Fast Fourier Transform

Part: Informal Development of Fast Fourier Transform

http://numericalmethods.eng.usf.edu

For more details on this topic

- Go to http://numericalmethods.eng.usf.edu
- Click on Keyword
- Click on Fast Fourier Transform

You are free

- to **Share** – to copy, distribute, display and perform the work
- to **Remix** – to make derivative works

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

Under the following conditions

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** — You may not use this work for commercial purposes.
- **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Lecture # 11

Chapter 11.05: Informal Development of Fast Fourier Transform

Major: All Engineering Majors

Authors: Duc Nguyen

http://numericalmethods.eng.usf.edu

Numerical Methods for STEM undergraduates

Informal Development of Fast Fourier Transform

Recall the DFT pairs of Equations (20) and (21) of Chapter 11.04 and swapping the indexes n, k one obtains

$$\tilde{C}_n = \sum_{k=0}^{N-1} f(k) e^{-in\left(w_0 = \frac{2\pi}{N}\right)k} \tag{1}$$

$$f(k) = \left(\frac{1}{N}\right) \sum_{n=0}^{N-1} \tilde{C}_n e^{in\left(w_0 = \frac{2\pi}{N}\right)k} \tag{2}$$

where $n, k = 0, 1, 2, 3, \ldots, N-1$ (3)

Let $E = e^{-i\frac{2\pi}{N}} \quad \left(hence \;\; E^N = e^{-i2\pi} = 1 \right)$ (4)

Informal Development cont.

Then Eq. (1) and Eq. (2) become

$$\tilde{C}_n = \tilde{C}(n) = \sum_{k=0}^{N-1} f(k) E^{nk} \tag{5}$$

$$f(k) = \left(\frac{1}{N}\right) \sum_{n=0}^{N-1} \tilde{C}_n E^{-nk}$$

Assuming $N = 4 = 2^{(r=2)}$, then

$$\left(\frac{1}{N}\right)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & E^{-1} & E^{-2} & E^{-3} \\ 1 & E^{-2} & E^{-4} & E^{-6} \\ 1 & E^{-3} & E^{-6} & E^{-9} \end{bmatrix} \begin{Bmatrix} \tilde{C}(0) \\ \tilde{C}(1) \\ \tilde{C}(2) \\ \tilde{C}(3) \end{Bmatrix} = \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \tag{5A}$$

Informal Development cont.

To obtain the above unknown vector $\left\{\tilde{C}\right\}$ for a given vector $\left\{f\right\}$, the coefficient matrix can be easily converted as

$$\left[\left(\frac{1}{N}\right)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & E^{-1} & E^{-2} & E^{-3} \\ 1 & E^{-2} & E^{-4} & E^{-6} \\ 1 & E^{-3} & E^{-6} & E^{-9} \end{bmatrix}\right]^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & E^{1} & E^{2} & E^{3} \\ 1 & E^{2} & E^{4} & E^{6} \\ 1 & E^{3} & E^{6} & E^{9} \end{bmatrix}$$

Informal Development cont.

Hence, the unknown vector $\{\tilde{C}\}$ can be computed as with matrix vector operations, as following

$$\begin{Bmatrix} \tilde{C}(0) \\ \tilde{C}(1) \\ \tilde{C}(2) \\ \tilde{C}(3) \end{Bmatrix} = \begin{bmatrix} E^{(0)(0)} & E^{(0)(1)} & E^{(0)(2)} & E^{(0)(3)} \\ E^{(1)(0)} & E^{(1)(1)} & E^{(1)(2)} & E^{(1)(3)} \\ E^{(2)(0)} & E^{(2)(1)} & E^{(2)(2)} & E^{(2)(3)} \\ E^{(3)(0)} & E^{(3)(1)} & E^{(3)(2)} & E^{(3)(3)} \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \qquad (6)$$

Informal Development cont.

$$\begin{Bmatrix} \tilde{C}(0) \\ \tilde{C}(1) \\ \tilde{C}(2) \\ \tilde{C}(3) \end{Bmatrix} = \begin{bmatrix} E^0 & E^0 & E^0 & E^0 \\ E^0 & E^1 & E^2 & E^3 \\ E^0 & E^2 & E^4 & E^6 \\ E^0 & E^3 & E^6 & E^9 \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \qquad (7)$$

For $N = 4, k = 3$ and $n = 2$, then:

$$E^{nk} = E^6 = \left[E^{(N=4)}\right]E^2 = \left(e^{\frac{-i2\pi}{N}}\right)^N E^2 = \left[e^{-i2\pi}\right]E^2 = E^2$$

Informal Development cont.

Thus, in general (for $nk \geq N$)

$$E^{nk} = E^{U} \text{ where } U = \text{mod}(nk, N) \tag{8}$$

$$U = remainder\left(\frac{nk}{N}\right)$$

Informal Development cont.

Remarks:

a) Matrix times vector, shown in Eq. (7), will require 16 (or N^2) complex multiplications and 12 (or $N(N-1)$) complex additions.

b) Use of Eq. (8) will help to reduce the number of operation counts, as explained in the next section.

THE END

http://numericalmethods.eng.usf.edu

Acknowledgement

This instructional power point brought to you by Numerical Methods for STEM undergraduate

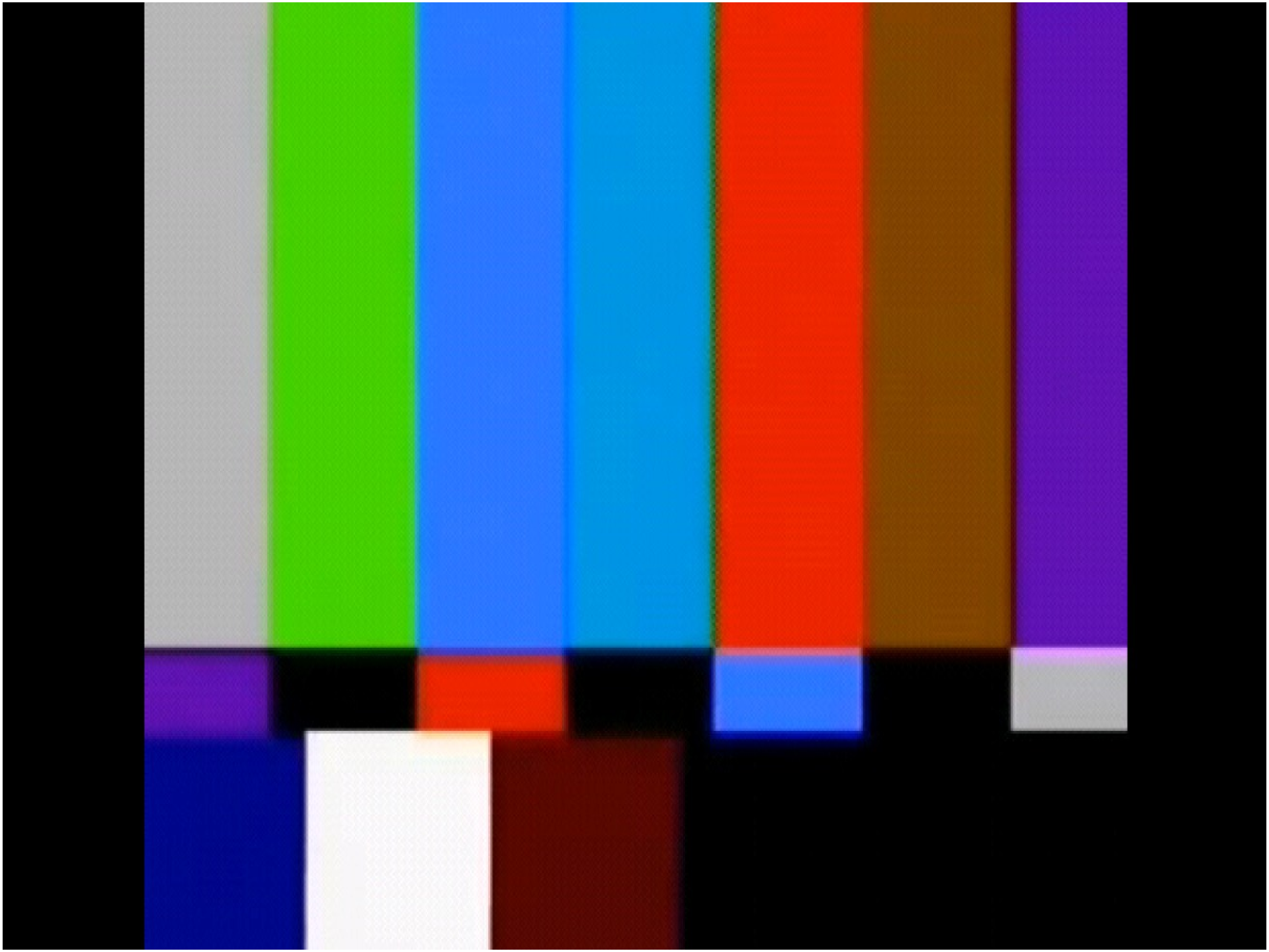
http://numericalmethods.eng.usf.edu

Committed to bringing numerical methods to the undergraduate

For instructional videos on other topics, go to

http://numericalmethods.eng.usf.edu/videos/

This material is based upon work supported by the National Science Foundation under Grant # 0717624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Fast Fourier Transform

Part: Factorized Matrix and Further Operation Count

http://numericalmethods.eng.usf.edu

For more details on this topic

- Go to http://numericalmethods.eng.usf.edu
- Click on Keyword
- Click on Fast Fourier Transform

You are free

- to **Share** – to copy, distribute, display and perform the work
- to **Remix** – to make derivative works

Under the following conditions

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** — You may not use this work for commercial purposes.
- **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Lecture # 12

Chapter 11.05: Factorized Matrix and Further Operation Count (Contd.)

Equation (7) can be factorized as

$$\begin{Bmatrix} \tilde{C}(0) \\ \tilde{C}(2) \\ \tilde{C}(1) \\ \tilde{C}(3) \end{Bmatrix} = \begin{bmatrix} 1 & E^0 & 0 & 0 \\ 1 & E^2 & 0 & 0 \\ 0 & 0 & 1 & E^1 \\ 0 & 0 & 1 & E^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & E^0 & 0 \\ 0 & 1 & 0 & E^0 \\ 1 & 0 & E^2 & 0 \\ 0 & 1 & 0 & E^2 \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \qquad (9)$$

Let's define the following "inner – product"

$$\begin{Bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & E^0 & 0 \\ 0 & 1 & 0 & E^0 \\ 1 & 0 & E^2 & 0 \\ 0 & 1 & 0 & E^2 \end{bmatrix} \begin{Bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{Bmatrix} \qquad (10)$$

Factorized Matrix cont.

From Eq. (9) and (10) we obtain

$$f_1(0) = f(0) + E^0 f(2) \qquad (11A)$$

$$f_1(1) = f(1) + E^0 f(3) \qquad (11B)$$

$$f_1(2) = f(0) + E^2 f(2)$$

$$= f(0) - E^0 f(2) \qquad (11C)$$

with $$E^2 = e^{-i\frac{2\pi}{4}*2} = e^{-i\pi} = -1 = -E^0$$

$$f_1(3) = f(1) + E^2 f(3)$$

$$= f(1) - E^0 f(3) \qquad (11D)$$

Factorized Matrix cont.

Equations(11A through 11D) for the “inner” matrix times vector requires 2 complex multiplications and 4 complex additions.

Factorized Matrix cont.

Finally, performing the "outer" product (matrix times vector) on the RHS of Equation(9), one obtains

$$\begin{Bmatrix} \tilde{C}(0) \\ \tilde{C}(2) \\ \tilde{C}(1) \\ \tilde{C}(3) \end{Bmatrix} = \begin{Bmatrix} f_2(0) \\ f_2(1) \\ f_2(2) \\ f_2(3) \end{Bmatrix} = \begin{bmatrix} 1 & E^0 & 0 & 0 \\ 1 & E^2 & 0 & 0 \\ 0 & 0 & 1 & E^1 \\ 0 & 0 & 1 & E^3 \end{bmatrix} \begin{Bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{Bmatrix} \quad (12)$$

Factorized Matrix cont.

$$f_2(0) = f_1(0) + E^0 f_1(1) \qquad (13A)$$

$$f_2(1) = f_1(0) + E^2 f_1(1) = f_1(0) - E^0 f_1(1) \qquad (13B)$$

$$f_2(2) = f_1(2) + E^1 f_1(3) \qquad (13C)$$

$$f_2(3) = f_1(2) + E^3 f_1(3) = f_1(2) + E^2 E^1 f_1(3)$$

$$= f_1(2) - E^1 f_1(3) \qquad (13D)$$

Factorized Matrix cont.

Again, Eqs (13A-13D) requires 2 complex multiplications And 4 complex additions. Thus, the complete RHS of Eq. (9) Can be computed by only 4 complex multiplications (or $N\frac{r}{2} = 4\frac{2}{2}$) and 8 complex additions (or $Nr = 4 * 2$),

where $N = 2^r$.

Since computational time is mainly controlled by the number of multiplications, implementing Eq. (9) will significantly reduce the number of multiplication operations, as compared to a direct matrix times vector operations. (as shown in Eq. (7)).

Factorized Matrix cont.

For a large number of data points,

$$Ratio = \frac{N^2}{\left(\frac{Nr}{2}\right)} = \left(\frac{2N}{r}\right) \qquad (14)$$

For $N = 2048 = 2^{(r=11)}$,Equation (14) gives:

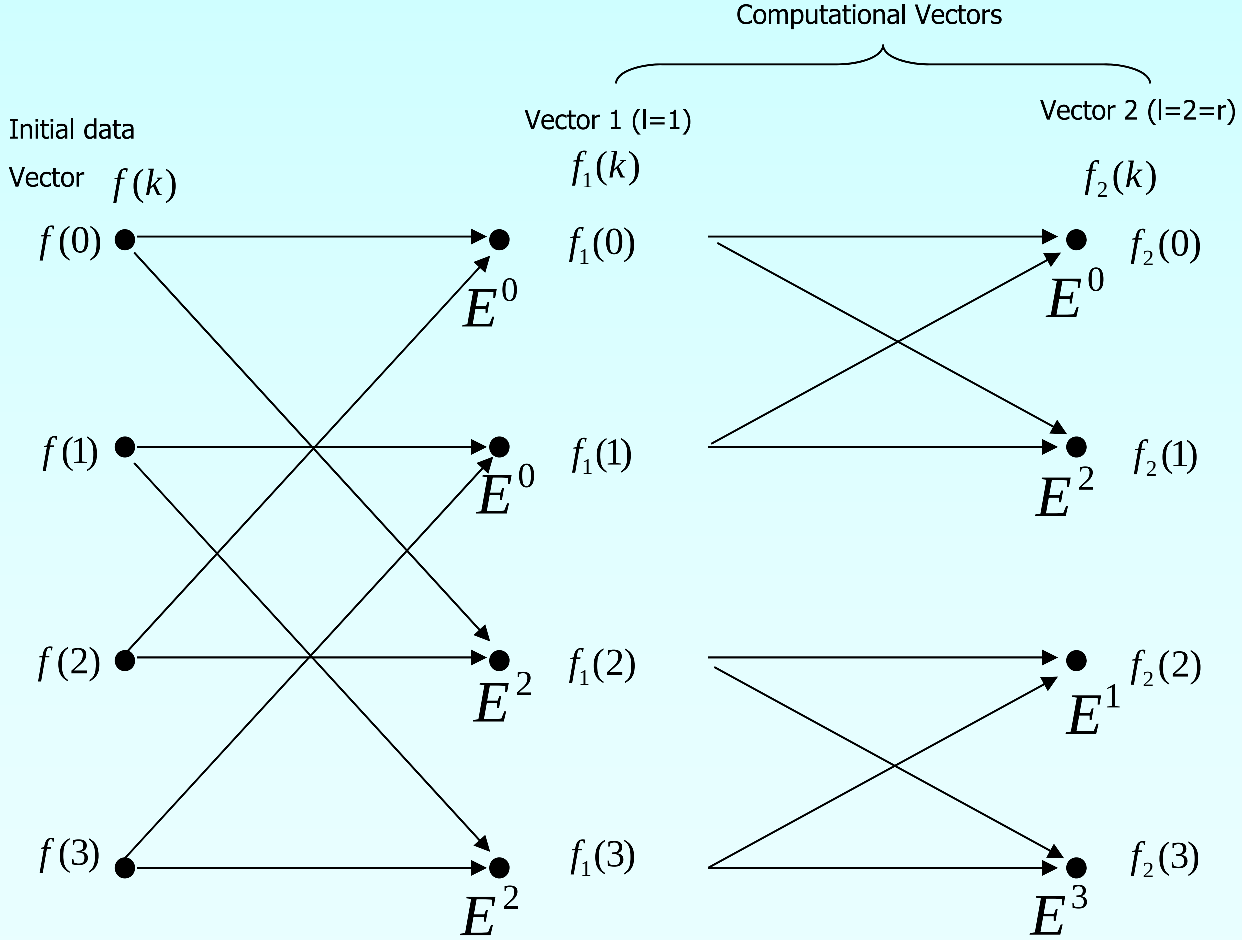
$$Ratio = \frac{2(2048)}{11} = 372.36$$

This implies that the number of complex multiplications involved in Eq. (9) is about 372 times less than the one involved in Eq. (7).

Graphical Flow of Eq. 9

- Consider the case $N = 2^r = 2^2 = 4$

Figure 1. Graphical form of FFT (Eq. 9) for the case $N = 2^r = 2^2 = 4$



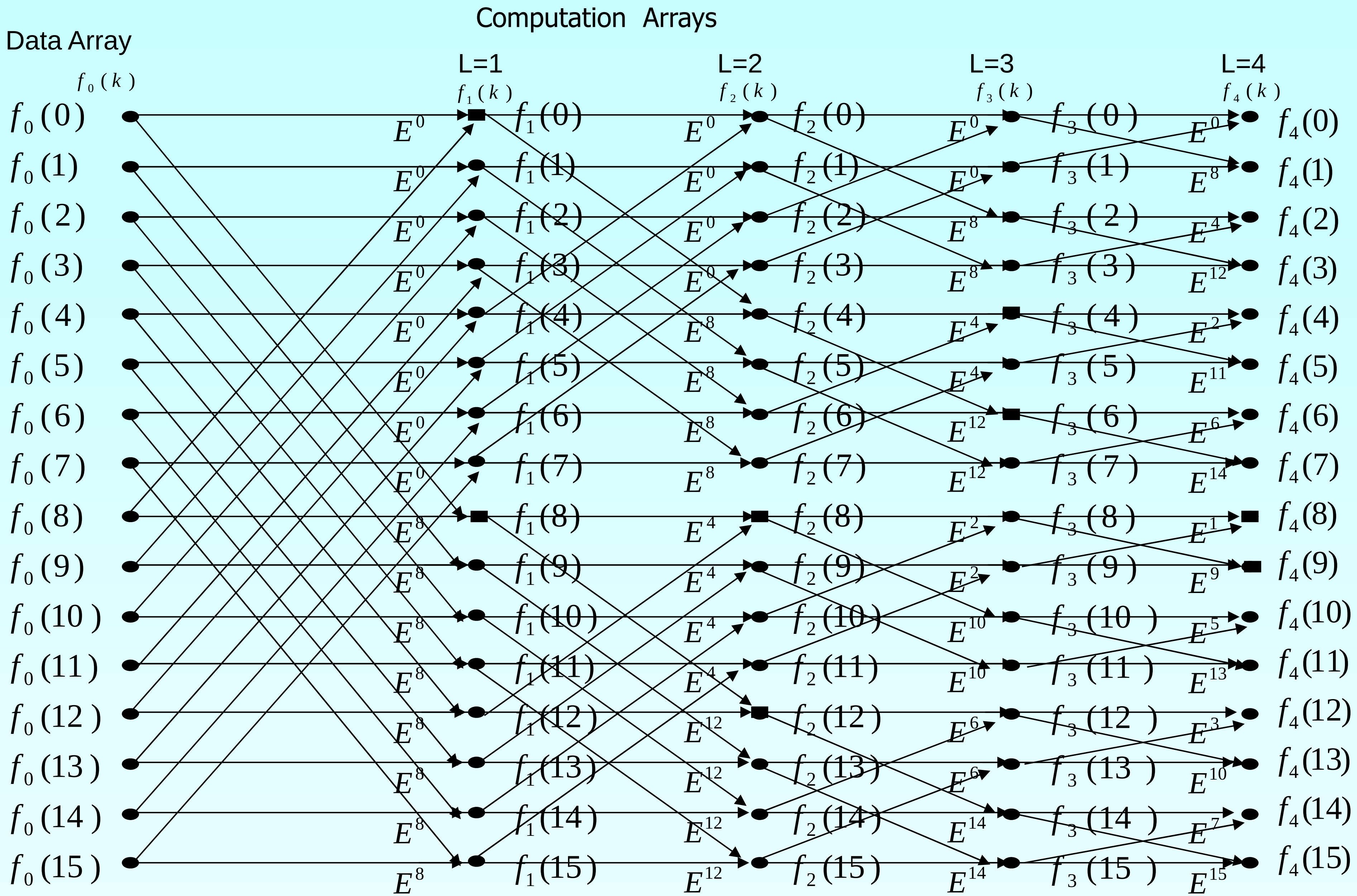


Figure 2. Graphical Form of FFT (Eq. 9) for the case

$$N = 2^r = 2^4 = 16$$

THE END

http://numericalmethods.eng.usf.edu

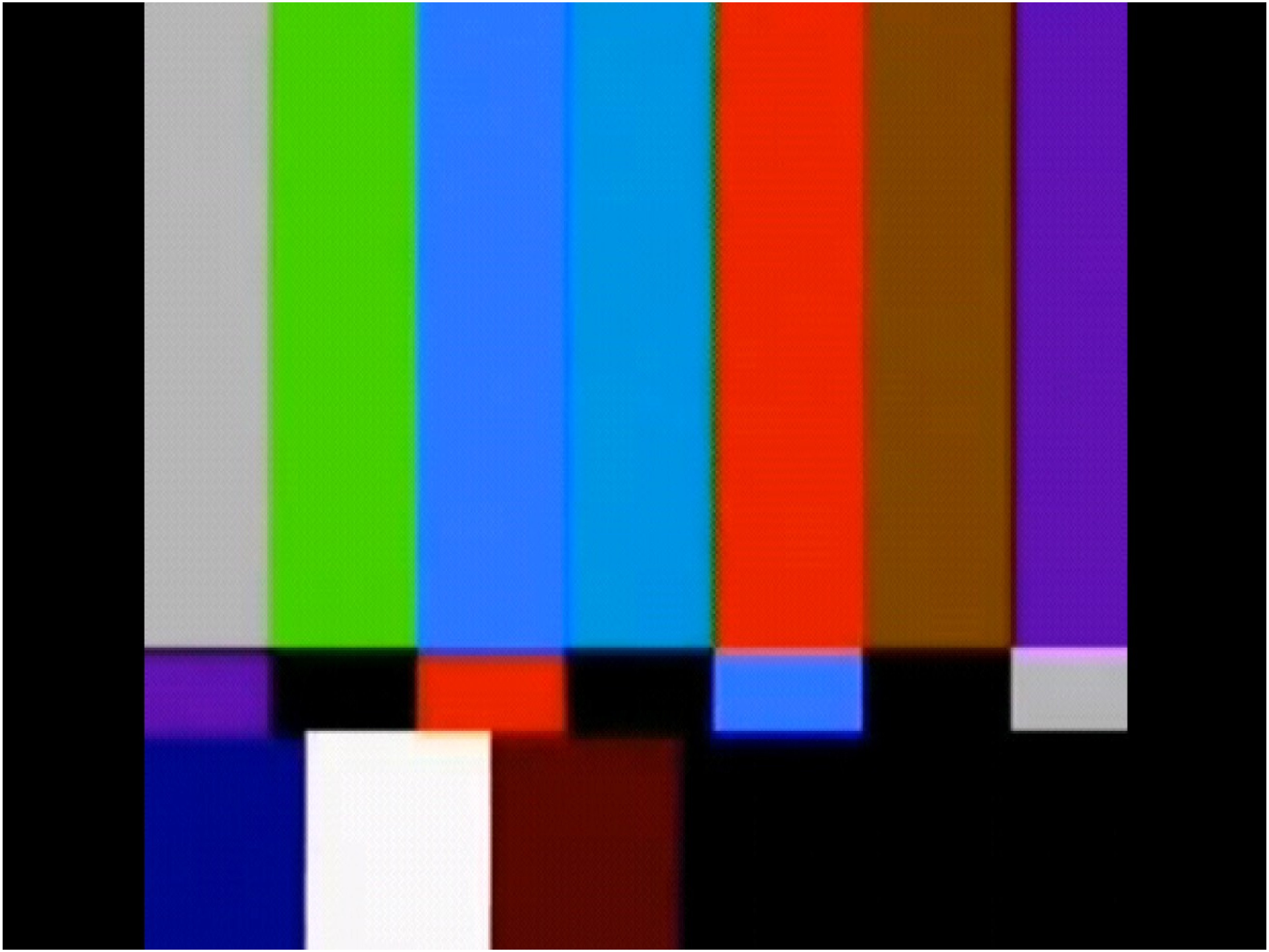
Acknowledgement

This instructional power point brought to you by
Numerical Methods for STEM undergraduate
http://numericalmethods.eng.usf.edu
Committed to bringing numerical methods to the undergraduate

For instructional videos on other topics, go to

http://numericalmethods.eng.usf.edu/videos/

This material is based upon work supported by the National Science Foundation under Grant # 0717624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Fast Fourier Transform

Part: Companion Node Observation

http://numericalmethods.eng.usf.edu

For more details on this topic

- Go to http://numericalmethods.eng.usf.edu
- Click on Keyword
- Click on Fast Fourier Transform

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

You are free

- to **Share** – to copy, distribute, display and perform the work
- to **Remix** – to make derivative works

Under the following conditions

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** — You may not use this work for commercial purposes.
- **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Lecture # 13

Chapter 11.05 : Companion Node Observation (Contd.)

Careful observation of Figure 2 has revealed that each computed l^{th} vector (where $l = 1,2,\ldots, r$ and $N = 2^r = 2^4 = 16$) we can always find two (companion) nodes which came from the same pair of nodes in the previous vector.

For example, $f_1(0)$ and $f_1(8)$ are computed in terms of $f(0)$ and $f(8)$.

Similarly, the companion nodes $f_2(8)$ and $f_2(12)$ are computed from the same pair of nodes as $f_1(8)$ and $f_1(12)$.

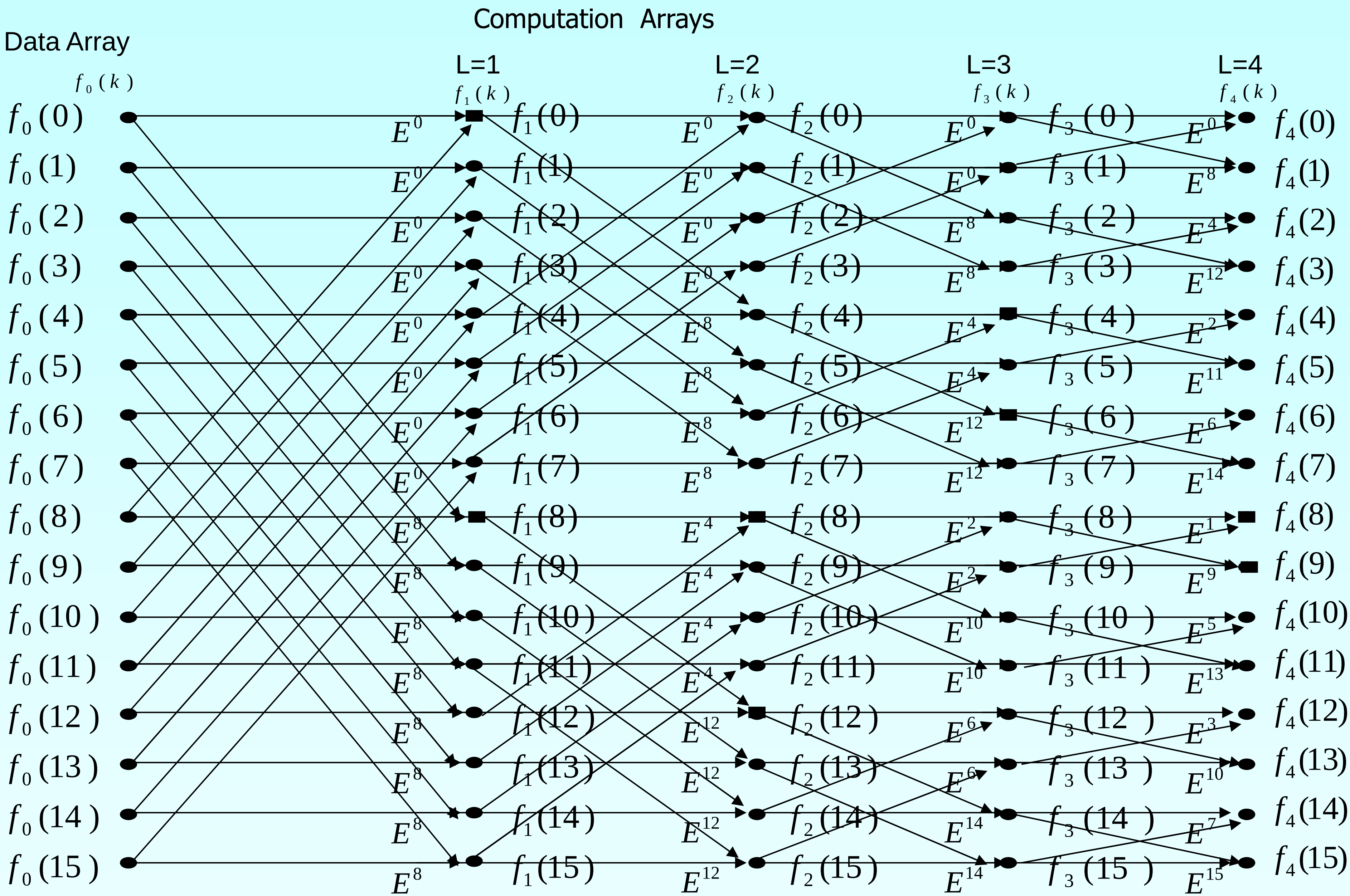


Figure 2. Graphical Form of FFT (Eq. 9) for the case

$$N = 2^r = 2^4 = 16$$

Companion Node Observation cont.

Furthermore, the computation of companion nodes are independent of other nodes (within the l^{th} –vector). Therefore, the computed $f_1(0)$ and $f_1(8)$ will override the original space of $f(0)$ and $f(8)$.

Similarly, the computed $f_2(8)$ and $f_2(12)$ will override the space occupied by $f_1(8)$ and $f_1(12)$ which in turn, will occupy the original space of $f(8)$ and $f(12)$.

Hence, only one complex vector (or 2 real vectors) of length N are needed for the entire FFT process.

Companion Node Spacing

Observing Figure 2, the following statements can be made:

a) in the first vector ($l = 1$), the companion $f_1(0)$ nodes $f_1(8)$ and are separated by $k = 8\left(or\ \frac{N}{2^l} = \frac{16}{2^1}\right)$

b) in the second vector ($l = 2$), the companion nodes $f_2(8)$ and $f_2(12)$ are separated by $k = 4$.

$(or\ \frac{N}{2^l} = \frac{16}{2^2} = \frac{16}{4}), etc.$

Companion Node Computation

The operation counts in any companion nodes (of the $l^{th} = 2^{nd}$ vector), such as $f_2(8)$ and $f_2(12)$ can be explained as (see Figure 2).

$$f_2(8) = f_1(8) + f_1(12) \times E^4 \quad (15)$$

$$\begin{aligned} f_2(12) &= f_1(8) + f_1(12) \times E^{12} \\ &= f_1(8) + f_1(12) \times E^8 E^4 \\ &= f_1(8) + f_1(12) \left[e^{-i\frac{2\pi}{(N=16)}} \right]^8 E^4 \\ &= f_1(8) + f_1(12) \left[e^{-i\pi} \right] E^4 \end{aligned}$$

$$f_2(12) = f_1(8) - f_1(12) \times E^4 \quad (16)$$

Companion Node Computation cont.

Thus, the companion nodes $f_2(8)$ and $f_2(12)$ computation will require 1 complex multiplication and 2 complex additions (see Eq. (15) and (16)). The weighting factors for the companion nodes ($f_2(8)$ and $f_2(12)$) are E^4(or E^U) and E^{12}(or $E^{U+N/2}$), respectively.

$$f_l(k) = f_{l-1}(k) + E^U f_{l-1}(k + \frac{N}{2^l}) \qquad (17)$$

$$f_l(k + \frac{N}{2^l}) = f_{l-1}(k) - E^U f_{l-1}(k + \frac{N}{2^l}) \qquad (18)$$

Skipping Computation of Certain Nodes

Because the pair of companion nodes k and $k+\frac{N}{2^L}$ are separated by the "distance" $\frac{N}{2^L}$, at the L^{th} level, after every $\frac{N}{2^L}$ node computation, then the next $\frac{N}{2^L}$ nodes will be skipped. (see Figure 2)

THE END

http://numericalmethods.eng.usf.edu

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

Acknowledgement

This instructional power point brought to you by Numerical Methods for STEM undergraduate

http://numericalmethods.eng.usf.edu

Committed to bringing numerical methods to the undergraduate

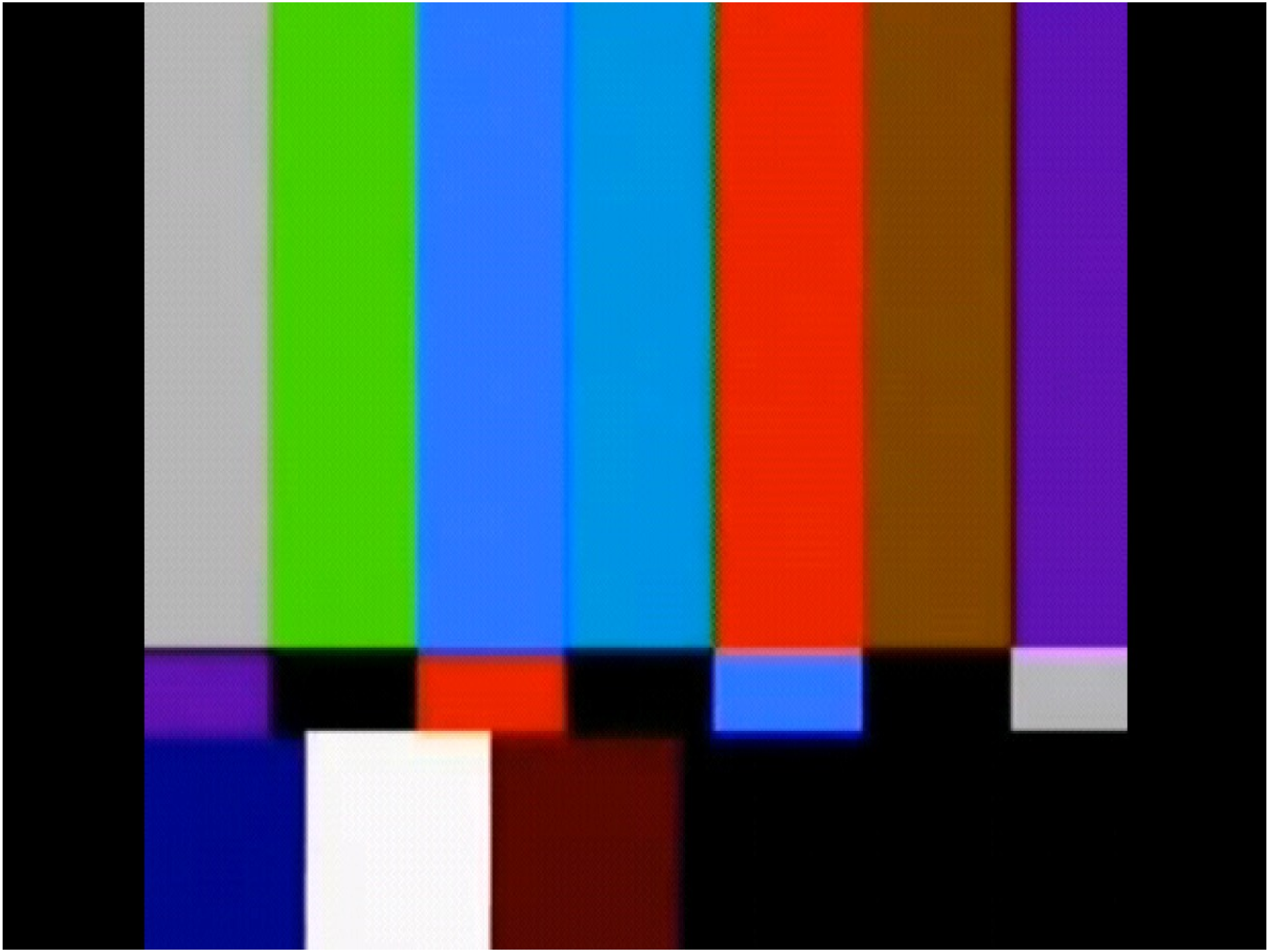
Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

For instructional videos on other topics, go to

http://numericalmethods.eng.usf.edu/videos/

This material is based upon work supported by the National Science Foundation under Grant # 0717624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Numerical Methods

Fast Fourier Transform

Part: Determination of E^U

http://numericalmethods.eng.usf.edu

Old Dominion University

University of South Florida

For more details on this topic

- Go to http://numericalmethods.eng.usf.edu
- Click on Keyword
- Click on Fast Fourier Transform

You are free

- to **Share** – to copy, distribute, display and perform the work
- to **Remix** – to make derivative works

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

Under the following conditions

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** — You may not use this work for commercial purposes.
- **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Lecture # 14

Chapter 11.05: Determination of E^U

The values of "U"

$$f_l(k) = f_{l-1}(k) + E^U f_{l-1}\left(k + \frac{N}{2^l}\right);$$

$$f_l\left(k + \frac{N}{2^l}\right) = f_{l-1}(k) - E^U f_{l-1}\left(k + \frac{N}{2^l}\right)$$

can be determined by the following steps:

<u>Step 1:</u> Express the index $k(= 0,1,2,\ldots, N-1)$ in binary form, using r bits. For $k = 8, L = 2, r = 4,$ and $N = 2^r = 2^4 = 16,$ one obtains:

$$k = 8 = 1,0,0,0 = (1)2^{r-1=3} + (0)2^2 + (0)2^1 + (0)2^0$$

Determination of E^U cont.

<u>Step 2:</u> Sliding this binary number $r-L=4-2=2$ positions to the right, and fill in zeros, the results are

$$1,0,0,0 \rightarrow X, X, 1,0 \rightarrow 0,0,1,0$$

It is important to realize that the results of Step 2 (0,0,1,0) are equivalent to expressing an integer $M=\frac{k}{2^{r-L}}=\frac{8}{2^{4-2}}=2$ in binary format. In other words

$$M=2=(0,0,1,0)$$

Determination of E^U cont.

<u>Step 3:</u> Reverse the order of the bits, then (0,0,1,0) becomes (0,1,0,0) = U.
Thus,

$$U = (0)2^3 + (1)2^2 + (0)2^1 + (0)2^0 = 4$$

It is "NOT" really necessary to perform Step 3, since the results of Step 2 can be used to compute "U" as following

$$U = (0)2^0 + (0)2^1 + (1)2^2 + (0)2^3 = 4$$

Computer Implementation to find E^U

Based on the previous discussions (with the 3-step procedures), to find the value of "U", one only needs a procedure to express an integer $M = \frac{k}{2^{r-L}}$ in binary format, with r bits.

Assuming M (a base 10 number) can be expressed as (assuming $r = 4$ bits)

$$M = a_4 a_3 a_2 a_1 = J_1 \tag{19}$$

Computer Implementation cont.

Divide M by 2, $J_2 = J_1/2$ then multiply the truncated result by 2 $(JJ_2 = J_2 \times 2)$, and compute the difference between the original number and the new number.

Computer Implementation cont.

Compute the difference between the original number and the new number $(= M = J_1) \& \& JJ_2$:

$$IDIFF = J_1 - JJ_2 \left\{ = M - \left(\frac{M}{2} \right)_{Truncated} \times 2 \right\} \qquad (20)$$

If *IDIFF = 0, then the bit* $a_1 = 0$

If *IDIFF ≠ 0, then the bit* $a_1 = 1$

Computer Implementation cont.

Once the bit a_1 has been determined, the value of J_1 is set to J_2(or value of J_1 is reduced by a factor of 2; since the previous .

$$J_1 = M = a_4 a_3 a_2 a_1$$

$$J_1 = M = a_4(2^3) + a_3(2^2) + a_2(2^1) + a_1(2^0)$$

A similar process can be used to determine the value of process can be used to determine the next bit a_2 etc.

Example 1

For $k = 8, N = 16 = 2^r, r = 4$ bits and $L = 2$.
Find the value of U.

$$M = \frac{k}{2^{r-L}} = \frac{8}{2^{4-2}} = 2 = J_1$$

Determine the bit a_1 (Index $I = 1$)
Initialize $U = 0$

$$J_2 = \frac{J_1}{2} = \frac{2}{2} = 1$$

$$IDIFF = J_1 - (JJ_2 = J_2 \times 2) = 2 - (1)(2) = 0$$

Thus $a_1 = 0$

$$U = U \times 2 + IDIFF = 0 \times 2 + 0 = 0$$

Example 1 cont.

Determine the bit a_2 (Index $I = 2$)

$$J_1 = J_2 = 1$$

$$J_2 = \frac{J_1}{2} = \frac{1}{2} = 0$$

$$IDIFF = J_1 - (JJ_2 = J_2 \times 2) = 1 - (0 \times 2) = 1$$

Thus $a_2 = 1$

$$U = U \times 2 + IDIFF = 0 \times 2 + 1 = 1$$

Example 1 cont.

Determine the bit a_3 (Index $I = 3$)

$$J_1 = J_2 = 0$$

$$J_2 = \frac{J_1}{2} = \frac{0}{2} = 0$$

$$IDIFF = J_1 - (JJ_2 = J_2 \times 2) = 0 - (0)(2) = 0$$

Thus $a_3 = 0$

$$U = U \times 2 + IDIFF = 1 \times 2 + 0 = 2$$

Example 1 cont.

Determine the bit a_4 (Index $I = 4$)

$$J_1 = J_2 = 0$$

$$J_2 = \frac{J_1}{2} = \frac{1}{2} = 0$$

$$IDIFF = J_1 - (JJ_2 = J_2 \times 2) = 0 - (0)(2) = 0$$

Thus $a_4 = 0$

$$U = U \times 2 + IDIFF = 2 \times 2 + 0 = 4$$

THE END

http://numericalmethods.eng.usf.edu

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

Acknowledgement

This instructional power point brought to you by Numerical Methods for STEM undergraduate

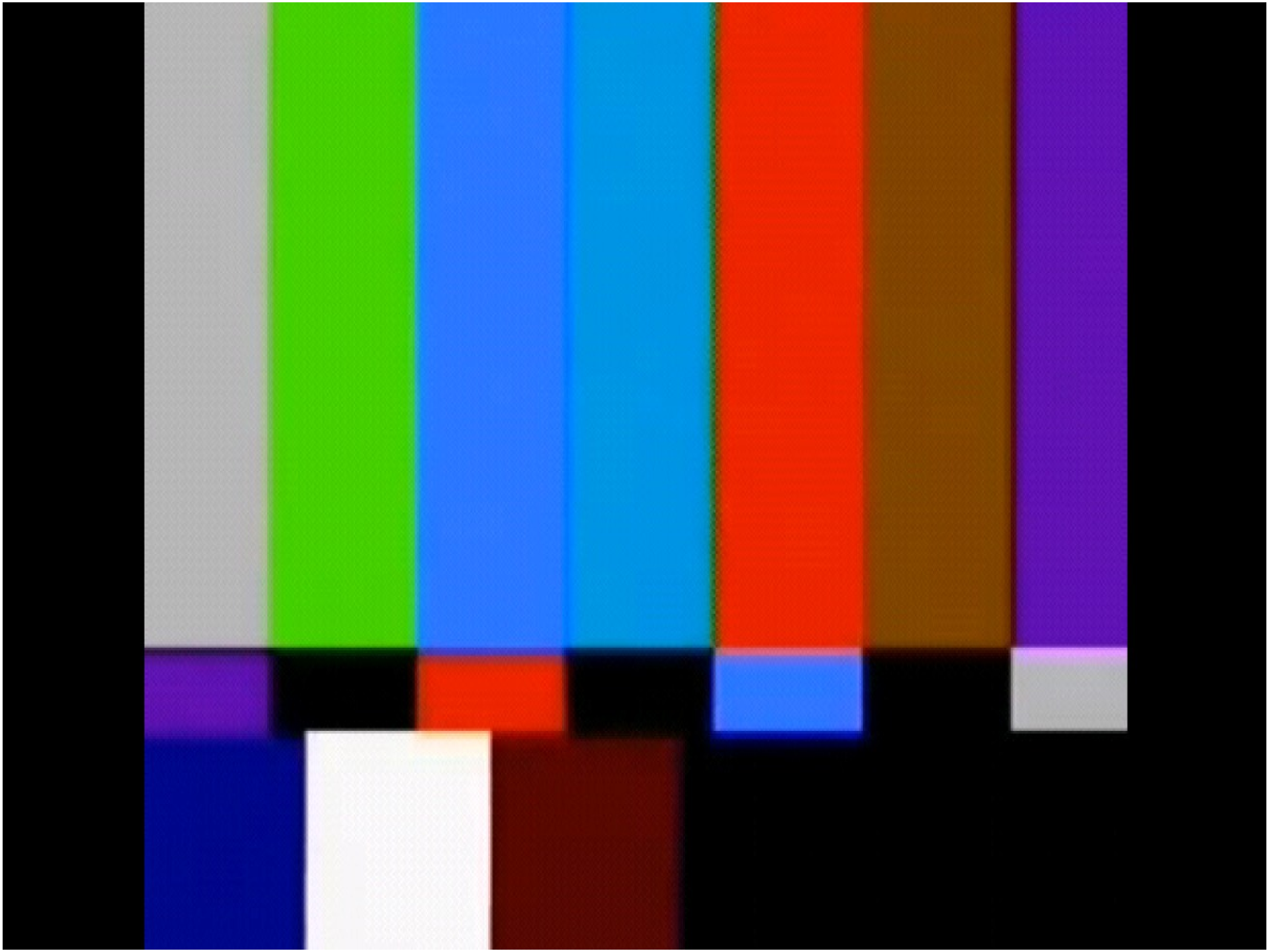
http://numericalmethods.eng.usf.edu

Committed to bringing numerical methods to the undergraduate

For instructional videos on other topics, go to

http://numericalmethods.eng.usf.edu/videos/

This material is based upon work supported by the National Science Foundation under Grant # 0717624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Old Dominion University
USF University of South Florida

Numerical Methods

Fast Fourier Transform

Part: Unscrambling the FFT

http://numericalmethods.eng.usf.edu

For more details on this topic

- Go to http://numericalmethods.eng.usf.edu
- Click on Keyword
- Click on Fast Fourier Transform

You are free

- to **Share** – to copy, distribute, display and perform the work
- to **Remix** – to make derivative works

Under the following conditions

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** — You may not use this work for commercial purposes.
- **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Lecture # 15

Chapter 11.05: Unscrambling the FFT (Contd.)

For the case

$$N = 16 = 2^{r=4}$$

, (see Figure 2), the final “bit-reversing” operation for FFT is shown in Figure 3.

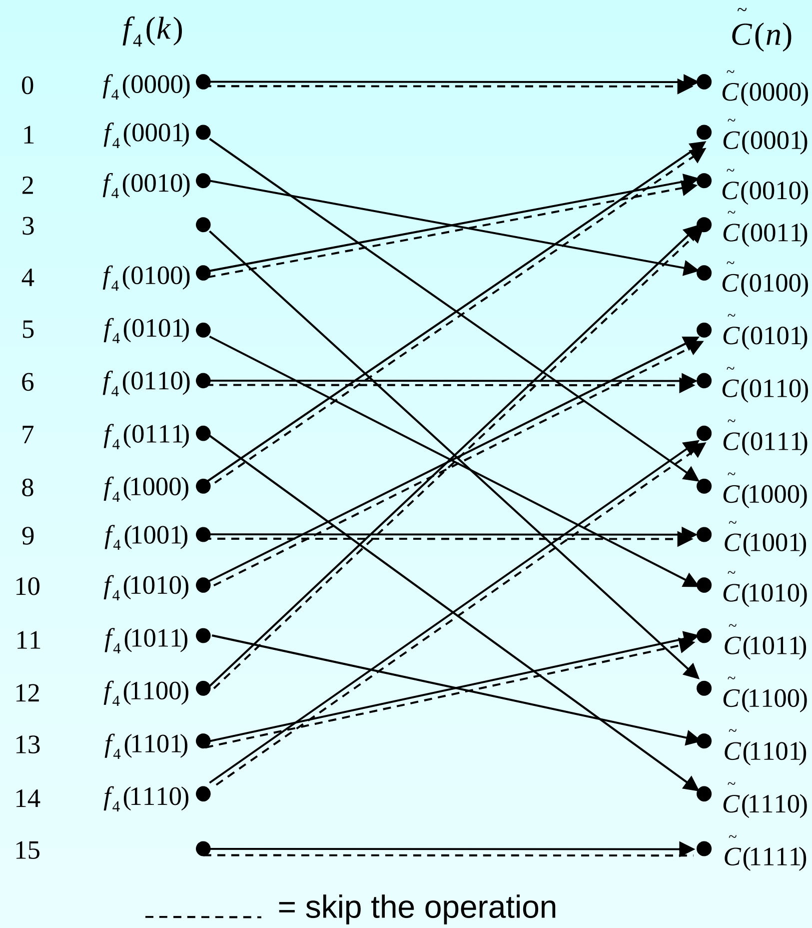


Figure 3. Final “bit-reversing” for FFT (with $N = 2^r = 2^4 = 16$)

For do-loop index $k = 0 = (0, 0, 0, 0) \Rightarrow i = (0, 0, 0, 0) =$ bit-reversion $= 0$

If (i.GT.k) Then

$T = f_4(k)$

$f_4(k) = f_4(i)$

$f_4(i) = T$

Endif

Hence, $f_4(0) = f_4(0)$ no swapping.

55

For $k = 1 = (0,0,0,1) \Rightarrow i = (1,0,0,0)$
$=$ bit-reversion $= 8$

If (i.GT.k) Then

$T = f_4(k=1)$

$f_4(k=1) = f_4(i=8)$

$f_4(i=8) = T$

Endif

Hence, $f_4(1) = f_4(8)$ are swapped.

.For $k=2=(0,0,1,0) \Rightarrow i = (0,1,0,0) = 4$
Hence, $f_4(2) = f_4(4)$; are swapped.

.For $k=3=(0,0,1,1) \Rightarrow i = (1,1,0,0) = 12$
Hence, $f_4(3) = f_4(12)$; are swapped.

. For $k=4=(0,1,0,0) \Rightarrow i=(0,0,1,0)=2$

In this case, since "i" is not greater than "k".
Hence, no swapping, since f_4 $(k = 2)$ and f_4 $(i = 4)$; had already been swapped earlier!
.etc.

Computer Implementation of FFT case for $N=2^r$

The pair of companion nodes computation are given by Eqs.(17) and (18). To avoid "complex number" operations,Eq.(17) can be computed based on "real number" operations, as following

$$\left\{f_L^R(k)+if_L^I(k)\right\}=\left\{f_{L-1}^R(k)+if_{L-1}^I(k)\right\}$$
$$+\left\{E^{U,R}+iE^{U,I}\right\}\times\left\{f_{L-1}^R\left(k+\frac{N}{2^L}\right)+if_{L-1}^I\left(k+\frac{N}{2^L}\right)\right\} \tag{21}$$

In Eq. (21), the superscripts R and I denote real and imaginary components, respectively.

Computer Implementation cont.

Multiplying the last 2 complex numbers, one obtains

$$\left\{ f_L^R(k) + i f_L^I(k) \right\} = \left\{ f_{L-1}^R(k) + i f_{L-1}^I(k) \right\}$$

$$+ \left\{ E^{U,R} \times f_{L-1}^R(k + \frac{N}{2^L}) - E^{U,I} \times f_{L-1}^I(k + \frac{N}{2^L}) \right\}$$

$$+ i \left\{ E^{U,R} \times f_{L-1}^I(k + \frac{N}{2^L}) + E^{U,I} \times f_{L-1}^R(k + \frac{N}{2^L}) \right\} \quad (22)$$

Equating the real (and then, imaginary) components on the Left-Hand-Side (LHS), and the Right-Hand-Side (RHS) of Eq. (22), one obtains

Computer implementation cont.

$$\left\{ f_L^R(k) \right\} = \left\{ f_{L-1}^R(k) \right\} + \left\{ E^{U,R} \times f_{L-1}^R(k + \frac{N}{2^L}) - E^{U,I} \times f_{L-1}^I(k + \frac{N}{2^L}) \right\}$$

(23A)

$$\left\{ f_L^I(k) \right\} = \left\{ f_{L-1}^I(k) \right\} + \left\{ E^{U,R} \times f_{L-1}^I(k + \frac{N}{2^L}) + E^{U,I} \times f_{L-1}^R(k + \frac{N}{2^L}) \right\}$$

(23B)

Computer implementation cont.

Recall Eq. (4)

$$E = e^{-i\frac{2\pi}{N}}$$

Hence

$$E^U = \left(e^{-i\frac{2\pi}{N}} \right)^U = e^{-i\frac{2\pi U}{N}} = e^{-i\theta} = \cos(\theta) - i\sin(\theta) \qquad (24)$$

where

$$\theta = \frac{2\pi U}{N} = \frac{6.28U}{N} \qquad (25)$$

Thus:

$$E^{U,R} = \cos(\theta) \qquad (26A)$$

$$E^{U,I} = -\sin(\theta) \qquad (26B)$$

Computer Implementation cont.

Substituting Eqs. (26A) and (26B) into Eqs. (23A) and (23B), one gets

$$\left\{f_L^R(k)\right\} = \left\{f_{L-1}^R(k)\right\} + \left\{\cos(\theta) \times f_{L-1}^R\left(k + \frac{N}{2^L}\right) + \sin(\theta) \times f_{L-1}^I\left(k + \frac{N}{2^L}\right)\right\} \quad \text{(27A)}$$

$$\left\{f_L^I(k)\right\} = \left\{f_{L-1}^I(k)\right\} + \left\{\cos(\theta) \times f_{L-1}^I\left(k + \frac{N}{2^L}\right) - \sin(\theta) \times f_{L-1}^R\left(k + \frac{N}{2^L}\right)\right\} \quad \text{(27B)}$$

Similarly, the single (complex number) Eq. (18) can be expressed as 2 equivalent (real number) Eqs. Like Eqs. (27A) and (27B).

THE END

http://numericalmethods.eng.usf.edu

Acknowledgement

This instructional power point brought to you by Numerical Methods for STEM undergraduate

http://numericalmethods.eng.usf.edu

Committed to bringing numerical methods to the undergraduate

Old Dominion UNIVERSITY

USF UNIVERSITY OF SOUTH FLORIDA

For instructional videos on other topics, go to

http://numericalmethods.eng.usf.edu/videos/

This material is based upon work supported by the National Science Foundation under Grant # 0717624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.