## Paul Vrbik University of Western Ontario

December 10, 2009

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<ロ> (四) (四) (注) (注) (注) (三)

L The Fast Fourier Transform (FFT)

## THE FAST FOURIER TRANSFORM

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

Let:

 R be an *effective ring* (meaning that there are effective procedures for computing the sum, difference and product of two elements of R[x]),

• 
$$\omega \in \mathcal{R}[x]$$
 be a primitive  $n = 2^p$ -th root of unity (i.e.  $\omega^{n/2} = -1$ ).

## Definition (FFT - technical)

The discrete Fast Fourier Transform (FFT) maps

$$(a_0,\ldots,a_{n-1})\in\mathcal{R}^n\stackrel{\mathsf{FFT}}{\longrightarrow}(\hat{a}_0,\ldots,\hat{a}_{n-1})\in\mathcal{R}^n$$

with

$$\hat{a}_i = \sum_{j=0}^{n-1} a_j \omega^{ij}.$$

Paul Vrbik University of Western Ontario

The Truncated Fourier Transform

L The Fast Fourier Transform (FFT)

## Definition (FFT - useful)

Let 
$$A = a_0 + a_x x + \dots + a_{n-1} x^{n-1} \in \mathcal{R}[x]$$
 then FFT maps  

$$A(x) \xrightarrow{\text{FFT}} (A(\omega^0), A(\omega^1), \dots, A(\omega^{n-1}))$$
or  $\hat{a}_i = A(\omega^i)$ .

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへで

L The Fast Fourier Transform (FFT)

The FFTcan be computed efficiently using binary splitting: Write

$$(a_0,\ldots,a_{n-1})=\left(b_0,c_0,\ldots,b_{\frac{n}{2}-1},c_{\frac{n}{2}-1}\right)$$

and do

$$\mathsf{FFT}_{\omega^2}(b_0, \dots, b_{n/2-1}) = (\hat{b}_0, \dots, \hat{b}_{n/2} - 1)$$
$$\mathsf{FFT}_{\omega^2}(c_0, \dots, c_{n/2-1}) = (\hat{c}_0, \dots, \hat{c}_{n/2} - 1).$$

Then we have

$$\mathsf{FFT}_{\omega}(a_0, \dots, a_{n-1}) = (\hat{b}_0 + \hat{c}_0, \dots, \hat{b}_{n/2-1} + \hat{c}_{n/2-1}\omega^{n/2-1} \\ \hat{b}_0 - \hat{c}_0, \dots, \hat{b}_{n/2-1} - \hat{c}_{n/2-1}\omega^{n/2-1}).$$

◆□> ◆□> ◆目> ◆目> ◆目> 目 のへで

The Fast Fourier Transform (FFT)

In practice it is more efficient to implement an *in-place* algorithm rather than a recursive one.

First we define the bitwise mirror of *i* at length *p*, denoted  $[i]_p$ .

## Example

$$[3]_5 = 24$$
 because  $3 = 00011_2$  whose reverse is  $11000_2 = 24$ .

 $[11]_5 = 26$  because  $11 = 01011_2$  whose reverse is  $11010_2 = 26$ .

(ロ) (同) (E) (E) (E)

L The Fast Fourier Transform (FFT)

At step 
$$s \in \{1, \ldots, p\}$$
 with  $m_s = 2^{p-s}$  we set  

$$\begin{bmatrix} x_{s,im_s+j} \\ x_{s,(i+1)m_s+j} \end{bmatrix} = \begin{bmatrix} 1 & \omega^{[i]_sm_s} \\ 1 & -\omega^{[i]_sm_s} \end{bmatrix} \begin{bmatrix} x_{s-1,im_s+j} \\ x_{s-1,(i+1)m_s+j} \end{bmatrix}.$$

Figure: Butterflies. The black dots correspond to the  $x_{s,i}$ . The top row corresponding to s = 0. In this case  $n = 16 = 2^4$ .

#### L The Fast Fourier Transform (FFT)



Figure: Butterflies. The black dots correspond to the  $x_{s,i}$ . The top row corresponding to s = 0. In this case  $n = 16 = 2^4$ .

The Fast Fourier Transform (FFT)

For every step *s* we calculate  $x_s = (x_{0,0}, ..., x_{0,n-1})$  from  $x_{s-1}$  (note  $x_0 = (a_0, ..., a_{n-1})$ ).

At step  $s \in \{1, \dots, p\}$ , we set

$$\begin{bmatrix} x_{s,im_s+j} \\ x_{s,(i+1)m_s+j} \end{bmatrix} = \begin{bmatrix} 1 & \omega^{[i]_sm_s} \\ 1 & -\omega^{[i]_sm_s} \end{bmatrix} \begin{bmatrix} x_{s-1,im_s+j} \\ x_{s-1,(i+1)m_s+j} \end{bmatrix}$$

for all  $i \in \{0, 2, ..., n/m_s - 2\}$  and  $j \in \{0, ..., m_s - 1\}$ , where  $m_s = 2^{p-s}$ .

## Theorem

For  $i \in \{0, \dots, n/m_s - 1\}$  $x_{p,i} = \hat{a}_{[i]_p}$  $\hat{a}_i = x_{p,[i]_p}$ 

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

#### L The Fast Fourier Transform (FFT)



Figure: The "Regular" Fast Fourier Transform.

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□> ◆□> ◆目> ◆目> ◆目> 目 のへで

L The Fast Fourier Transform (FFT)

It is also straightforward to recover  $\boldsymbol{a}$  from  $\hat{\boldsymbol{a}}$ 

$$\mathsf{FFT}_{\omega^{-1}}(\hat{\mathbf{a}})_i = \mathsf{FFT}_{\omega^{-1}}(\mathsf{FFT}_{\omega}(\mathbf{a}))_i = \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} a_j \omega^{(i-k)j} = na_i$$

since

$$\sum_{j=0}^{n-1} \omega^{(i-k)j} = 0$$

whenever  $i \neq k$ . This yields a polynomial multiplication algorithm of time complexity  $O(n \log n)$  in  $\mathcal{R}[x]$ .

L The Truncated Fourier Transform (TFT)

## THE TRUNCATED FAST FOURIER TRANSFORM

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

L The Truncated Fourier Transform (TFT)

The algorithm for FFT requires that *n* is a power of two. If we want to multiply two polynomials  $A, B \in \mathcal{R}[x]$  such that  $\deg(AB) + 1 = n + \delta$  we would need to carry out the FFT at precision 2n.

### L The Truncated Fourier Transform (TFT)

| • | • | • | ٠ | ٠ | ٠ | ٠ | • | ٠ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ο | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

### L The Truncated Fourier Transform (TFT)

| • | ٠ | • | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | • | • | ۰ | • | 0 | 0 | • |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

## L The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

## -The Truncated Fourier Transform (TFT)



< □ > < □ > < □ > < □ > < □ > < Ξ > = Ξ

L The Truncated Fourier Transform (TFT)

The idea behind the TFT is to only do the work required to get the desired output (eliminating a factor of 2).

Let  $n = 2^p$ ,  $\ell < n$  (usually  $\ell \ge n/2$ ) and let  $\omega$  be a primitive *n*-th root of unity.

$$\mathsf{TFT}_{\omega}\left(A=a_{0}+\cdots+a_{\ell}x^{\ell-1}
ight)\xrightarrow{\mathsf{TFT}}\left(A(\omega^{[0]_{p}}),\ldots,A(\omega^{[\ell-1]_{p}})
ight).$$

<ロ> (四) (四) (三) (三) (三)

We use the in-place algorithm from the previous section in order to compute the TFT. The claim is that this means many of the  $x_{s,i}$  can be skipped. Indeed, at stage s, it suffices to compute the vector  $(x_{s,0}, \ldots, x_{s,\lfloor (\ell-1)/m_s \rfloor + 1)m_s - 1})$ 

## Theorem

The TFT of an  $\ell$ -tuple w.r.t.  $\omega$  can be computed using at most  $\ell p + n$  additions and  $\lfloor (lp + n)/2 \rfloor$  multiplications with powers of  $\omega$ .

<ロ> (四) (四) (三) (三) (三) 三

L The Truncated Fourier Transform (TFT)

## THE INVERSE TRUNCATED FOURIER TRANSFORM

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

Unfortunately (quite unfortunately), the inverse TFTcannot be computed using a similar formula. Simply put, we are missing information and have to account for this.

We will use the fact that whenever one value among

and one value among

$$X_{s,(i+1)m_s+j}, X_{s-1,(i+1)m_s+j}$$

are known then we can deduce the other values.

(ロ) (同) (E) (E) (E)

#### The Truncated Fourier Transform (TFT)



Figure: You can deduce any part of the butterfly from two known parts.

$$\begin{bmatrix} x_{s,im_s+j} \\ x_{s,(i+1)m_s+j} \end{bmatrix} = \begin{bmatrix} 1 & \omega^{[i]_sm_s} \\ 1 & -\omega^{[i]_sm_s} \end{bmatrix} \begin{bmatrix} x_{s-1,im_s+j} \\ x_{s-1,(i+1)m_s+j} \end{bmatrix}$$

\_\_\_\_

프 🕨 🗉 프

### L The Truncated Fourier Transform (TFT)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure: The computations in the boxes are self contained.

(ロ) (回) (E) (E) (E) (O)

#### The Truncated Fourier Transform (TFT)



Figure: The computations in the boxes are self contained.

◆□▶ ◆□▶ ◆目▶ ◆目▶ ●目 - のへで

## The Truncated Fourier Transform (TFT)



< 🗇 >

≪ 臣 ≯

< ≣⇒

æ

### L The Truncated Fourier Transform (TFT)

| ( | o c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | • | • | • | • | • |
|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ( | o c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ( | o c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|   | o c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| • | • • | • | • | • | • | • | ٠ | • | ٠ | • | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

### L The Truncated Fourier Transform (TFT)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ٠ | ۲ | ۲ | ۲ | ۲ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| • | ٠ | ٠ | ٠ | • | ٠ | ٠ | ٠ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | • | ٠ | • | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

\_\_\_\_

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

< 17 >

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

\_\_\_\_

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

\_\_\_\_

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

\_\_\_\_

### L The Truncated Fourier Transform (TFT)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ο |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| • | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | 0 | 0 | 0 | ۰ | • | • | • | • |
| ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ٠ | 0 | 0 | 0 | 0 |
| ο | 0 | 0 | ο | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | • | ٠ | • | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

### L The Truncated Fourier Transform (TFT)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| • | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | 0 | 0 | 0 | ۰ | • | • | ۰ | ۰ |
| ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | • | ٠ | ٠ | ٠ | 0 | 0 | 0 | 0 |
| ο | 0 | 0 | 0 | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

## The Truncated Fourier Transform (TFT)



Paul Vrbik University of Western Ontario The Truncated Fourier Transform

< 🗇 >

### L The Truncated Fourier Transform (TFT)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ο |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| • | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | • | ٠ | • | • | ٠ | ٠ | • | ٠ |
| ο | 0 | 0 | 0 | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ο | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

<□> <□> <□> <=> <=> <=> <=> <=> <=> <<</p>

### L The Truncated Fourier Transform (TFT)

| • | • | • | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | ٠ | • | • | ٠ | ٠ | ٠ | • |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の < @

## Theorem

The  $\ell$ -tuple  $(a_0, \ldots, a_{\ell-1})$  can be recovered from its TFTwith respect to  $\omega$  using at most  $\ell p + n$  shifted additions (or subtractions) and  $\lfloor (\ell p + n)/2 \rfloor$  multiplications with powers of  $\omega$ .

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

(ロ) (同) (E) (E) (E)

L The Truncated Fourier Transform (TFT)

"Future Work"

- Paper for Eric.
- Finish my MAPLE implementation (almost there).

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

イロト イポト イヨト イヨト 三日

L The Truncated Fourier Transform (TFT)

## Done.

Paul Vrbik University of Western Ontario The Truncated Fourier Transform

(ロ) (回) (E) (E) (E) (O)