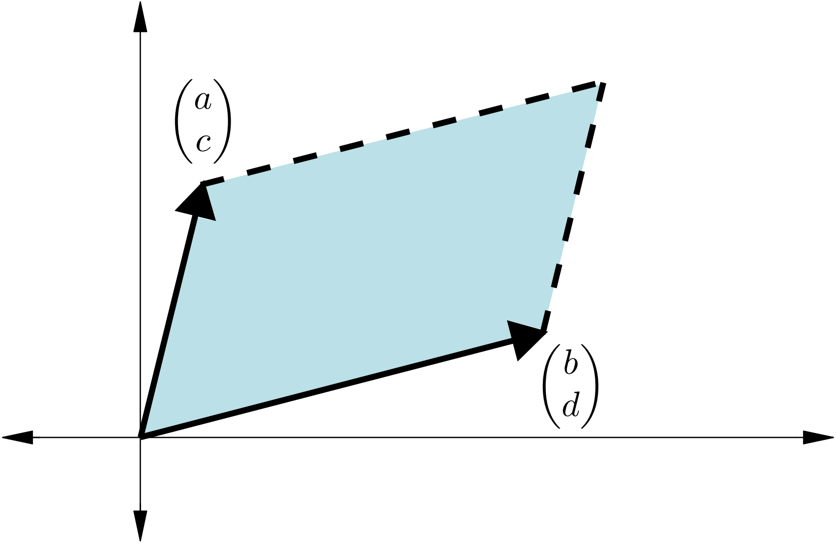
Efficient Calculation of Determinants of Symbolic Matrices with Many Variables

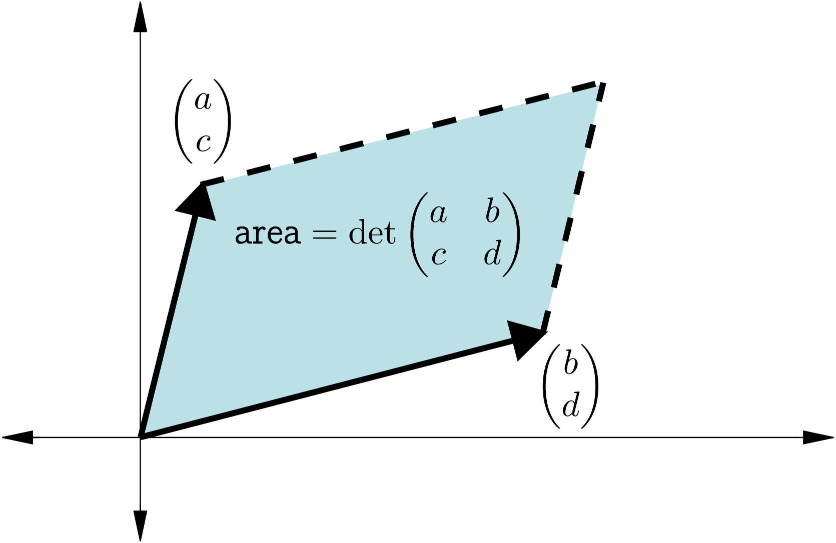
Ziv Scully

Second Annual MIT PRIMES Conference
May 20, 2012

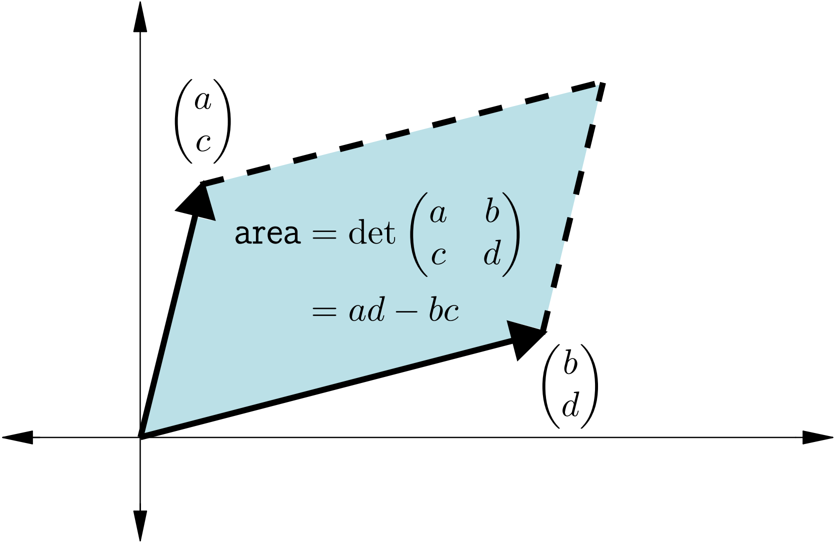
Vectors and Volumes



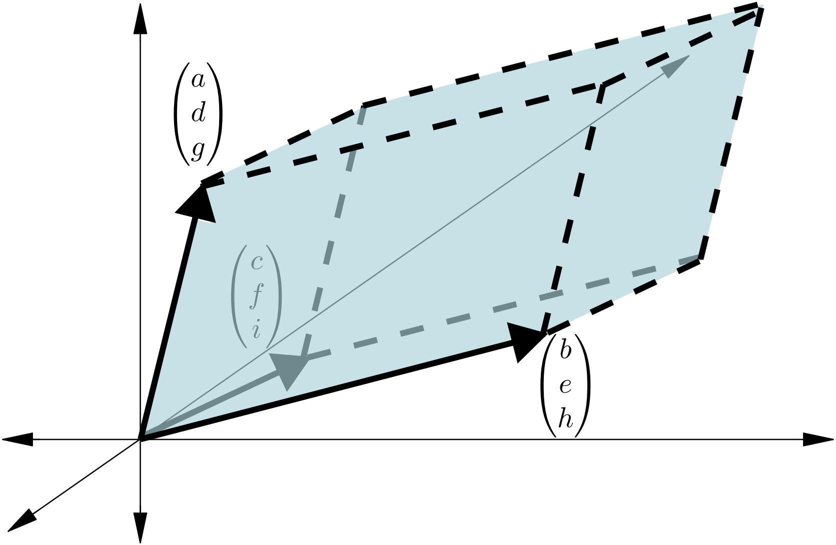
Vectors and Volumes



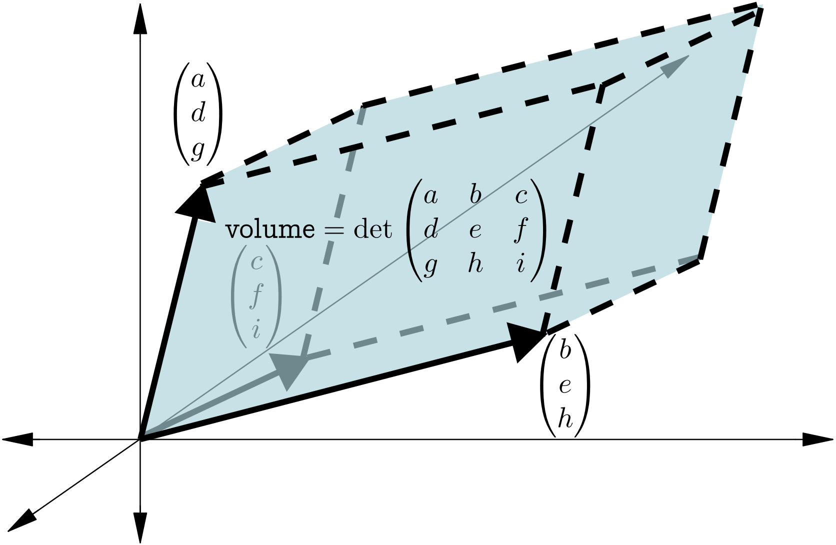
Vectors and Volumes



Vectors and Volumes



Vectors and Volumes



Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Determinants

2×2 matrices:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

3×3 matrices:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

$n \times n$ matrices:

$$\det(A) = \sum_{\sigma \in S_n} \left[\operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)} \right]$$

Motivation

- Linear systems.

Motivation

- Linear systems.
- Calculus.

Motivation

- Linear systems.
- Calculus.
- Control theory.

Motivation

- Linear systems.
- Calculus.
- Control theory.
- Engineering models.

Motivation

- Linear systems.
- Calculus.
- Control theory.
- Engineering models.
- Code generation.

Motivation

- Linear systems.
- Calculus.
- Control theory.
- Engineering models.
- Code generation.

We are interested in matrices with polynomial entries.

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} afkp - aflo - agjp + agln + ahjo - ahkn \\ - bekp + belo + bgip - bglm - bhio + bhkm \\ + cejp - celn - cfip + cflm + chin - chjm \\ - dejo + dekn + dfio - dfkm - dgin + dgjm \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(fkp - flo - gjp + gln + hjo - hkn) \\ -\, b(ekp - elo - gip + glm + hio - hkm) \\ +\, c(ejp - eln - fip + flm + hin - hjm) \\ -\, d(ejo - ekn - fio + fkm + gin - gjm) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(fkp - flo - gjp + gln + hjo - hkn) \\ -\, b(ekp - elo - gip + glm + hio - hkm) \\ +\, c(ejp - eln - fip + flm + hin - hjm) \\ -\, d(ejo - ekn - fio + fkm + gin - gjm) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(fkp - flo - gjp + gln + hjo - hkn) \\ -\, b(ekp - elo - gip + glm + hio - hkm) \\ +\, c(ejp - eln - fip + flm + hin - hjm) \\ -\, d(ejo - ekn - fio + fkm + gin - gjm) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(fkp - flo - gjp + gln + hjo - hkn) \\ - b(ekp - elo - gip + glm + hio - hkm) \\ + c(ejp - eln - fip + flm + hin - hjm) \\ - d(ejo - ekn - fio + fkm + gin - gjm) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(fkp - flo - gjp + gln + hjo - hkn) \\ - b(ekp - elo - gip + glm + hio - hkm) \\ + c(ejp - eln - fip + flm + hin - hjm) \\ - d(ejo - ekn - fio + fkm + gin - gjm) \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(fkp - flo - gjp + gln + hjo - hkn) \\ -\,b(ekp - elo - gip + glm + hio - hkm) \\ +\,c(ejp - eln - fip + flm + hin - hjm) \\ -\,d(ejo - ekn - fio + fkm + gin - gjm) \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -\,b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +\,c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -\,d(e(jo-kn)-f(io-km)+g(in-jm)) \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -\,b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +\,c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -\,d(e(jo-kn)-f(io-km)+g(in-jm)) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -d(e(jo-kn)-f(io-km)+g(in-jm)) \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -\,b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +\,c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -\,d(e(jo-kn)-f(io-km)+g(in-jm)) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -d(e(jo-kn)-f(io-km)+g(in-jm)) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{array}{l} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -\,b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +\,c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -\,d(e(jo-kn)-f(io-km)+g(in-jm)) \end{array}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -\,b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +\,c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -\,d(e(jo-kn)-f(io-km)+g(in-jm)) \end{matrix}$$

Minor Expansion

Naive calculation requires $\Theta(n!\,n)$ polynomial multiplications.

$$\det\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{matrix} a(f(kp-lo)-g(jp-ln)+h(jo-kn)) \\ -b(e(kp-lo)-g(ip-lm)+h(io-km)) \\ +c(e(jp-ln)-f(ip-lm)+h(in-jm)) \\ -d(e(jo-kn)-f(io-km)+g(in-jm)) \end{matrix}$$

Minor expansion requires $\displaystyle\sum_{i=2}^{n} i\binom{n}{i} \in \Theta(2^n n)$ polynomial multiplications.

Gaussian Elimination

$$\det \begin{pmatrix} a & b & c & d \\ 0 & f & g & h \\ 0 & 0 & k & l \\ 0 & 0 & 0 & p \end{pmatrix} = afkp$$

Gaussian Elimination

$$\det \begin{pmatrix} a & b & c & d \\ 0 & f & g & h \\ 0 & 0 & k & l \\ 0 & 0 & 0 & p \end{pmatrix} = afkp$$

Gaussian Elimination

$$\det\begin{pmatrix} a & b & c & \dots \\ e & f & g & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} =$$

Gaussian Elimination

$$\det \begin{pmatrix} a & b & c & \dots \\ e & f & g & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \det \begin{pmatrix} a & b & c & \dots \\ e - \frac{e}{a}a & f - \frac{e}{a}b & g - \frac{e}{a}c & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Gaussian Elimination

$$\det\begin{pmatrix} a & b & c & \dots \\ e & f & g & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \det\begin{pmatrix} a & b & c & \dots \\ 0 & f - \frac{e}{a}b & g - \frac{e}{a}c & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Gaussian Elimination

$$\det\begin{pmatrix} a & b & c & \dots \\ e & f & g & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \det\begin{pmatrix} a & b & c & \dots \\ 0 & f - \frac{e}{a}b & g - \frac{e}{a}c & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$A^{(1)} = A,$$

$$A_{i,j}^{(k+1)} = A_{i,j}^{(k)} - \frac{A_{i,k}^{(k)}}{A_{k,k}^{(k)}} A_{k,j}^{(k)}$$

Gaussian Elimination

$$\det\begin{pmatrix} a & b & c & \dots \\ e & f & g & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \det\begin{pmatrix} a & b & c & \dots \\ 0 & f - \frac{e}{a}b & g - \frac{e}{a}c & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$A^{(1)} = A,$$

$$A_{i,j}^{(k+1)} = A_{i,j}^{(k)} - \frac{A_{i,k}^{(k)}}{A_{k,k}^{(k)}} A_{k,j}^{(k)}$$

$$A^{(1)} = A, \quad A_{0,0}^{(0)} = 1,$$

$$A_{i,j}^{(k+1)} = \frac{A_{i,j}^{(k)} A_{k,k}^{(k)} - A_{i,k}^{(k)} A_{k,j}^{(k)}}{A_{k-1,k-1}^{(k-1)}}$$

Gaussian Elimination

$$\det\begin{pmatrix} a & b & c & \cdots \\ e & f & g & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \det\begin{pmatrix} a & b & c & \cdots \\ 0 & f - \frac{e}{a}b & g - \frac{e}{a}c & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$A^{(1)} = A,$$

$$A_{i,j}^{(k+1)} = A_{i,j}^{(k)} - \frac{A_{i,k}^{(k)}}{A_{k,k}^{(k)}} A_{k,j}^{(k)}$$

$$A^{(1)} = A, \quad A_{0,0}^{(0)} = 1,$$

$$A_{i,j}^{(k+1)} = \frac{A_{i,j}^{(k)} A_{k,k}^{(k)} - A_{i,k}^{(k)} A_{k,j}^{(k)}}{A_{k-1,k-1}^{(k-1)}}$$

Fraction-free Gaussian elimination requires $\sum_{i=1}^{n} \Theta(i^2) \in \Theta(n^3)$ polynomial multiplications and divisions.

Comparison

Preservation of “simple” polynomials (e.g., those with few terms):

Comparison

Preservation of “simple” polynomials (e.g., those with few terms):

- Minor expansion: Entries are preserved.

Comparison

Preservation of “simple” polynomials (e.g., those with few terms):

- Minor expansion: Entries are preserved.
- Gaussian elimination: Entries made more complicated each step.

Comparison

Preservation of “simple” polynomials (e.g., those with few terms):

- Minor expansion: Entries are preserved.
- Gaussian elimination: Entries made more complicated each step.

Consider an $n \times n$ matrix with entries of the form $\sum_{i=1}^{s} a_i x_i$:

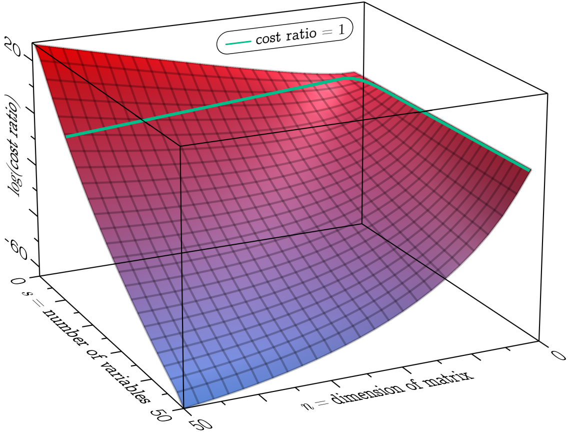
Comparison

Preservation of "simple" polynomials (e.g., those with few terms):

- Minor expansion: Entries are preserved.
- Gaussian elimination: Entries made more complicated each step.

Consider an $n \times n$ matrix with entries of the form $\sum_{i=1}^{s} a_i x_i$:

$$\text{cost ratio} = \frac{\text{cost of ME}}{\text{cost of FFGE}}$$



Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

- $(pq)r$ requires $ab + abc$ integer multiplications.

Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

- $(pq)r$ requires $ab + abc$ integer multiplications.
- $(pr)q$ requires $ac + abc$ integer multiplications.

Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

- $(pq)r$ requires $ab + abc$ integer multiplications.
- $(pr)q$ requires $ac + abc$ integer multiplications.
- $(qr)p$ requires $bc + abc$ integer multiplications.

Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

- $(pq)r$ requires $ab + abc$ integer multiplications.
- $(pr)q$ requires $ac + abc$ integer multiplications.
- $(qr)p$ requires $bc + abc$ integer multiplications.

We want to defer multiplying by polynomials with many terms.

Row Permutation

Let p, q and r be polynomials with a, b and c terms, respectively, with no variables in common.

- $(pq)r$ requires $ab + abc$ integer multiplications.
- $(pr)q$ requires $ac + abc$ integer multiplications.
- $(qr)p$ requires $bc + abc$ integer multiplications.

We want to defer multiplying by polynomials with many terms.

- Absolute value of determinant is invariant under row swaps.

Experiment Setup

Random polynomial matrices:

Experiment Setup

Random polynomial matrices:

- 9×9.

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

Sort rows r in ascending order based on these scores:

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

Sort rows r in ascending order based on these scores:

- Control, no sorting.

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

Sort rows r in ascending order based on these scores:

- Control, no sorting.
- Number of nonzero entries in a row, $\sum_{i=1}^{n}(r_i \neq 0)$.

Experiment Setup

Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

Sort rows r in ascending order based on these scores:

- Control, no sorting.
- Number of nonzero entries in a row, $\sum_{i=1}^{n}(r_i \neq 0)$.
- Total number of terms in a row, $\sum_{i=1}^{n} \mathrm{nterms}(r_i)$.

Experiment Setup

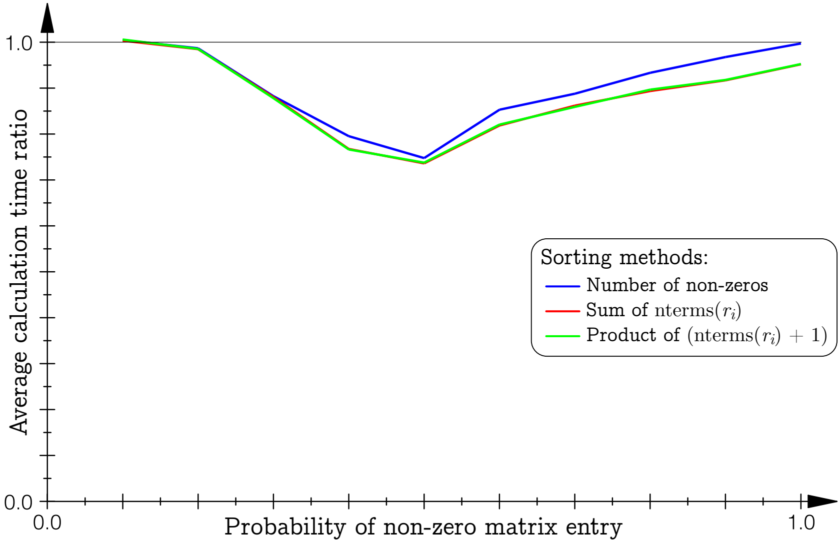
Random polynomial matrices:

- 9×9.
- 5 variables.
- For various values of p: an entry is 0 with probability p and has between one and four terms otherwise. (Any number is equally likely.)
- Each term is constant or linear in each variable.

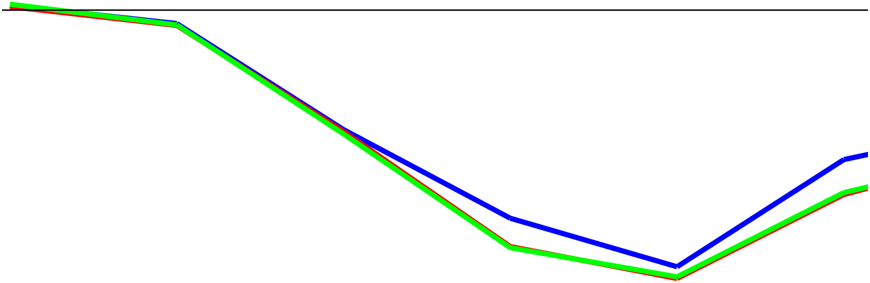
Sort rows r in ascending order based on these scores:

- Control, no sorting.
- Number of nonzero entries in a row, $\sum_{i=1}^{n}(r_i \neq 0)$.
- Total number of terms in a row, $\sum_{i=1}^{n} \mathrm{nterms}(r_i)$.
- Product of one more than number of terms for each entry of a row, $\prod_{i=1}^{n}(\mathrm{nterms}(r_i) + 1)$.

Data


1.0
0.0
Average calculation time ratio
0.0
1.0
Probability of non-zero matrix entry
Sorting methods:
Number of non-zeros
Sum of nterms(r_i)
Product of (nterms(r_i) + 1)

Data



Further Questions

Define “complexity” of matrix entries:

Further Questions

Define “complexity” of matrix entries:

- Number of terms.

Further Questions

Define “complexity” of matrix entries:

- Number of terms.
- Variables present.

Further Questions

Define "complexity" of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Further Questions

Define “complexity” of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

Further Questions

Define "complexity" of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Further Questions

Define "complexity" of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Do more experiments!

Further Questions

Define "complexity" of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Do more experiments!

- Vary other criteria.

Further Questions

Define “complexity” of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Do more experiments!

- Vary other criteria.
- Different algorithms and variations.

Further Questions

Define "complexity" of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Do more experiments!

- Vary other criteria.
- Different algorithms and variations.
- Crossover points between algorithms.

Further Questions

Define “complexity” of matrix entries:

- Number of terms.
- Variables present.
- Context of a row.

Investigate specific types of matrices:

- Bézout matrices.

Do more experiments!

- Vary other criteria.
- Different algorithms and variations.
- Crossover points between algorithms.
- Machine learning.

Acknowledgments

- Dr. Tanya Khovanova, MIT, for mentoring me.
- Dr. Stefan Wehmeier, MathWorks, for suggesting the project and introducing me to the field.
- Dr. Ben Hinkle, MathWorks, for arranging software license and an international conference call.
- MIT PRIMES, for giving me this unique research opportunity.
- The MathWorks, Inc., for providing software and supporting the research.