In-place fast polynomial modular remainder

Jean-Guillaume Dumas & **Bruno Grenet**

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES · INFORMATIQUE

Université Grenoble Alpes

ISSAC, Raleigh, NC
July 16–19, 2024

Introduction

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ without extra space as fast as possible

Introduction

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ without extra space as fast as possible

Space complexity models

1. `ro/wo`: read-only inputs & write-only output
   - 👍 Classical model in complexity theory
   - 👎 $\Omega(n^2)$ lower bounds on time × space
2. `ro/rw`: read-only inputs & read-write output
   - 👍 Allows parallel access to the inputs
   - 👎 Too restrictive in a sequential model
3. `rw/rw`: read-write inputs & read-write output, *inputs restored at the end*
   - 👍 Allows recursive calls / use as subroutines
   - 👎 Not suitable for parallel programming

Introduction

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ without extra space as fast as possible

Space complexity models

1. `ro/wo`: read-only inputs & write-only output
   - 👍 Classical model in complexity theory
   - 👎 $\Omega(n^2)$ lower bounds on time × space
2. `ro/rw`: read-only inputs & read-write output
   - 👍 Allows parallel access to the inputs
   - 👎 Too restrictive in a sequential model
3. `rw/rw`: read-write inputs & read-write output, *inputs restored at the end*
   - 👍 Allows recursive calls / use as subroutines
   - 👎 Not suitable for parallel programming

alternative view: no distinction input/output $\rightarrow$ full specification

- $C \leftarrow A \bmod B$: $(A, B, C) \mapsto (A, B, A \bmod B)$ *in place*
- $C \mathrel{+}= AB$: $(A, B, C) \mapsto (A, B, C + AB)$ *in place with accumulation*
- $B \mathrel{*}= A$: $(A, B) \mapsto (A, AB)$ *over-place*

Known results

$A = BQ + R$ with $\deg(R) < \deg(B)$ $\quad n = \deg(A)$, $m = \deg(B)$, $q = n - m$, $k = \max(q, m)$

Algorithm	Model		QuoRem	Quotient only	Remainder only
Classical	ro/wo	time	$O(qm)$	same result [Monagan'93]	same result [Monagan'93]
		space	$O(1)$		
Fast [Strassen'73, ...]	ro/wo	time	$O(\mathsf{M}(k))$	same result	same result
		space	$O(k)$		
Fast & low space [Giorgi G. Roche'20]	ro/rw	time	$O(\mathsf{M}(k))$	$O(\mathsf{M}(q)\log q)^*$	$O(\mathsf{M}(k))$
		space	$O(1)$	$O(1)$	$\min(q, m) + O(1)$

* $O(\mathsf{M}(q))$ if $\mathsf{M}(q) = \Omega(q^{1+\varepsilon})$

Known results

$A = BQ + R$ with $\deg(R) < \deg(B)$ $\quad n = \deg(A)$, $m = \deg(B)$, $q = n - m$, $k = \max(q, m)$

Algorithm	Model		QuoRem	Quotient only	Remainder only
Classical	ro/wo	time	$O(qm)$	same result [Monagan'93]	same result [Monagan'93]
		space	$O(1)$		
Fast [Strassen'73, ...]	ro/wo	time	$O(\mathsf{M}(k))$	same result	same result
		space	$O(k)$		
Fast & low space [Giorgi G. Roche'20]	ro/rw	time	$O(\mathsf{M}(k))$	$O(\mathsf{M}(q)\log q)^*$	$O(\mathsf{M}(k))$
		space	$O(1)$	$O(1)$	$\min(q, m) + O(1)$

* $O(\mathsf{M}(q))$ if $\mathsf{M}(q) = \Omega(q^{1+\varepsilon})$

- Open problems that "seem" impossible: Remainder, modular inverse, factorization

Dan Roche's slides @ ISSAC'20 [Giorgi G. Roche'20]

Our results

In-place remainder (model `rw`/`rw` but *A read-only*)

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ in time $O(\mathsf{M}(k)\log m)^*$ and space $O(1)^\dagger$

$n = \deg(A)$, $m = \deg(B)$, $k = \max(n - m, m)$

* $O(\mathsf{M}(k))$ if $\mathsf{M}(k) = \Omega(k^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Our results

In-place remainder

(model `rw/rw` but *A read-only*)

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ in time $O(\mathsf{M}(k)\log m)^*$ and space $O(1)^\dagger$

$n = \deg(A)$, $m = \deg(B)$, $k = \max(n - m, m)$

* $O(\mathsf{M}(k))$ if $\mathsf{M}(k) = \Omega(k^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Variants (same time & space)

- $C \mathrel{+}= A \bmod B$
- $R \mathrel{+}= AB \bmod C$
- $(A, B) \mapsto (Q|R, B)$

Our results

In-place remainder

(model `rw/rw` but *A read-only*)

Given $A, B \in \mathbb{F}[x]$, compute $A \bmod B$ in time $O(\mathsf{M}(k)\log m)^*$ and space $O(1)^\dagger$

$n = \deg(A)$, $m = \deg(B)$, $k = \max(n - m, m)$

* $O(\mathsf{M}(k))$ if $\mathsf{M}(k) = \Omega(k^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Variants (same time & space)

- $C \mathrel{+}= A \bmod B$
- $R \mathrel{+}= AB \bmod C$
- $(A, B) \mapsto (Q|R, B)$

Building blocks: in-place Toeplitz computations

- $\vec{c} \mathrel{+}= A \cdot \vec{b}$ where A is a Toeplitz matrix — *in place with accumulation*
  - polynomial short and middle products
- $\vec{b} \leftarrow A \cdot \vec{b}$ and $\vec{b} \leftarrow A^{-1} \cdot \vec{b}$ where A is a triangular Toeplitz matrix — *over place*
  - (truncated) power series multiplication and division

Contents

Goals

Main goal: *Over-place* triangular Toeplitz computations

- $\vec{b} \leftarrow A \cdot \vec{b}$ where A is a lower triangular Toeplitz matrix
- $\vec{b} \leftarrow A^{-1} \cdot \vec{b}$ where A is an upper triangular Toeplitz matrix

Intermediate goal: *In-place* Toeplitz computations *with accumulation*

- $\vec{c} \mathrel{+}= A \cdot \vec{b}$ where A is
  - a triangular Toeplitz matrix
  - a square Toeplitz matrix
  - a rectangular Toeplitz matrix

The tool: *In-place* polynomial convolutions *with accumulation*

- $C \mathrel{+}= A \cdot B \bmod x^n - \lambda$ for any $\lambda \in \mathbb{F}$
  - $\vec{c} \mathrel{+}= A \cdot \vec{b}$ for a λ-circulant matrix A

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ \lambda \cdot a_{n-1} & a_0 & \cdots & a_{n-3} & a_{n-2} \\ \lambda \cdot a_{n-2} & \lambda \cdot a_{n-1} & \cdots & a_{n-4} & a_{n-3} \\ & & \ddots & & \\ \lambda \cdot a_1 & \lambda \cdot a_2 & \cdots & \lambda \cdot a_{n-1} & a_0 \end{bmatrix}$$

Convolution

$$A = a_0 + x^{n/2} a_1$$

$$B = b_0 + x^{n/2} b_1$$

$$\begin{array}{lcl}
 & & a_0 \quad a_1 \\
 & \times & b_0 \quad b_1 \\
\hline
a_0 b_0 & + & m_0^{00} \quad m_1^{00} \\
a_1 b_0 & + & \quad m_0^{10} \quad m_1^{10} \\
a_0 b_1 & + & \quad m_0^{01} \quad m_1^{01} \\
a_1 b_1 & + & \qquad m_0^{11} \quad m_1^{11}
\end{array}$$

Convolution

$$A = a_0 + x^{n/2} a_1$$

$$B = b_0 + x^{n/2} b_1$$

$$\begin{array}{lcl}
 & & a_0 \quad a_1 \\
 & \times & b_0 \quad b_1 \qquad \bmod x^n - 1 \\
\hline
a_0 b_0 & + & m_0^{00} \quad m_1^{00} \\
a_1 b_0 & + & m_1^{10} \quad m_0^{10} \\
a_0 b_1 & + & m_1^{01} \quad m_0^{01} \\
a_1 b_1 & + & m_0^{11} \quad m_1^{11}
\end{array}$$

Convolution

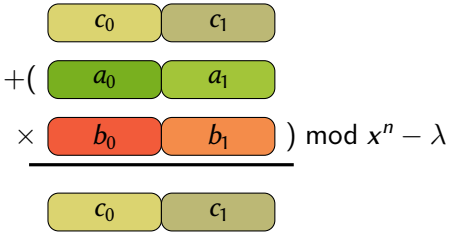
$$A = a_0 + x^{n/2} a_1$$

$$B = b_0 + x^{n/2} b_1$$

		a_0	a_1	
	$\times$	b_0	b_1	$\bmod x^n - \lambda$
$a_0 b_0$	$+$	m_0^{00}	m_1^{00}	
$a_1 b_0$	$+$	$\lambda \cdot m_1^{10}$	m_0^{10}	
$a_0 b_1$	$+$	$\lambda \cdot m_1^{01}$	m_0^{01}	
$a_1 b_1$	$+$	$\lambda \cdot m_0^{11}$	$\lambda \cdot m_1^{11}$	

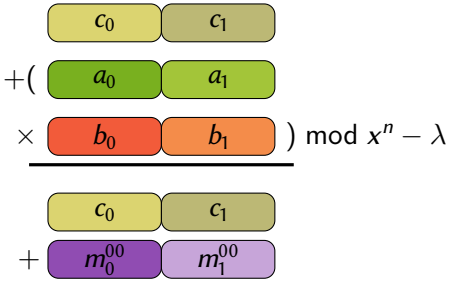
Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_0$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$



Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_0$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$



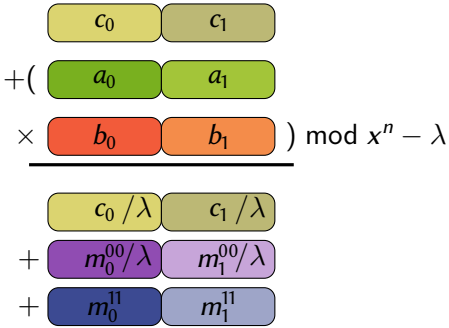
Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_0$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$

$$
\begin{array}{rcc}
 & c_0 & c_1 \\
+(& a_0 & a_1 \\
\times & b_0 & b_1 \quad) \bmod x^n - \lambda \\
\hline
 & c_0/\lambda & c_1/\lambda \\
+ & m_0^{00}/\lambda & m_1^{00}/\lambda
\end{array}
$$

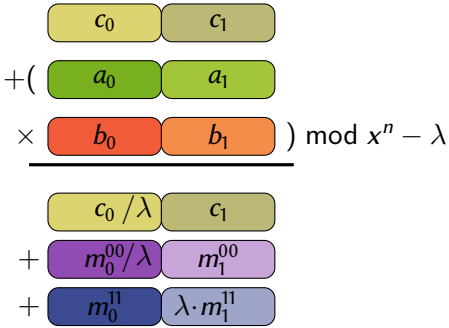
Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_1$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$



Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_0$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$



Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_1$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$

$$
\begin{array}{rcc}
 & c_0 & c_1 \\
+(& a_0 & a_1 \\
\times & b_0 & b_1 \quad) \bmod x^n - \lambda \\
\hline
 & c_0 / \lambda & c_1 \\
+ & m_0^{00} / \lambda & m_1^{00} \\
+ & m_0^{11} & \lambda \cdot m_1^{11} \\
+ & m_1^{01} & m_0^{01}
\end{array}
$$

Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_1$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$

$$
\begin{array}{rcc}
 & c_0 & c_1 \\
+(& a_0 & a_1 \\
\times & b_0 & b_1 \quad) \bmod x^n - \lambda \\
\hline
 & c_0 / \lambda & c_1 \\
+ & m_0^{00} / \lambda & m_1^{00} \\
+ & m_0^{11} & \lambda \cdot m_1^{11} \\
+ & m_1^{01} & m_0^{01} \\
+ & m_1^{10} & m_0^{10}
\end{array}
$$

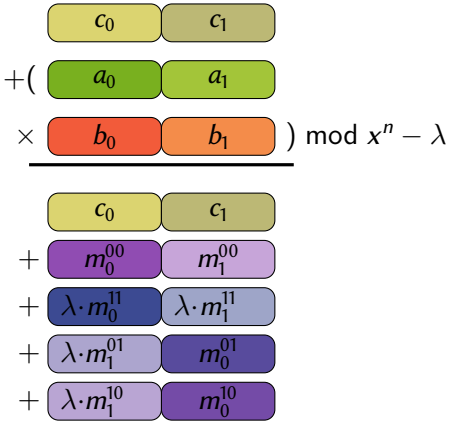
Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_0$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$

$$
\begin{array}{rcc}
 & c_0 & c_1 \\
+(& a_0 & a_1 \\
\times & b_0 & b_1 \quad) \bmod x^n - \lambda \\
\hline
 & c_0 & c_1 \\
+ & m_0^{00} & m_1^{00} \\
+ & \lambda \cdot m_0^{11} & \lambda \cdot m_1^{11} \\
+ & \lambda \cdot m_1^{01} & m_0^{01} \\
+ & \lambda \cdot m_1^{10} & m_0^{10}
\end{array}
$$

Convolution: $C \mathrel{+}= AB \bmod x^n - \lambda$, $\lambda \neq 0$

1. $(c_0, c_1) \mathrel{+}= a_0 \cdot b_0$
2. $C \mathrel{/}= \lambda$
3. $(c_0, c_1) \mathrel{+}= a_1 \cdot b_1$
4. $c_1 \mathrel{*}= \lambda$
5. $(c_1, c_0) \mathrel{+}= a_0 \cdot b_1$
6. $(c_1, c_0) \mathrel{+}= a_1 \cdot b_0$
7. $c_0 \mathrel{*}= \lambda$



Remarks

- Requires $C \mathrel{+}= AB$ [D.G.'24 (previous talk)]
- Case n odd: additional technicalities
- Karatsuba-like improvement

- Time: $O(\mathsf{M}(n))$
- Space: $O(1)$

Convolution: case $\lambda = 0$

Remark

- If $A \cdot B = x^n \times Q + R$

  then $A \cdot B = x^n - \mu \times Q + R + \mu Q$

Convolution: case $\lambda = 0$

Remark

- If $A \cdot B = x^n \times Q + R$

  then $A \cdot B = x^n - \mu \times Q + R + \mu Q$

- If $\alpha + (1-\alpha)\cdot\mu = 0$ $\qquad$ $\alpha \notin \{0,1\}$ and $\mu = \alpha(\alpha-1)^{-1}$

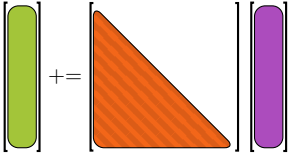
$$\alpha \underbrace{R+Q}_{(AB \bmod x^n - 1)} + (1-\alpha) \underbrace{R+\mu Q}_{(AB \bmod x^n - \mu)} = \underbrace{R}_{(AB \bmod x^n)}$$

Algorithms: $C \mathrel{+}= A \cdot B \bmod x^n$

- If $\mathbb{F} \neq \mathbb{F}_2$: two accumulated convolutions $\bmod x^n - 1$ and $\bmod x^n - \mu$
- If $\mathbb{F} = \mathbb{F}_2$: dedicated algorithm based on a modified Toom-3 algorithm

- Time: $O(\mathrm{M}(n))$
- Space: $O(1)$

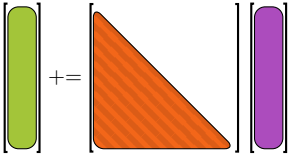
Accumulated Toeplitz matrix-vector product



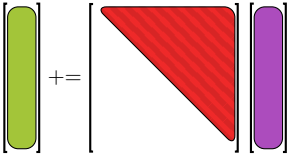
$C \mathrel{+}= A \cdot B \bmod x^n$

- Time: $O(\mathrm{M}(n))$
- Space: $O(1)$

Accumulated Toeplitz matrix-vector product



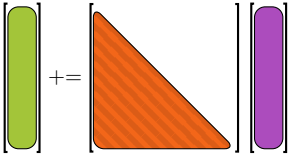
$C \mathrel{+}= A \cdot B \bmod x^n$



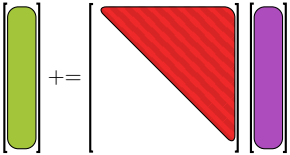
$C \mathrel{+}= A \cdot B \operatorname{div} x^n$

- Time: $O(\mathrm{M}(n))$
- Space: $O(1)$

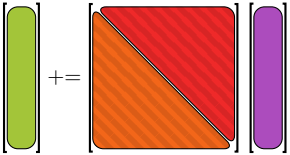
Accumulated Toeplitz matrix-vector product



$C \mathrel{+}= A \cdot B \bmod x^n$



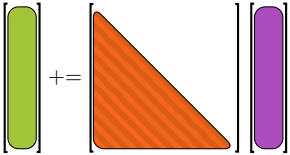
$C \mathrel{+}= A \cdot B \operatorname{div} x^n$



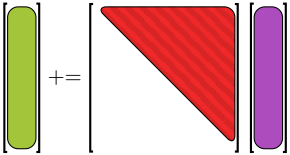
$C \mathrel{+}= \textsc{MidProd}(A, B)$

- Time: $O(\mathsf{M}(n))$
- Space: $O(1)$

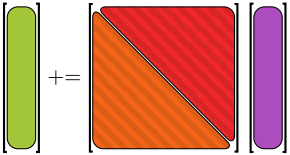
Accumulated Toeplitz matrix-vector product



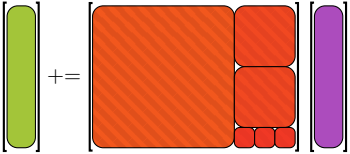
$C \mathrel{+}= A \cdot B \bmod x^n$



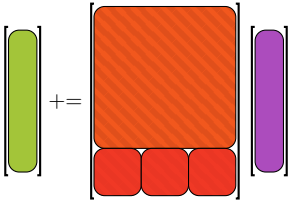
$C \mathrel{+}= A \cdot B \operatorname{div} x^n$



$C \mathrel{+}= \textsc{MidProd}(A, B)$



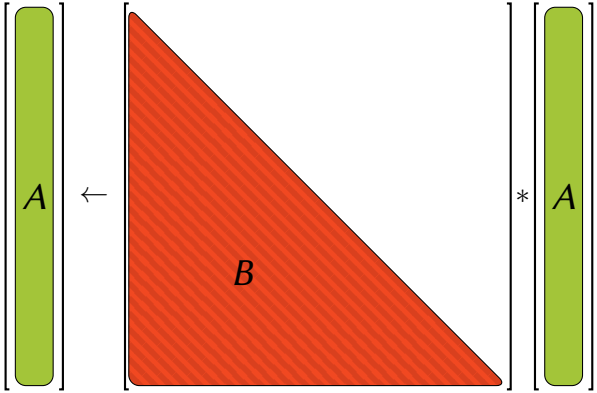
$C \mathrel{+}= \textsc{MidProd}(A, B)$



$C \mathrel{+}= \textsc{MidProd}(A, B)$

- Time: $O(\mathsf{M}(n))$ or $O(\frac{n}{m}\mathsf{M}(m))$
- Space: $O(1)$

Over-place triangular Toeplitz matrix-vector product

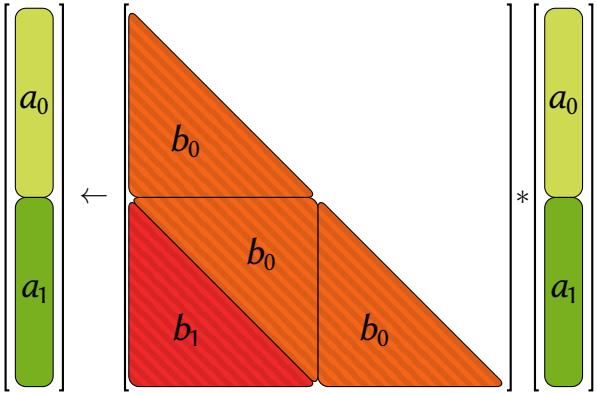


$A \leftarrow B \cdot A \bmod x^n$

Inputs: A and B of degree $< n$ (even)

Output: $A \leftarrow A \cdot B \bmod x^n$

Over-place triangular Toeplitz matrix-vector product



$A \leftarrow B \cdot A \bmod x^n$

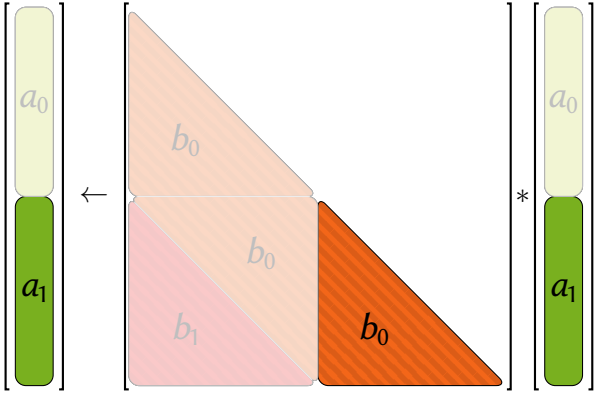
Inputs: A and B of degree $< n$ (even)

Output: $A \leftarrow A \cdot B \bmod x^n$

0. $A = a_0 + x^{n/2} a_1$

$B = b_0 + x^{n/2} b_1$

Over-place triangular Toeplitz matrix-vector product



$A \leftarrow B \cdot A \bmod x^n$

Inputs: A and B of degree $< n$ (even)

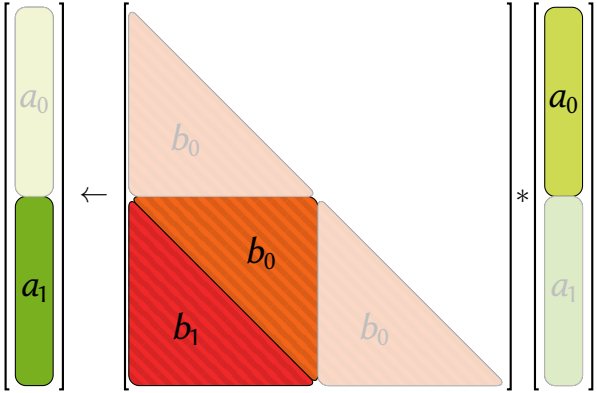
Output: $A \leftarrow A \cdot B \bmod x^n$

0. $A = a_0 + x^{n/2} a_1$

$B = b_0 + x^{n/2} b_1$

1. $a_1 \mathrel{*}= b_0 \bmod x^{n/2}$ (recursive call)

Over-place triangular Toeplitz matrix-vector product



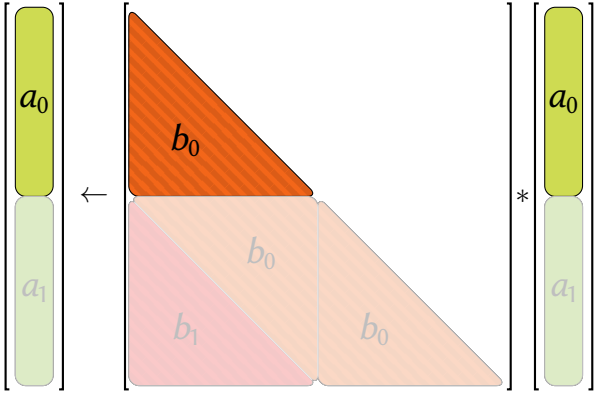
$A \leftarrow B \cdot A \bmod x^n$

Inputs: A and B of degree $< n$ (even)

Output: $A \leftarrow A \cdot B \bmod x^n$

0. $A = a_0 + x^{n/2} a_1$
   $B = b_0 + x^{n/2} b_1$
1. $a_1 *= b_0 \bmod x^{n/2}$ (recursive call)
2. $a_1 += \textsc{MidProd}(a_0, B)$

Over-place triangular Toeplitz matrix-vector product



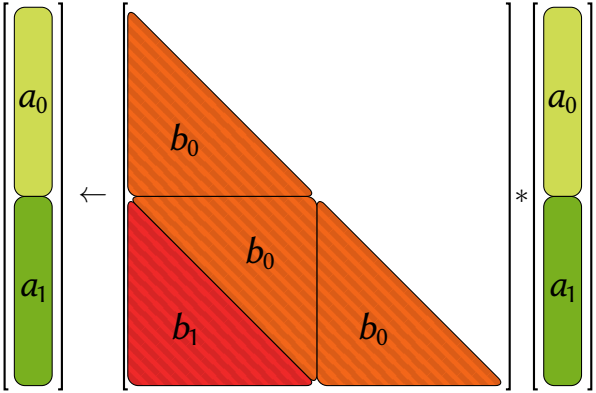
$A \leftarrow B \cdot A \bmod x^n$

Inputs: A and B of degree $< n$ (even)

Output: $A \leftarrow A \cdot B \bmod x^n$

0. $A = a_0 + x^{n/2} a_1$
   $B = b_0 + x^{n/2} b_1$
1. $a_1 *= b_0 \bmod x^{n/2}$ (recursive call)
2. $a_1 += \textsc{MidProd}(a_0, B)$
3. $a_0 *= b_0 \bmod x^{n/2}$ (recursive call)

Over-place triangular Toeplitz matrix-vector product



$A \leftarrow B \cdot A \bmod x^n$

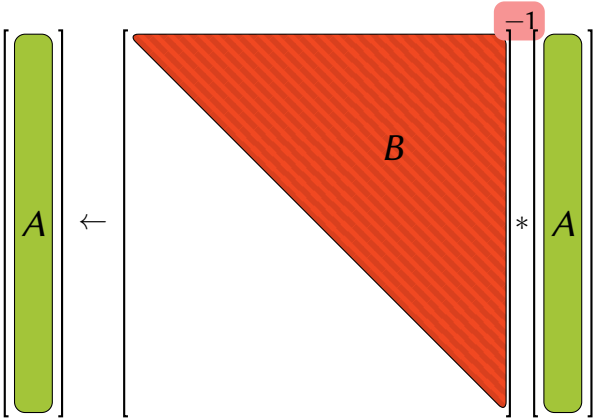
Inputs: A and B of degree $< n$ (even)
Output: $A \leftarrow A \cdot B \bmod x^n$

0. $A = a_0 + x^{n/2} a_1$
   $B = b_0 + x^{n/2} b_1$
1. $a_1 *= b_0 \bmod x^{n/2}$ (recursive call)
2. $a_1 += \textsc{MidProd}(a_0, B)$
3. $a_0 *= b_0 \bmod x^{n/2}$ (recursive call)

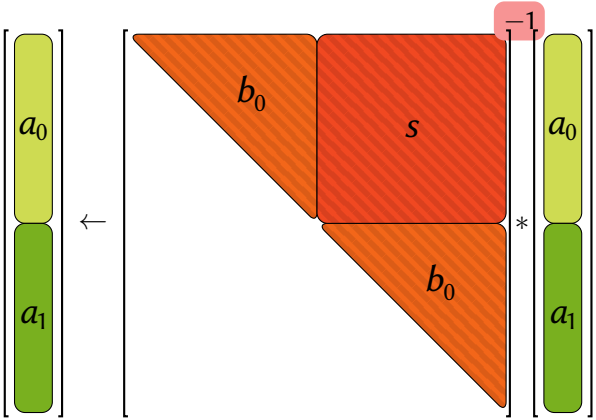
- Time: $O(\mathsf{M}(n) \log n)^*$
- Space: $O(1)^\dagger$

* $O(\mathsf{M}(n))$ if $\mathsf{M}(n) = \Omega(n^{1+\varepsilon})$
† $O(\log n)$ pointers for call stack

Over-place triangular Toeplitz system solving

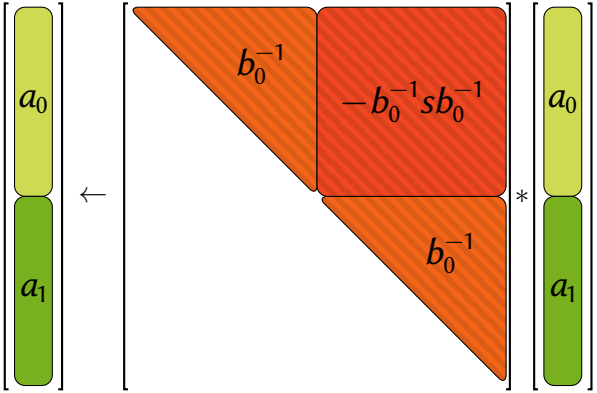


Over-place triangular Toeplitz system solving



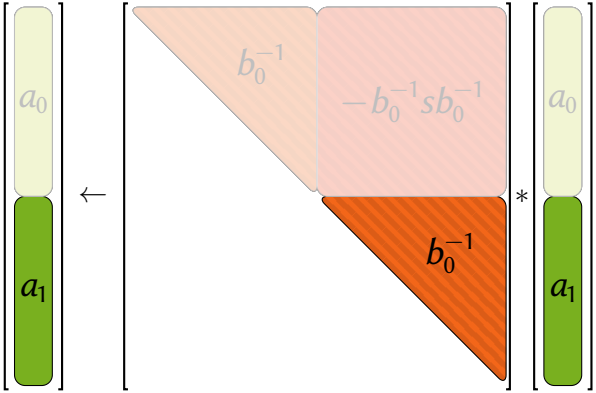
0. $A \rightarrow a_0$ and a_1
$B \rightarrow b_0$ and s

Over-place triangular Toeplitz system solving



0. $A \rightarrow a_0$ and a_1
$B \rightarrow b_0$ and s

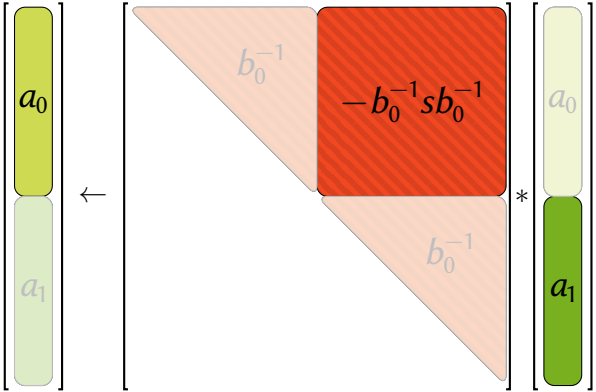
Over-place triangular Toeplitz system solving



0. $A \to a_0$ and a_1
$B \to b_0$ and s

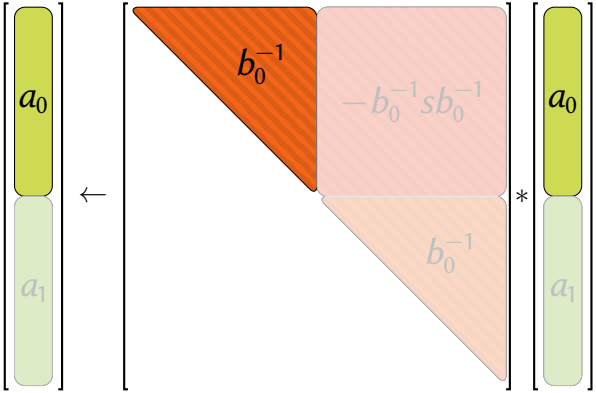
1. $a_1 *= b_0^{-1}$ (recursive call)

Over-place triangular Toeplitz system solving



0. $A \to a_0$ and a_1
   $B \to b_0$ and s
1. $a_1 *= b_0^{-1}$ (recursive call)
2. $a_0 -= s \cdot a_1$

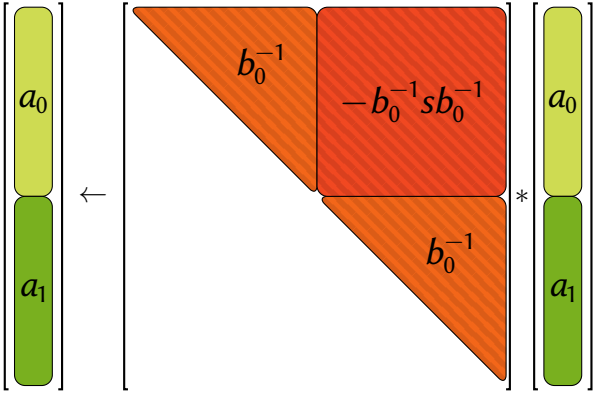
Over-place triangular Toeplitz system solving



0. $A \to a_0$ and a_1
   $B \to b_0$ and s
1. $a_1 *= b_0^{-1}$ (recursive call)
2. $a_0 -= s \cdot a_1$
3. $a_0 *= b_0^{-1}$ (recursive call)

$b_0^{-1}(a_0 - sb_0^{-1}a_1)$	$b_0^{-1}a_1$

Over-place triangular Toeplitz system solving



0. $A \to a_0$ and a_1
   $B \to b_0$ and s
1. $a_1 *= b_0^{-1}$ (recursive call)
2. $a_0 -= s \cdot a_1$
3. $a_0 *= b_0^{-1}$ (recursive call)

- Time: $O(\mathsf{M}(n)\log n)^*$
- Space: $O(1)^\dagger$

* $O(\mathsf{M}(n))$ if $\mathsf{M}(n) = \Omega(n^{1+\varepsilon})$
† $O(\log n)$ pointers for call stack

Contents

$A = B \times Q + R$ as matrix-vector products

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
=
\begin{bmatrix}
b_0 & & & & & & & & & & & & & \\
b_1 & b_0 & & & & & & & & & & & & \\
b_2 & b_1 & b_0 & & & & & & & & & & & \\
b_3 & b_2 & b_1 & b_0 & & & & & & & & & 0 & \\
b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & & & \\
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & & & & \\
 & & & & & & b_5 & b_4 & b_3 & b_2 & & & & \\
 & & & & & & & b_5 & b_4 & b_3 & & & & \\
 & & & & 0 & & & & b_5 & b_4 & & & & \\
 & & & & & & & & & b_5 & & & &
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix}
+
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$A = B \times Q + R$ as matrix-vector products

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ \hline a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
=
\left[\begin{array}{ccccccccccccc}
b_0 & & & & & & & & & & & & \\
b_1 & b_0 & & & & & & & & & & & \\
b_2 & b_1 & b_0 & & & & & & & & & & \\
b_3 & b_2 & b_1 & b_0 & & & & & & & & & \\
b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & & \\
\hline
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & & & \\
 & & & & & & b_5 & b_4 & b_3 & b_2 & & & \\
 & & & & & & & b_5 & b_4 & b_3 & & & \\
 & & & & & & & & b_5 & b_4 & & & \\
 & & & & & & & & & b_5 & & &
\end{array}\right]
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix}
+
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ \hline 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$A = B \times Q + R$ as matrix-vector products

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
=
\begin{bmatrix}
b_0 & & & & \\
b_1 & b_0 & & & \\
b_2 & b_1 & b_0 & & \\
b_3 & b_2 & b_1 & b_0 & \\
b_4 & b_3 & b_2 & b_1 & b_0
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_8 \\ q_9 \end{bmatrix}
+
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
$$

$$
\begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
=
\begin{bmatrix}
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 \\
 & & & & & & b_5 & b_4 & b_3 & b_2 \\
 & & & & & & & b_5 & b_4 & b_3 \\
 & & & 0 & & & & & b_5 & b_4 \\
 & & & & & & & & & b_5
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix}
+
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$A = B \times Q + R$ as matrix-vector products

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
=
\begin{bmatrix}
b_0 & & & & \\
b_1 & b_0 & & & \\
b_2 & b_1 & b_0 & & \\
b_3 & b_2 & b_1 & b_0 & \\
b_4 & b_3 & b_2 & b_1 & b_0
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}
+
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
$$

$$
\begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
=
\begin{bmatrix}
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 \\
 & & & & & & b_5 & b_4 & b_3 & b_2 \\
 & & & & & & & b_5 & b_4 & b_3 \\
 & & & 0 & & & & & b_5 & b_4 \\
 & & & & & & & & & b_5
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix}
+
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$A = B \times Q + R$ as matrix-vector products

$$
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
=
\begin{bmatrix}
b_0 & & & & \\
b_1 & b_0 & & & \\
b_2 & b_1 & b_0 & & \\
b_3 & b_2 & b_1 & b_0 & \\
b_4 & b_3 & b_2 & b_1 & b_0
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}
+
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
$$

$$
\begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
=
\begin{bmatrix}
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 \\
 & & & & & & b_5 & b_4 & b_3 & b_2 \\
 & & & & & & & b_5 & b_4 & b_3 \\
 & & & 0 & & & & & b_5 & b_4 \\
 & & & & & & & & & b_5
\end{bmatrix}
\times
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix}
$$

A matrix formula for the remainder

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} - \begin{bmatrix} b_0 & & & & \\ b_1 & b_0 & & & \\ b_2 & b_1 & b_0 & & \\ b_3 & b_2 & b_1 & b_0 & \\ b_4 & b_3 & b_2 & b_1 & b_0 \end{bmatrix} \times \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix} = \begin{bmatrix} b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\ & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\ & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & \\ & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & \\ & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ & & & & & b_5 & b_4 & b_3 & b_2 & b_1 \\ & & & & & & b_5 & b_4 & b_3 & b_2 \\ & & & & & & & b_5 & b_4 & b_3 \\ & & & & & & & & b_5 & b_4 \\ & & & & & & & & & b_5 \end{bmatrix}^{-1} \times \begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}$$

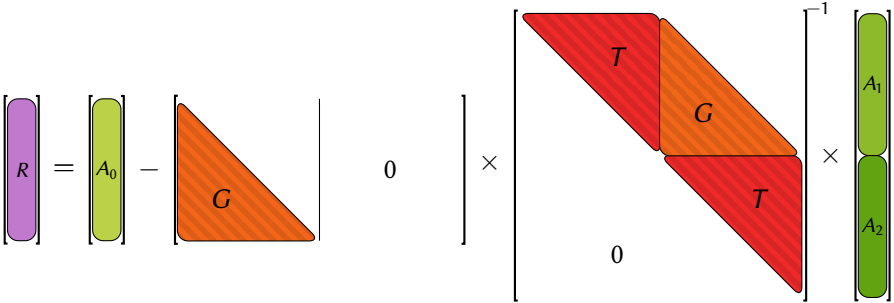
A matrix formula for the remainder

$$
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
=
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
-
\begin{bmatrix}
b_0 & & & & \\
b_1 & b_0 & & & \\
b_2 & b_1 & b_0 & & \\
b_3 & b_2 & b_1 & b_0 & \\
b_4 & b_3 & b_2 & b_1 & b_0
\end{bmatrix}
\times
\left[\begin{array}{ccccc|c}
1 & & & & & \\
 & 1 & & & & \\
 & & 1 & & & \\
 & & & 1 & & \\
 & & & & 1 & \\
\hline
 & & & & & 0
\end{array}\right]
\times
\begin{bmatrix}
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & & & \\
 & & & & & & & b_5 & b_4 & b_3 & b_2 & & & \\
 & & & & & & & & b_5 & b_4 & b_3 & & & \\
 & & & & & & & & & b_5 & b_4 & & & \\
 & & & & & & & & & & b_5 & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & &
\end{bmatrix}^{-1}
\times
\begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
$$

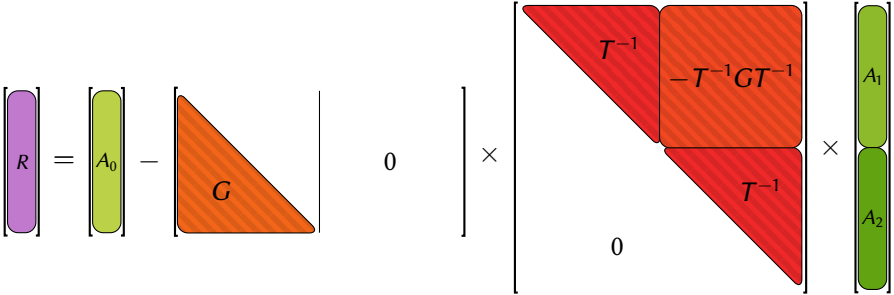
A matrix formula for the remainder

$$
\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
=
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
-
\left[\begin{array}{c|c}
\begin{matrix} b_0 & & & & \\ b_1 & b_0 & & & \\ b_2 & b_1 & b_0 & & \\ b_3 & b_2 & b_1 & b_0 & \\ b_4 & b_3 & b_2 & b_1 & b_0 \end{matrix} & 0
\end{array}\right]
\times
\begin{bmatrix}
b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & & \\
 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & & \\
 & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & & \\
 & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & & \\
 & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & & \\
 & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 & & \\
 & & & & & & b_5 & b_4 & b_3 & b_2 & b_1 & & \\
 & & & & & & & b_5 & b_4 & b_3 & b_2 & & \\
 & & & & & & & & b_5 & b_4 & b_3 & & \\
 & & & & & & & & & b_5 & b_4 & & \\
 & & & & & & & & & & b_5 & &
\end{bmatrix}^{-1}
\times
\begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}
$$

A matrix formula for the remainder



A matrix formula for the remainder



A matrix formula for the remainder

$$\begin{bmatrix} R \end{bmatrix} = \begin{bmatrix} A_0 \end{bmatrix} - \begin{bmatrix} GT^{-1} & -(GT^{-1})^2 \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

A matrix formula for the remainder

$$\begin{bmatrix} R \end{bmatrix} = \begin{bmatrix} A_0 \end{bmatrix} - \begin{bmatrix} GT^{-1} & -(GT^{-1})^2 \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

Theorem

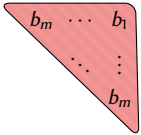
$$\begin{bmatrix} R \end{bmatrix} = \sum_{i=0}^{\mu} \begin{bmatrix} -GT^{-1} \end{bmatrix}^i \cdot \begin{bmatrix} A_i \end{bmatrix}$$

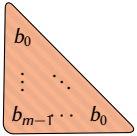
$\rightarrow$ Horner-like computation

A "not-in-place" algorithm

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m **(model ro/rw)**

Output: $R = A \bmod B$ of degree $< m$

0. $\mu = \lceil (n - m + 1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$ 

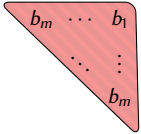
; $G =$ 

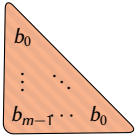
1. $R := A_\mu$
2. for $i = \mu - 1$ to 0:
3. $\quad Q := T^{-1} \cdot R$
4. $\quad R := -G \cdot Q$
5. $\quad R \mathrel{+}= A_i$
6. return R

A "not-in-place" algorithm

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m **(model ro/rw)**

Output: $R = A \bmod B$ of degree $< m$

0. $\mu = \lceil (n-m+1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$ 

; $G =$ 

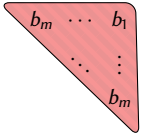
1. $R := A_\mu$
2. for $i = \mu - 1$ to 0:
3. $\quad Q := T^{-1} \cdot R$
4. $\quad R := -G \cdot Q$
5. $\quad R \mathrel{+}= A_i$
6. return R

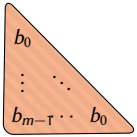
- Time: $O(n/m \cdot \mathsf{M}(m))$
- Space: $m + O(1)$ *storage of* Q
- Remark: similar result in [Giorgi-G.-Roche'20]

An *in-place* algorithm

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m **A: read-only** / B: read-write

Output: $R = A \bmod B$ of degree $< m$

0. $\mu = \lceil (n-m+1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$ 

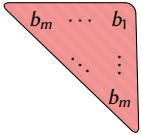
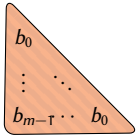
   ; $G =$ 

1. $R := A_\mu$
2. for $i = \mu - 1$ to 0 :
3. $\quad R *= T^{-1}$
4. $\quad R *= -G$
5. $\quad R += A_i$
6. return R

An *in-place* algorithm

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m **A: read-only** / B: read-write

Output: $R = A \bmod B$ of degree $< m$

0. $\mu = \lceil (n-m+1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$  ; $G =$ 
1. $R := A_\mu$
2. for $i = \mu - 1$ to 0:
3. $\quad R \mathrel{*}= T^{-1}$
4. $\quad R \mathrel{*}= -G$
5. $\quad R \mathrel{+}= A_i$
6. return R

- Time: $O(n/m \cdot \mathsf{M}(m) \log m)^*$
- Space: $O(1)^\dagger$

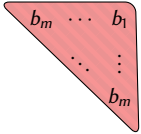
* $O(\mathsf{M}(m))$ if $\mathsf{M}(m) = \Omega(m^{1+\varepsilon})$

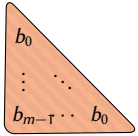
† $O(\log m)$ pointers for call stack

Over-place Euclidean division

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m — A, B: read-write

Output: $A \leftarrow R\ Q$ s.t. $A = BQ + R$ with R of degree $< m$

0. $\mu = \lceil (n - m + 1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$ 

; $G =$ 

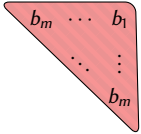
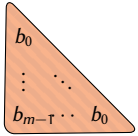
1. for $i = \mu - 1$ to 0:
2. $\quad A_{i+1} *= T^{-1}$
3. $\quad A_i -= G \cdot A_{i+1}$
4. return $R = A_0$ and $Q = A_1 \cdots A_\mu$

simplification: m divides $n - 1$

Over-place Euclidean division

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m — A, B: read-write

Output: $A \leftarrow R$ Q s.t. $A = BQ + R$ with R of degree $< m$

0. $\mu = \lceil (n - m + 1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$  ; $G =$ 
1. for $i = \mu - 1$ to 0:
2. $\quad A_{i+1} *= T^{-1}$
3. $\quad A_i \mathrel{-}= G \cdot A_{i+1}$
4. return $R = A_0$ and $Q = A_1 \cdots A_\mu$

simplification: m divides $n - 1$

- Time: $O(n/m \cdot \mathrm{M}(m) \log m)^*$
- Space: $O(1)^\dagger$
- Remark: the algorithm is *reversible*

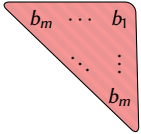
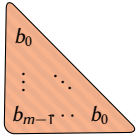
* $O(n/m \cdot \mathrm{M}(m))$ if $\mathrm{M}(m) = \Omega(m^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Over-place Euclidean division

Inputs: $A \in \mathbb{F}[x]$ of degree n, $B \in \mathbb{F}[x]$ of degree m — A, B: read-write

Output: $A \leftarrow R\ Q$ s.t. $A = BQ + R$ with R of degree $< m$

0. $\mu = \lceil (n - m + 1)/m \rceil$
   $A = A_0 A_1 \cdots A_\mu$ (blocks of size m); $T =$  ; $G =$ 
1. for $i = \mu - 1$ to 0:
2. $\quad A_{i+1} *= T^{-1}$
3. $\quad A_i \mathrel{-}= G \cdot A_{i+1}$
4. return $R = A_0$ and $Q = A_1 \cdots A_\mu$

simplification: m divides $n - 1$

- Time: $O(n/m \cdot \mathrm{M}(m) \log m)^*$
- Space: $O(1)^\dagger$
- Remark: the algorithm is *reversible*

$C \mathrel{+}= A \bmod B$

1. Replace A by $R|Q$
2. $C \mathrel{+}= R$
3. Restore A

* $O(n/m \cdot \mathrm{M}(m))$ if $\mathrm{M}(m) = \Omega(m^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Accumulated modular multiplication

Inputs: $A, B, R \in \mathbb{F}[x]$ of degrees $< m$; $C \in \mathbb{F}[x]$ of degree m

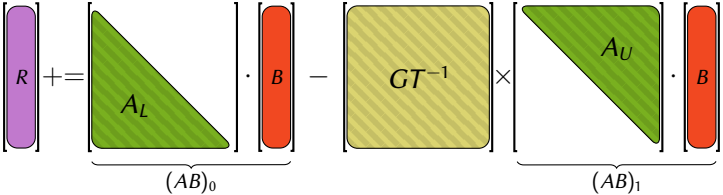
Output: $R \mathrel{+}= A \cdot B \bmod C$

Accumulated modular multiplication

Inputs: $A, B, R \in \mathbb{F}[x]$ of degrees $< m$; $C \in \mathbb{F}[x]$ of degree m

Output: $R \mathrel{+}= A \cdot B \bmod C$

An adapted remainder formula



Algorithm

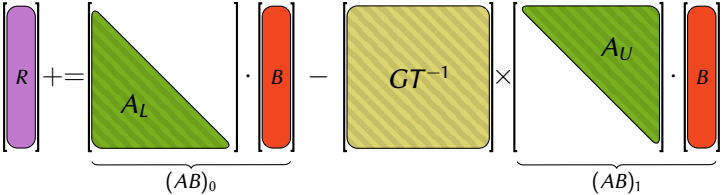
1. $R \mathrel{+}= A_L \cdot B$
2. $B \mathrel{*}= A_U$
3. $B \mathrel{*}= T^{-1}$
4. $R \mathrel{-}= G \cdot B$
5. $B \mathrel{*}= T$ *undo* 3.
6. $B \mathrel{*}= A_U^{-1}$ *undo* 2.

Accumulated modular multiplication

Inputs: $A, B, R \in \mathbb{F}[x]$ of degrees $< m$; $C \in \mathbb{F}[x]$ of degree m

Output: $R \mathrel{+}= A \cdot B \bmod C$

An adapted remainder formula



Algorithm

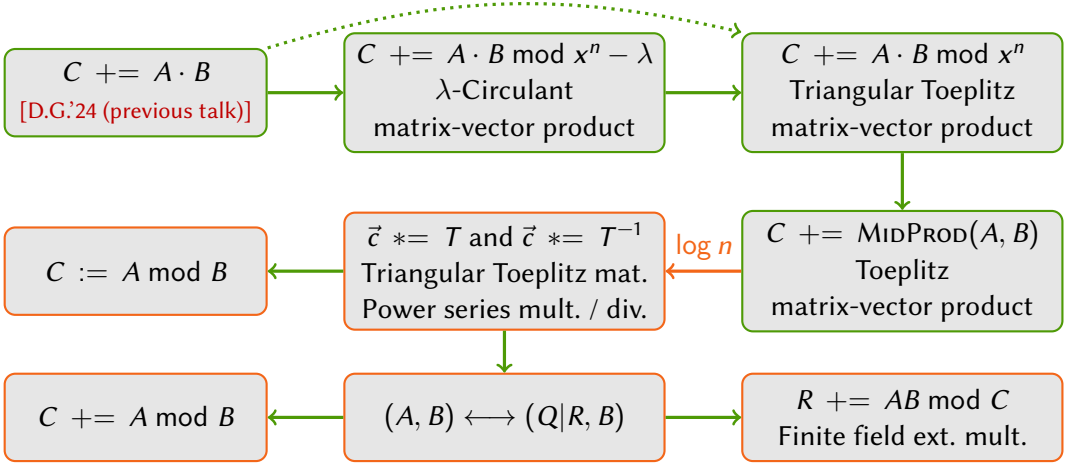
1. $R \mathrel{+}= A_L \cdot B$
2. $B \mathrel{*}= A_U$
3. $B \mathrel{*}= T^{-1}$
4. $R \mathrel{-}= G \cdot B$
5. $B \mathrel{*}= T$ *undo* 3.
6. $B \mathrel{*}= A_U^{-1}$ *undo* 2.

- Time: $O(\mathsf{M}(m) \log m)^*$
- Space: $O(1)^\dagger$
- Generalization to all degrees

* $O(\mathsf{M}(m))$ if $\mathsf{M}(m) = \Omega(m^{1+\varepsilon})$

† $O(\log m)$ pointers for call stack

Summary of reductions



Open problems

Non-optimal results

- $C := A \bmod B, \ldots$ in $O(\mathsf{M}(n))$
- Avoid the call stack
- Bypass the over-place computations

Other models and lower bounds

- What can be done in the model `ro/rw`?
- Time-space lower bounds in models `ro/rw` and `rw/rw`?
- Over-place computation: $\boxed{A}\boxed{B} \rightarrow \boxed{A \cdot B}$ [Roche'09]

Practical efficiency

- No implementation yet
- Encouraging results in [D.G.'24 (previous talk)]

Open problems

Non-optimal results

- $C := A \bmod B, \ldots$ in $O(\mathsf{M}(n))$
- Avoid the call stack
- Bypass the over-place computations

Other models and lower bounds

- What can be done in the model `ro/rw`?
- Time-space lower bounds in models `ro/rw` and `rw/rw`?
- Over-place computation: $\boxed{A}\boxed{B} \to \boxed{A \cdot B}$ [Roche'09]

Practical efficiency

- No implementation yet
- Encouraging results in [D.G.'24 (previous talk)]

Thank you!