What can we do with matrices (over finite fields)?

Max Neunhöffer

University of St Andrews

Kirchberg/Hunsrück, 8.–12.8.2011

Definition

Let $\mathbb{F}$ be a field, $A = (A_{i,j}) \in \mathbb{F}^{k \times m}$ and $B = (B_{j,\ell}) \in \mathbb{F}^{m \times n}$. Then the matrix product $C = (C_{i,\ell}) \in \mathbb{F}^{k \times n}$ is the matrix defined by

$$C_{i,\ell} = \sum_{j=1}^{m} A_{i,k} \cdot B_{k,\ell} \qquad \text{for } 1 \leq i \leq k \text{ and } 1 \leq \ell \leq n.$$

Definition

Let $\mathbb{F}$ be a field, $A = (A_{i,j}) \in \mathbb{F}^{k \times m}$ and $B = (B_{j,\ell}) \in \mathbb{F}^{m \times n}$. Then the matrix product $C = (C_{i,\ell}) \in \mathbb{F}^{k \times n}$ is the matrix defined by

$$C_{i,\ell} = \sum_{j=1}^{m} A_{i,k} \cdot B_{k,\ell} \qquad \text{for } 1 \leq i \leq k \text{ and } 1 \leq \ell \leq n.$$

Proposition

Evaluating $C = A \cdot B$ needs $k \cdot m \cdot n$ multiplications and $k \cdot (m-1) \cdot n$ additions in $\mathbb{F}$. For $d = k = m = n$ this is about $2d^3$ elementary field operations.

Definition

Let $\mathbb{F}$ be a field, $A = (A_{i,j}) \in \mathbb{F}^{k \times m}$ and $B = (B_{j,\ell}) \in \mathbb{F}^{m \times n}$. Then the matrix product $C = (C_{i,\ell}) \in \mathbb{F}^{k \times n}$ is the matrix defined by

$$C_{i,\ell} = \sum_{j=1}^{m} A_{i,k} \cdot B_{k,\ell} \qquad \text{for } 1 \leq i \leq k \text{ and } 1 \leq \ell \leq n.$$

Proposition

Evaluating $C = A \cdot B$ needs $k \cdot m \cdot n$ multiplications and $k \cdot (m-1) \cdot n$ additions in $\mathbb{F}$. For $d = k = m = n$ this is about $2d^3$ elementary field operations.

Similarly: Inversion, Gaussian elimination.

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

- Reduces memory usage

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

- Reduces memory usage
- Use processor word operations for vector operations

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

- Reduces memory usage
- Use processor word operations for vector operations
- Memory access is often a bottleneck, since operations are simple

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

- Reduces memory usage
- Use processor word operations for vector operations
- Memory access is often a bottleneck, since operations are simple
- Improve cache-locality

$\Longrightarrow$ Increases performance

For $\mathbb{F}_p$, we need $\lceil \log_2(p) \rceil$ bits of memory to store one field element.

For $\mathbb{F}_q = \mathbb{F}_{p^e}$, we need $e \cdot \lceil \log_2(p) \rceil$ bits of memory.

Idea (Compressed vectors)

Pack for small p some field elements in the same machine word.

- Reduces memory usage
- Use processor word operations for vector operations
- Memory access is often a bottleneck, since operations are simple
- Improve cache-locality

$\Longrightarrow$ Increases performance

Arithmetic can be done table driven or machine word-wise.

Operation	Time		Memory	
	C	U	C	U
Mult. in $\mathbb{F}_2^{4370\times 4370}$	320 ms	1335 s	2.3 MB	152 MB
Add. in $\mathbb{F}_2^{1\times 4370}$	240 ns	209 μs	550 B	35 kB
Mult. in $\mathbb{F}_3^{500\times 500}$	50 ms	2140 ms	78 kB	2 MB

Idea (Grease)

Over a (small) finite field $\mathbb{F}_q$, k rows have only q^k different linear combinations.

Idea (Grease)

Over a (small) finite field $\mathbb{F}_q$, k rows have only q^k different linear combinations. If you need more than q^k linear combinations of them, then make them all beforehand and just look up.

Idea (Grease)

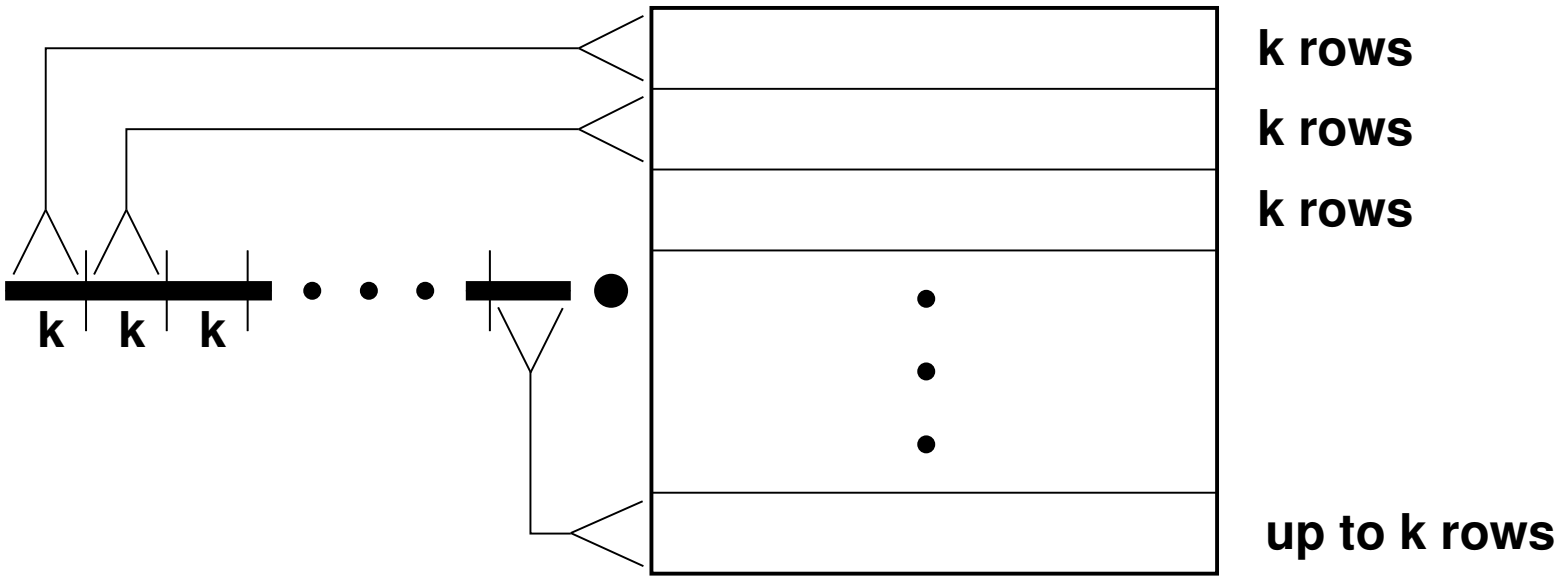
Over a (small) finite field $\mathbb{F}_q$, k rows have only q^k different linear combinations. If you need more than q^k linear combinations of them, then make them all beforehand and just look up.

In matrix multiplication, you do need many linear combinations of the rows of the right hand factor:

Idea (Grease)

Over a (small) finite field $\mathbb{F}_q$, k rows have only q^k different linear combinations. If you need more than q^k linear combinations of them, then make them all beforehand and just look up.

In matrix multiplication, you do need many linear combinations of the rows of the right hand factor:



Can we do better than $2d^3$ for one matrix multiplication?

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

$S_1 := C + D$	$S_2 := S_1 - A$	$S_3 := A - C$	$S_4 := B - S_2$
$S_5 := F - E$	$S_6 := H - S_5$	$S_7 := H - F$	$S_8 := S_6 - G$

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

$S_1 := C + D$	$S_2 := S_1 - A$	$S_3 := A - C$	$S_4 := B - S_2$
$S_5 := F - E$	$S_6 := H - S_5$	$S_7 := H - F$	$S_8 := S_6 - G$
$M_1 := S_2 \cdot S_6$	$M_2 := A \cdot E$	$M_3 := B \cdot G$	$M_4 := S_3 \cdot S_7$

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

$S_1 := C + D$	$S_2 := S_1 - A$	$S_3 := A - C$	$S_4 := B - S_2$
$S_5 := F - E$	$S_6 := H - S_5$	$S_7 := H - F$	$S_8 := S_6 - G$
$M_1 := S_2 \cdot S_6$	$M_2 := A \cdot E$	$M_3 := B \cdot G$	$M_4 := S_3 \cdot S_7$
$M_5 := S_1 \cdot S_5$	$M_6 := S_4 \cdot H$	$M_7 := D \cdot S_8$	☺

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

$$\begin{array}{c|c|c|c} S_1 := C+D & S_2 := S_1 - A & S_3 := A - C & S_4 := B - S_2 \\ \hline S_5 := F - E & S_6 := H - S_5 & S_7 := H - F & S_8 := S_6 - G \end{array}$$

$$\begin{array}{c|c|c|c} M_1 := S_2 \cdot S_6 & M_2 := A \cdot E & M_3 := B \cdot G & M_4 := S_3 \cdot S_7 \\ M_5 := S_1 \cdot S_5 & M_6 := S_4 \cdot H & M_7 := D \cdot S_8 & \text{☺} \end{array}$$

$$\begin{array}{c|c} X := M_2 + M_3 & Y := M_1 + M_2 + M_5 + M_6 \\ \hline V := M_1 + M_2 + M_4 - M_7 & W := M_1 + M_2 + M_4 + M_5 \end{array}$$

Can we do better than $2d^3$ for one matrix multiplication?

Strassen/Winograd multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix} = \begin{bmatrix} X & Y \\ V & W \end{bmatrix}$$

$\Longrightarrow$ needs 8 multiplications and 4 additions of $d/2 \times d/2$-matrices.

$$\begin{array}{c|c|c|c} S_1 := C+D & S_2 := S_1 - A & S_3 := A - C & S_4 := B - S_2 \\ \hline S_5 := F - E & S_6 := H - S_5 & S_7 := H - F & S_8 := S_6 - G \end{array}$$

$$\begin{array}{c|c|c|c} M_1 := S_2 \cdot S_6 & M_2 := A \cdot E & M_3 := B \cdot G & M_4 := S_3 \cdot S_7 \\ M_5 := S_1 \cdot S_5 & M_6 := S_4 \cdot H & M_7 := D \cdot S_8 & \text{☺} \end{array}$$

$$\begin{array}{c|c} X := M_2 + M_3 & Y := M_1 + M_2 + M_5 + M_6 \\ \hline V := M_1 + M_2 + M_4 - M_7 & W := M_1 + M_2 + M_4 + M_5 \end{array}$$

$\Longrightarrow$ needs 7 multiplications and 15 additions of $d/2 \times d/2$-matrices.

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$f(2^k) \quad = \quad 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$f(2^k) \quad = \quad 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1}$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$\begin{aligned} f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\ &= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \end{aligned}$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$\begin{aligned} f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\ &= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\ &= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\ &= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \end{aligned}$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$
\begin{aligned}
f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\
&= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\
&= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\
&= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \\
&\leq 7^{k-1} \cdot \left(12 + \frac{60}{7} \cdot \frac{7}{3}\right) = 32 \cdot 7^{k-1}
\end{aligned}
$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$
\begin{aligned}
f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\
&= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\
&= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\
&= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \\
&\leq 7^{k-1} \cdot \left(12 + \frac{60}{7} \cdot \frac{7}{3}\right) = 32 \cdot 7^{k-1} \\
f(d) &\leq 32 \cdot 7^{\lceil \log_2(d) \rceil - 1}
\end{aligned}
$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$
\begin{aligned}
f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\
&= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\
&= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\
&= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \\
&\leq 7^{k-1} \cdot \left(12 + \frac{60}{7} \cdot \frac{7}{3}\right) = 32 \cdot 7^{k-1} \\
f(d) &\leq 32 \cdot 7^{\lceil \log_2(d) \rceil - 1} = 32 \cdot 2^{(\log_2(7)) \cdot \lceil \log_2(d) \rceil}
\end{aligned}
$$

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$\begin{aligned}
f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\
&= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\
&= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\
&= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \\
&\leq 7^{k-1} \cdot \left(12 + \frac{60}{7} \cdot \frac{7}{3}\right) = 32 \cdot 7^{k-1}
\end{aligned}$$

$$\begin{aligned}
f(d) &\leq 32 \cdot 7^{\lceil \log_2(d) \rceil - 1} = 32 \cdot 2^{(\log_2(7)) \cdot \lceil \log_2(d) \rceil} \\
&= 32 \cdot \tilde{d}^{\log_2(7)} \approx 32 \cdot \tilde{d}^{2.807}
\end{aligned}$$

where $\tilde{d} = 2^k$ if $2^{k-1} < d \leq 2^k$.

Analysis of complexity:

Let $f(d)$ be the number of elementary field operations needed to multiply two $(d \times d)$-matrices.

$$\begin{aligned}
f(2^k) &= 7 \cdot f(2^{k-1}) + 15 \cdot \left(2^{k-1}\right)^2 = 7 \cdot f(2^{k-1}) + 15 \cdot 4^{k-1} \\
&= 7^{k-1} \cdot f(2^1) + 15 \cdot (4^{k-1} + 7 \cdot 4^{k-2} + \cdots + 7^{k-2} \cdot 4^1) \\
&= 12 \cdot 7^{k-1} + 15 \cdot 7^{k-1} \cdot \left((4/7)^{k-1} + \cdots + (4/7)^1\right) \\
&= 12 \cdot 7^{k-1} + 15 \cdot (4/7) \cdot 7^{k-1} \cdot \frac{1 - (4/7)^{k-1}}{1 - 4/7} \\
&\leq 7^{k-1} \cdot \left(12 + \frac{60}{7} \cdot \frac{7}{3}\right) = 32 \cdot 7^{k-1}
\end{aligned}$$

$$\begin{aligned}
f(d) &\leq 32 \cdot 7^{\lceil \log_2(d) \rceil - 1} = 32 \cdot 2^{(\log_2(7)) \cdot \lceil \log_2(d) \rceil} \\
&= 32 \cdot \tilde{d}^{\log_2(7)} \approx 32 \cdot \tilde{d}^{2.807}
\end{aligned}$$

where $\tilde{d} = 2^k$ if $2^{k-1} < d \leq 2^k$.

The best known exponent is 2.3, that is totally useless in practice.

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d \times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d \times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Minimal polynomial)

Let $M \in \mathbb{F}^{d \times d}$, then the minimal polynomial of M is the monic polynomial $\mu_M \in \mathbb{F}[x]$ of least degree for which $\mu_M(M) = 0$ holds. μ_M divides every polynomial $f \in \mathbb{F}[x]$ with $f(M) = 0$.

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Minimal polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the minimal polynomial of M is the monic polynomial $\mu_M \in \mathbb{F}[x]$ of least degree for which $\mu_M(M) = 0$ holds. μ_M divides every polynomial $f \in \mathbb{F}[x]$ with $f(M) = 0$.

Proposition (Invariant factors)

Let $A := x \cdot \mathbf{1} - M \in \mathbb{F}[x]^{d\times d}$. Then there are matrices $S, T \in \mathbb{F}[x]^{d\times d}$ with determinant 1 and

$$S \cdot A \cdot T = \operatorname{diag}(1, \ldots, 1, p_1, p_2, \ldots, p_k)$$

with $p_1 \mid p_2 \mid \cdots \mid p_k$.

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Minimal polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the minimal polynomial of M is the monic polynomial $\mu_M \in \mathbb{F}[x]$ of least degree for which $\mu_M(M) = 0$ holds. μ_M divides every polynomial $f \in \mathbb{F}[x]$ with $f(M) = 0$.

Proposition (Invariant factors)

Let $A := x \cdot \mathbf{1} - M \in \mathbb{F}[x]^{d\times d}$. Then there are matrices $S, T \in \mathbb{F}[x]^{d\times d}$ with determinant 1 and

$$S \cdot A \cdot T = \operatorname{diag}(1, \ldots, 1, p_1, p_2, \ldots, p_k)$$

with $p_1 \mid p_2 \mid \cdots \mid p_k$. The p_i are uniquely determined (up to scalars)

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Minimal polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the minimal polynomial of M is the monic polynomial $\mu_M \in \mathbb{F}[x]$ of least degree for which $\mu_M(M) = 0$ holds. μ_M divides every polynomial $f \in \mathbb{F}[x]$ with $f(M) = 0$.

Proposition (Invariant factors)

Let $A := x \cdot \mathbf{1} - M \in \mathbb{F}[x]^{d\times d}$. Then there are matrices $S, T \in \mathbb{F}[x]^{d\times d}$ with determinant 1 and

$$S \cdot A \cdot T = \operatorname{diag}(1, \ldots, 1, p_1, p_2, \ldots, p_k)$$

with $p_1 \mid p_2 \mid \cdots \mid p_k$. The p_i are uniquely determined (up to scalars) and the matrices S and T can be computed explicitly.

Definition (Characteristic polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the characteristic polynomial $\chi_M \in \mathbb{F}[x]$ of M is $\chi_M := \det(x \cdot \mathbf{1} - M)$.

Definition (Minimal polynomial)

Let $M \in \mathbb{F}^{d\times d}$, then the minimal polynomial of M is the monic polynomial $\mu_M \in \mathbb{F}[x]$ of least degree for which $\mu_M(M) = 0$ holds.
μ_M divides every polynomial $f \in \mathbb{F}[x]$ with $f(M) = 0$.

Proposition (Invariant factors)

Let $A := x \cdot \mathbf{1} - M \in \mathbb{F}[x]^{d\times d}$. Then there are matrices $S, T \in \mathbb{F}[x]^{d\times d}$ with determinant 1 and

$$S \cdot A \cdot T = \operatorname{diag}(1, \ldots, 1, p_1, p_2, \ldots, p_k)$$

with $p_1 \mid p_2 \mid \cdots \mid p_k$. The p_i are uniquely determined (up to scalars) and the matrices S and T can be computed explicitly.
We have $\mu_M = p_r$ and $\chi_M = p_1 \cdot \cdots \cdot p_k$.

Definition (Order polynomial)

$\mathbb{F}$ field, $\mathcal{A}$ f.d. $\mathbb{F}$-algebra, $V \in \mathrm{mod}\text{-}\mathcal{A}$, $v \in V$, $M \in \mathcal{A}$.
Then the order polynomial $q := \mathrm{ord}_M(v) \in \mathbb{F}[x]$ is the monic polynomial of least degree such that $v \cdot q(M) = 0$.

Definition (Order polynomial)

$\mathbb{F}$ field, $\mathcal{A}$ f.d. $\mathbb{F}$-algebra, $V \in \mathrm{mod}\text{-}\mathcal{A}$, $v \in V$, $M \in \mathcal{A}$.
Then the order polynomial $q := \mathrm{ord}_M(v) \in \mathbb{F}[x]$ is the monic polynomial of least degree such that $v \cdot q(M) = 0$.

Definition (Relative order polynomial)

If additionally $W < V$ is M-invariant, then we call $\mathrm{ord}_M(v + W)$ the relative order polynomial of $v + W \in V/W$.

Definition (Order polynomial)

$\mathbb{F}$ field, $\mathcal{A}$ f.d. $\mathbb{F}$-algebra, $V \in \mathrm{mod}\text{-}\mathcal{A}$, $v \in V$, $M \in \mathcal{A}$.
Then the order polynomial $q := \mathrm{ord}_M(v) \in \mathbb{F}[x]$ is the monic polynomial of least degree such that $v \cdot q(M) = 0$.

Definition (Relative order polynomial)

If additionally $W < V$ is M-invariant, then we call $\mathrm{ord}_M(v + W)$ the relative order polynomial of $v + W \in V/W$.

Lemma (Generator of annihilator)

The order polynomial $\mathrm{ord}_M(v)$ *divides every polynomial* $q \in \mathbb{F}[x]$ *with* $v \cdot q(M) = 0$.

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i \rangle_M$ the $\mathbb{F}[M]$-span.

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i \rangle_M$ the $\mathbb{F}[M]$-span. Find smallest $d_1 \in \mathbb{N}$ such that $(v_1, v_1M, v_1M^2, \ldots, v_1M^{d_1})$ is linearly dependent.

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i\rangle_M$ the $\mathbb{F}[M]$-span. Find smallest $d_1 \in \mathbb{N}$ such that $(v_1, v_1M, v_1M^2, \ldots, v_1M^{d_1})$ is linearly dependent. If

$$v_1M^{d_1} = \sum_{i=0}^{d_1-1} a_i v_1 M^i \quad \text{then} \quad \operatorname{ord}_M(v_1) = x^{d_1} - \sum_{i=0}^{d_1-1} a_i x^i.$$

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i\rangle_M$ the $\mathbb{F}[M]$-span. Find smallest $d_1 \in \mathbb{N}$ such that $(v_1, v_1M, v_1M^2, \ldots, v_1M^{d_1})$ is linearly dependent. If

$$v_1M^{d_1} = \sum_{i=0}^{d_1-1} a_i v_1 M^i \quad \text{then} \quad \operatorname{ord}_M(v_1) = x^{d_1} - \sum_{i=0}^{d_1-1} a_i x^i.$$

Choose some $v_2 \in V \setminus \langle v_1\rangle_M$ and find smallest $d_2 \in \mathbb{N}$, such that $(v_1, v_1M, \ldots, v_1M^{d_1-1}, v_2, v_2M, \ldots, v_2M^{d_2})$ is linearly dependent.

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i\rangle_M$ the $\mathbb{F}[M]$-span. Find smallest $d_1 \in \mathbb{N}$ such that $(v_1, v_1M, v_1M^2, \ldots, v_1M^{d_1})$ is linearly dependent. If

$$v_1M^{d_1} = \sum_{i=0}^{d_1-1} a_i v_1 M^i \quad \text{then} \quad \operatorname{ord}_M(v_1) = x^{d_1} - \sum_{i=0}^{d_1-1} a_i x^i.$$

Choose some $v_2 \in V \setminus \langle v_1\rangle_M$ and find smallest $d_2 \in \mathbb{N}$, such that $(v_1, v_1M, \ldots, v_1M^{d_1-1}, v_2, v_2M, \ldots, v_2M^{d_2})$ is linearly dependent. If

$$v_2M^{d_2} = \sum_{i=0}^{d_1-1} b_i v_1 M^i + \sum_{i=0}^{d_2-1} c_i v_2 M^i \quad \text{then}$$

$$\operatorname{ord}_M(v + \langle v_1\rangle_M) = x^{d_2} - \sum_{i=0} c_i x^i.$$

Let $M \in \mathbb{F}^{d\times d}$ and $v_1, \ldots, v_i \in V$ and $V_i := \langle v_1, \ldots, v_i\rangle_M$ the $\mathbb{F}[M]$-span. Find smallest $d_1 \in \mathbb{N}$ such that $(v_1, v_1M, v_1M^2, \ldots, v_1M^{d_1})$ is linearly dependent. If

$$v_1M^{d_1} = \sum_{i=0}^{d_1-1} a_i v_1 M^i \quad \text{then} \quad \operatorname{ord}_M(v_1) = x^{d_1} - \sum_{i=0}^{d_1-1} a_i x^i.$$

Choose some $v_2 \in V \setminus \langle v_1\rangle_M$ and find smallest $d_2 \in \mathbb{N}$, such that $(v_1, v_1M, \ldots, v_1M^{d_1-1}, v_2, v_2M, \ldots, v_2M^{d_2})$ is linearly dependent. If

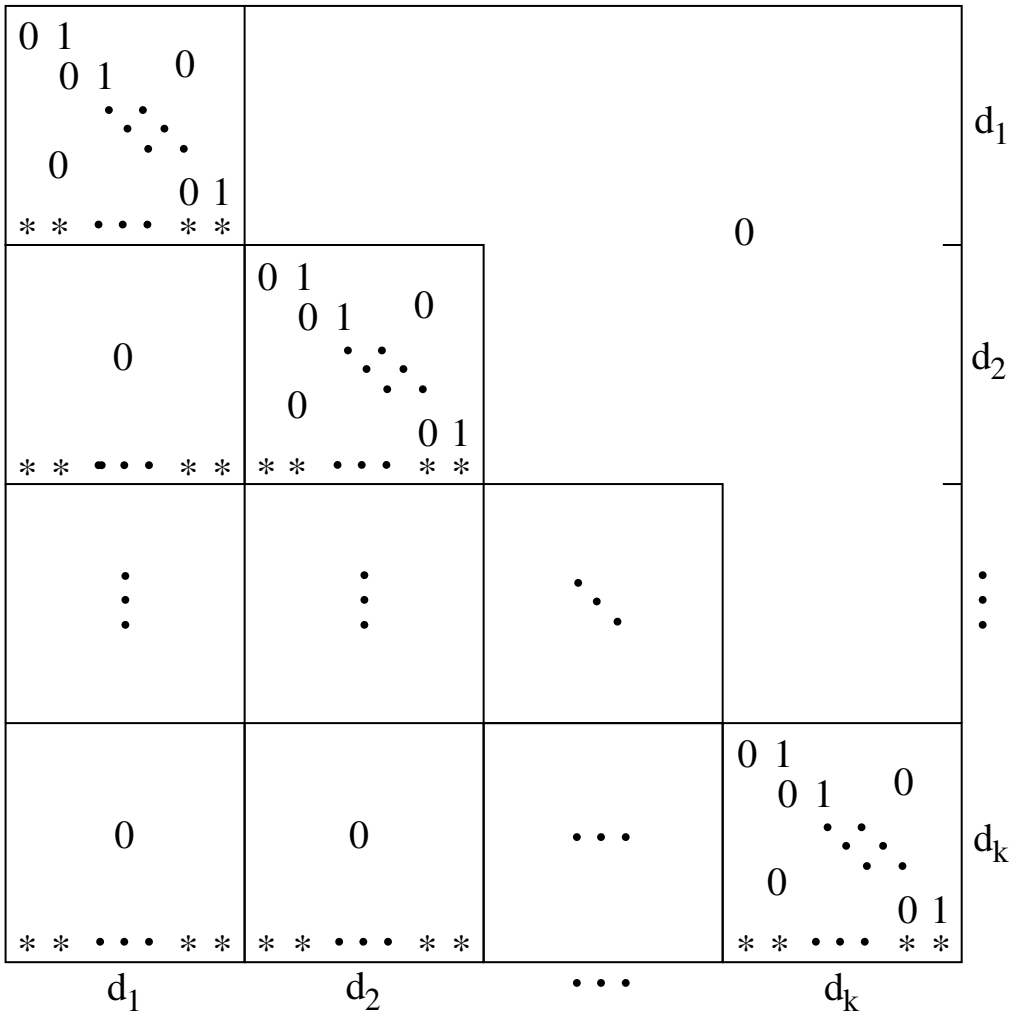
$$v_2M^{d_2} = \sum_{i=0}^{d_1-1} b_i v_1 M^i + \sum_{i=0}^{d_2-1} c_i v_2 M^i \quad \text{then}$$

$$\operatorname{ord}_M(v + \langle v_1\rangle_M) = x^{d_2} - \sum_{i=0} c_i x^i.$$

Going on like this we find an $\mathbb{F}$-basis Y of V:

$$Y := (v_1, v_1M, \ldots, v_1^{d_1-1}, \ldots, v_k, v_kM, \ldots, v_kM_k^{d_k-1}).$$

$$Y \cdot M \cdot Y^{-1} =$$



- Block lower-triangular
- with companion matrices along diagonal
- some sparse garbage below the diagonal

The minimal polynomial

→ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1} \rangle)_M$.

The minimal polynomial

$\rightarrow$ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1}\rangle)_M$.

Lemma (Minimal polynomial)

If $V = \langle v_1, \ldots, v_k\rangle_M$ *then*

$$\mu_M = \mathrm{lcm}(\mathrm{ord}_M(v_1), \ldots, \mathrm{ord}_M(v_k)).$$

The minimal polynomial

$\rightarrow$ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1}\rangle)_M$.

Lemma (Minimal polynomial)

If $V = \langle v_1, \ldots, v_k\rangle_M$ *then*
$\mu_M = \mathrm{lcm}(\mathrm{ord}_M(v_1), \ldots, \mathrm{ord}_M(v_k))$.

Problem:

- $\dim_{\mathbb{F}}(V_i) - \dim_{\mathbb{F}}(V_{i-1})$ might be small
- even if $\dim_{\mathbb{F}}(V_i)$ is big.

(set $V_i := \langle v_1, \ldots, v_i\rangle_M$)

The minimal polynomial

$\rightarrow$ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1}\rangle)_M$.

Lemma (Minimal polynomial)

If $V = \langle v_1, \ldots, v_k\rangle_M$ *then*
$\mu_M = \mathrm{lcm}(\mathrm{ord}_M(v_1), \ldots, \mathrm{ord}_M(v_k))$.

Problem:

- $\dim_{\mathbb{F}}(V_i) - \dim_{\mathbb{F}}(V_{i-1})$ might be small
- even if $\dim_{\mathbb{F}}(V_i)$ is big.

(set $V_i := \langle v_1, \ldots, v_i\rangle_M$)

Characteristic polynomial: asymptotically $\leq 5n^3$ field ops.

The minimal polynomial

$\rightarrow$ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1}\rangle)_M$.

Lemma (Minimal polynomial)

If $V = \langle v_1, \ldots, v_k\rangle_M$ *then*
$\mu_M = \mathrm{lcm}(\mathrm{ord}_M(v_1), \ldots, \mathrm{ord}_M(v_k))$.

Problem:

- $\dim_{\mathbb{F}}(V_i) - \dim_{\mathbb{F}}(V_{i-1})$ might be small
- even if $\dim_{\mathbb{F}}(V_i)$ is big.

(set $V_i := \langle v_1, \ldots, v_i\rangle_M$)

Characteristic polynomial: asymptotically $\leq 5n^3$ field ops.

Minimal polynomial: asymptotically $\sim n^4$ field ops.

(both worst case analysis)

The minimal polynomial

$\rightarrow$ compute the absolute order polynomials $\mathrm{ord}_M(v_i)$
instead of the relative ones $\mathrm{ord}_M(v_i + \langle v_1, \ldots, v_{i-1}\rangle)_M$.

Lemma (Minimal polynomial)

If $V = \langle v_1, \ldots, v_k\rangle_M$ *then*
$\mu_M = \mathrm{lcm}(\mathrm{ord}_M(v_1), \ldots, \mathrm{ord}_M(v_k))$.

Problem:

- $\dim_{\mathbb{F}}(V_i) - \dim_{\mathbb{F}}(V_{i-1})$ might be small
- even if $\dim_{\mathbb{F}}(V_i)$ is big.

(set $V_i := \langle v_1, \ldots, v_i\rangle_M$)

Characteristic polynomial: asymptotically $\leq 5n^3$ field ops.

Minimal polynomial: asymptotically $\sim n^4$ field ops.

(both worst case analysis)

Minimal polynomial: can be done in asymptotically $\sim n^3$.

Proposition (N., Praeger (2008))

Let $\mathbb{F} = \mathbb{F}_q$, randomise $v_1, \ldots, v_u \in V$ independently and uniformly distributed, $\chi_M = \prod_{i=1}^{t} q_i^{e_i}$. Then:

$$\text{Prob}\left(\text{lcm}(\text{ord}_M(v_1), \ldots, \text{ord}_M(v_u)) = \mu_M\right)$$

is at least $\prod_{i=1}^{t}(1 - q^{-u\deg(q_i)})$.

Proposition (N., Praeger (2008))

Let $\mathbb{F} = \mathbb{F}_q$, randomise $v_1, \ldots, v_u \in V$ independently and uniformly distributed, $\chi_M = \prod_{i=1}^{t} q_i^{e_i}$. Then:

$$\text{Prob}\left(\text{lcm}(\text{ord}_M(v_1), \ldots, \text{ord}_M(v_u)) = \mu_M\right)$$

is at least $\prod_{i=1}^{t}(1 - q^{-u\deg(q_i)})$.

Algorithm MINIMALPOLYNOMIALMC: Input M, $0 < \epsilon < 1/2$

- Compute χ_M, Y, $\text{ord}_M(v_i + V_{i-1})$ for $1 \leq i \leq k$
- Determine least u, such that probability $> 1 - \epsilon$
- Compute $\text{ord}_M(v_1), \ldots, \text{ord}_M(v_u)$
- Return least common multiple

Proposition (N., Praeger (2008))

Let $\mathbb{F} = \mathbb{F}_q$, randomise $v_1, \ldots, v_u \in V$ independently and uniformly distributed, $\chi_M = \prod_{i=1}^{t} q_i^{e_i}$. Then:

$$\text{Prob}\left(\text{lcm}(\text{ord}_M(v_1), \ldots, \text{ord}_M(v_u)) = \mu_M\right)$$

is at least $\prod_{i=1}^{t}(1 - q^{-u\deg(q_i)})$.

Algorithm MINIMALPOLYNOMIALMC: Input M, $0 < \epsilon < 1/2$

- Compute χ_M, Y, $\text{ord}_M(v_i + V_{i-1})$ for $1 \leq i \leq k$
- Determine least u, such that probability $> 1 - \epsilon$
- Compute $\text{ord}_M(v_1), \ldots, \text{ord}_M(v_u)$
- Return least common multiple

Needs asymptotically $\leq 5n^3 + \mathit{Factorisation}(n)$ field ops.

How can we compute the centraliser of an involution?

How can we compute the centraliser of an involution?

The following method by John Bray does the job:

Algorithm: InvolutionCentraliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ and an involution $x \in G$.
initialise *gens* := $[x]$
repeat
$y :=$ RandomElement(G)
$c := x^{-1}y^{-1}xy$ **and** $o :=$ Order(c)

How can we compute the centraliser of an involution?

The following method by John Bray does the job:

Algorithm: InvolutionCentraliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ and an involution $x \in G$.
initialise *gens* := $[x]$
repeat
    $y :=$ RandomElement(G)
    $c := x^{-1}y^{-1}xy$ **and** $o :=$ Order(c)
    if o is even **then**
        append $c^{o/2}$ and $(x^{-1}yxy^{-1})^{o/2}$ to *gens*
    else
        append $z := y \cdot c^{(o-1)/2}$ to *gens*

How can we compute the centraliser of an involution?

The following method by John Bray does the job:

Algorithm: InvolutionCentraliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ and an involution $x \in G$.
initialise *gens* := $[x]$
repeat
    $y :=$ RandomElement(G)
    $c := x^{-1}y^{-1}xy$ **and** $o :=$ Order(c)
    if o is even **then**
        append $c^{o/2}$ and $(x^{-1}yxy^{-1})^{o/2}$ to *gens*
    else
        append $z := y \cdot c^{(o-1)/2}$ to *gens*
until o was odd often enough or gens long enough
return *gens*

How can we compute the centraliser of an involution?

The following method by John Bray does the job:

Algorithm: InvolutionCentraliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ and an involution $x \in G$.
initialise *gens* := $[x]$
repeat
  $y :=$ RandomElement(G)
  $c := x^{-1}y^{-1}xy$ **and** $o :=$ Order(c)
  if o is even **then**
    append $c^{o/2}$ and $(x^{-1}yxy^{-1})^{o/2}$ to *gens*
  else
    append $z := y \cdot c^{(o-1)/2}$ to *gens*
until o was odd often enough or gens long enough
return *gens*

Note: If $xy = yx$ then $c = 1_G$ and $o = 1$ and $z = y$.

How can we compute the centraliser of an involution?

The following method by John Bray does the job:

Algorithm: InvolutionCentraliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ and an involution $x \in G$.
initialise *gens* := $[x]$
repeat
  $y :=$ RandomElement(G)
  $c := x^{-1}y^{-1}xy$ **and** $o :=$ Order(c)
  if o is even **then**
    append $c^{o/2}$ and $(x^{-1}yxy^{-1})^{o/2}$ to *gens*
  else
    append $z := y \cdot c^{(o-1)/2}$ to *gens*
until o was odd often enough or gens long enough
return *gens*

Note: If $xy = yx$ then $c = 1_G$ and $o = 1$ and $z = y$.

And: If o is odd, then z is uniformly distributed in $C_G(x)$.