + \cdots + b_n \frac{\bar{x}^n - x^n}{\bar{x} - x}$$

i.e. $X_0 = \{1, x, \ldots, x^{n-1}\}$, similarly $X_1 = \{1, x, \ldots, x^{m-1}\}$

Transformation matrix 1D case

$$T = \begin{pmatrix} & & & & & & & & & & a_m \\ & & & b_n & & & & & & a_m & a_{m-1} \\ & & b_n & b_{n-1} & & & & & \iddots & \vdots & \vdots \\ & \iddots & \vdots & \vdots & & & & a_m & \cdots & a_4 & a_3 \\ & b_n & \cdots & b_3 & b_2 & & a_m & a_{m-1} & \cdots & a_3 & a_2 \\ b_n & b_{n-1} & \cdots & b_2 & b_1 & a_m & a_{m-1} & a_{m-2} & \cdots & a_2 & a_1 \end{pmatrix}$$

Matrix size is $\max\{m,n\}\times(m+n)$

Assuming $m>n$, note that determinant of any minor of T will have factor of a_m^{m-n}, and hence Bezout matrix

$$\Theta = T \times M$$

will have extra factor of a_m^{m-n}.

Conclusion

- Simple way to construct Sparse matrices
- The method is efficient in theory and practice
- Implicitly exploits sparse structure
- Empirically have shown to construct smaller matrices than subdivision method
- It is believed that this technique to construct matrices together with its cousin Dixon Resultant Matrix will help to better identify extraneous factors.