The Scribed Note for AMSC 661 on May 5, 2005 – Transform Methods

Instructor: Dr. O'Leary

Mian Li

1. **The set up:**
   1) A domain Ω: can be continuous or discrete
   2) Basis functions $z(s, x)$ need to be orthogonal (linear independent each other) since we prefer the coefficient of one basis function will not affect others.
2. **Continuous Fourier Transform**

   $f(x)$ need to decay when x go to $\pm\infty$
3. **Discrete Fourier Transform: DFT**
   1) $\Omega = \{0, 1,\ldots, n\text{-}1\}$. In normalized space $x = [0,1]$

      let $j = 0,1,\ldots,n\text{-}1$ then

      $x = j * \Delta x$, where $\Delta x = 1/n$

   2) We sample the function $z(s,x)$ and $f(x)$ on those discrete x points:

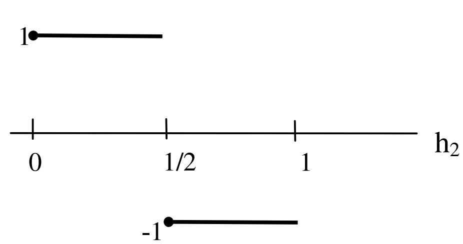
      $z(s, x) = e^{2\pi isx} = e^{2\pi isj/n}$

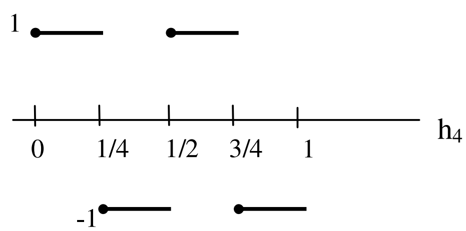
$$F(s) = \sum_{j=0}^{n-1} f(x) e^{2\pi isj/n}$$

   2) DFT has capability to solve the problem in complex domain since

      $e^{2\pi isx} = \cos(2\pi sx) + i*\sin(2\pi sx)$
4. **Discrete Sin Transform: DST**
   1) If the problem defined in the real space, then DST will be more efficient than DFT (does not need to take care of the imagine part).
   2) DST will be a good tool for the function whose boundary condition is equal to 0 since *sin* function will be 0 on the boundary 0 and 1 (the original domain can be normalized to [0,1]).
5. **Haar Transform**

   The figures for h_1, h_2 and h_4





6. Discrete Haar Transform: changed notes, please look at the new version for this part.

7. One property of the Fourier Transform

1) Basic idea: $F(s)=\int_{-\infty}^{\infty} f(x)e^{2\pi isx}dx$ is our problem. If $f(x)$ is a periodic function, say like a *cos* function scaled from [0, 2π] to [0, p] for one period. And we know $z(s, x) = e^{2\pi isx}$ is also periodic with respect to p. Based on that, we can change the boundary of integration from [-∞, +∞] to [0, p].

2) So if f is periodic, so that $f(x + p) = f(x)$, then (it can be shown that) $F(s)$ is nonzero only for $s = k/p$ for $k = 0, \pm 1, \pm 2, \ldots$

8. Fast computation of the discrete transform

1) Continuous transform will calculate the integration of functions, which is not a quick or "easy" work to do.

2) Discrete transform may work in less than $O(n^2)$ time, which is faster.

9. Fast Fourier Transform: FFT

Problem definition: $F(s)=\sum_{x=0}^{n-1} f(x)e^{-2\pi isx/n}$

Example: Let n =4, $w = e^{-2\pi i/n} = \cos(2\pi/n) - i\sin(2\pi/n)$

Then $F_4\begin{bmatrix} f(0)\\ f(1)\\ f(2)\\ f(3)\end{bmatrix}=\begin{bmatrix} F(0)\\ F(1)\\ F(2)\\ F(3)\end{bmatrix}$. We fill numbers in to F_4 step by step

When s =0, whatever x is 0,…,3, $z(s, x) = e^{2\pi isx/4} = e^{0/n}$ $F_4(1, :) = [1\ 1\ 1\ 1]$

When s =1, x = 0,…,3, $z(s, x) = e^{-2\pi ix/n} = [\cos(2\pi/n) - i\sin(2\pi/n)]^x = w^x$ so $F_4(2, :) = [1\ w\ w^2\ w^3]$

Do the same thing when s = 2 and 3 we can have

$$F_4=\begin{bmatrix}1&1&1&1\\1&w&w^2&w^3\\1&w^2&w^4&w^6\\1&w^3&w^6&w^9\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&-i&-1&i\\1&-1&1&-1\\1&i&-1&-i\end{bmatrix} \text{ and } \begin{bmatrix}1&1&1&1\\1&w&w^2&w^3\\1&w^2&w^4&w^6\\1&w^3&w^6&w^9\end{bmatrix}\begin{bmatrix}f(0)\\f(1)\\f(2)\\f(3)\end{bmatrix}=\begin{bmatrix}F(0)\\F(1)\\F(2)\\F(3)\end{bmatrix}$$

We can verify that the columns in F_4 are orthogonal and $F_4^T F_4 = I$

Then we let n =8, $z = e^{-2\pi isx/8} = w^2$.

We have:

$z^2 = w$

$z^8 = 1$

$z^4 = -1$

$z^5 = zz^4 = -z$

And then we have

$$F_8 = \begin{bmatrix} \mathbf{1} & 1 & \mathbf{1} & 1 & \mathbf{1} & 1 & \mathbf{1} & 1 \\ \mathbf{1} & z & \mathbf{z^2} & z^3 & \mathbf{z^4} & z^5 & \mathbf{z^6} & z^7 \\ \mathbf{1} & z^2 & \mathbf{z^4} & z^6 & \mathbf{1} & z^2 & \mathbf{z^4} & z^6 \\ \mathbf{1} & z^3 & \mathbf{z^6} & z^1 & \mathbf{z^4} & z^7 & \mathbf{z^2} & z^5 \\ \mathbf{1} & z^4 & \mathbf{1} & z^4 & \mathbf{1} & z^4 & \mathbf{1} & z^4 \\ \mathbf{1} & z^5 & \mathbf{z^2} & z^7 & \mathbf{z^4} & z & \mathbf{z^6} & z^3 \\ \mathbf{1} & z^6 & \mathbf{z^4} & z^2 & \mathbf{1} & z^6 & \mathbf{z^4} & z^2 \\ \mathbf{1} & z^7 & \mathbf{z^6} & z^5 & \mathbf{z^4} & z^3 & \mathbf{z^2} & z^1 \end{bmatrix}$$

(*)

We reorder matrix F_8 by changing the ordering of columns from [1 2 3 4 5 6 7 8] to [1 3 5 7 2 4 6 8] (put columns with odd indices at the beginning and then the columns with even indices.)

So

$$F_8^{reordered} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 1 & 1 & 1 & 1 \\ \mathbf{1} & \mathbf{z^2} & \mathbf{z^4} & \mathbf{z^6} & z & z^3 & z^5 & z^7 \\ \mathbf{1} & \mathbf{z^4} & \mathbf{1} & \mathbf{z^4} & z^2 & z^6 & z^2 & z^6 \\ \mathbf{1} & \mathbf{z^6} & \mathbf{z^4} & \mathbf{z^2} & z^3 & z & z^7 & z^5 \\ 1 & 1 & 1 & 1 & \mathbf{-1} & \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \\ 1 & z^2 & z^4 & z^6 & \mathbf{-z} & \mathbf{-z^3} & \mathbf{-z^5} & \mathbf{-z^7} \\ 1 & z^4 & 1 & z^4 & \mathbf{-z^2} & \mathbf{-z^6} & \mathbf{-z^2} & \mathbf{-z^6} \\ 1 & z^6 & z^4 & z^2 & \mathbf{-z^3} & \mathbf{-z} & \mathbf{-z^7} & \mathbf{-z^5} \end{bmatrix} = \begin{bmatrix} \mathbf{F_4} & DF_4 \\ F_4 & \mathbf{-DF_4} \end{bmatrix}$$

(*)

So F_8 can be expressed as $F_8^{reordered} = \begin{bmatrix} F_4 & DF_4 \\ F_4 & -DF_4 \end{bmatrix} = \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_4 & 0 \\ 0 & F_4 \end{bmatrix}$

And

$$F_8 x = F_8^{reordered} x^{reordered} = \begin{bmatrix} F_4 & DF_4 \\ F_4 & -DF_4 \end{bmatrix} \begin{bmatrix} x_{odd} \\ x_{even} \end{bmatrix} = \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_4 & 0 \\ 0 & F_4 \end{bmatrix} \begin{bmatrix} x_{odd} \\ x_{even} \end{bmatrix}$$

$$= \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_4 x_{odd} \\ F_4 x_{even} \end{bmatrix}$$

Where $D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & z^2 & 0 & 0 \\ 0 & 0 & z^3 & 0 \\ 0 & 0 & 0 & z^4 \end{bmatrix}$

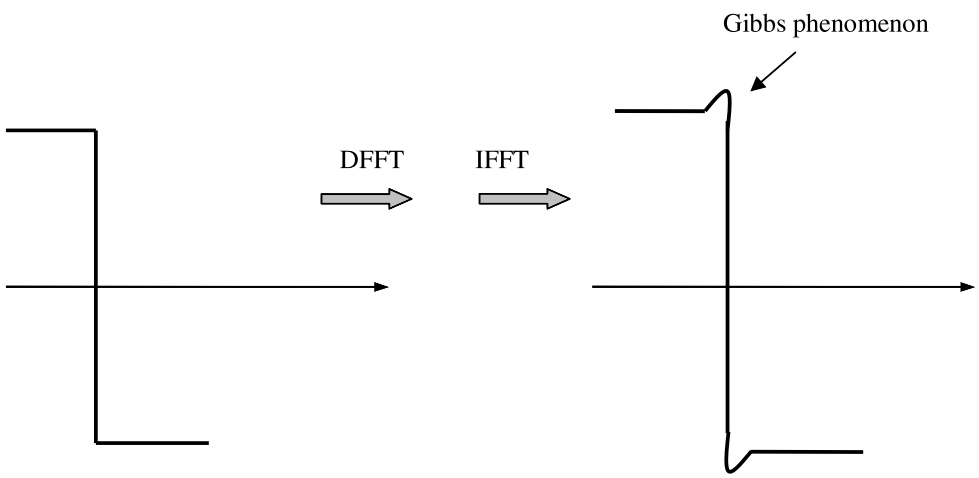
In this way, we can recursively decompose n to solve the DFT. For example, when n =128. If we use FFT, 5053 float point operation will be performed as well as 214665 float point operation will be performance if we don't use Fast DFT. The fast algorithm work grows as $n\log_2 n$ as long as n is a power of 2. We call it Radix 2 FFT.

If n is not a power of 2, the fast algorithm work will be O(n*p) where p is a small number if n has small prime factors, such as p will be small for n= $2^6 5^9$, but may not be small for n = 127^5.

(*): formulations with * are referred from the file:

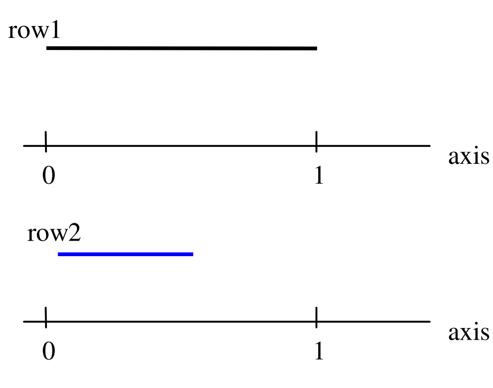
www.cs.umd.edu/users/oleary/c460/460matrixhand.pdf

Notice: when $f(x)$ is a discontinuous function, like step function, after DFFT and IFFT, the result function is different with the original one.



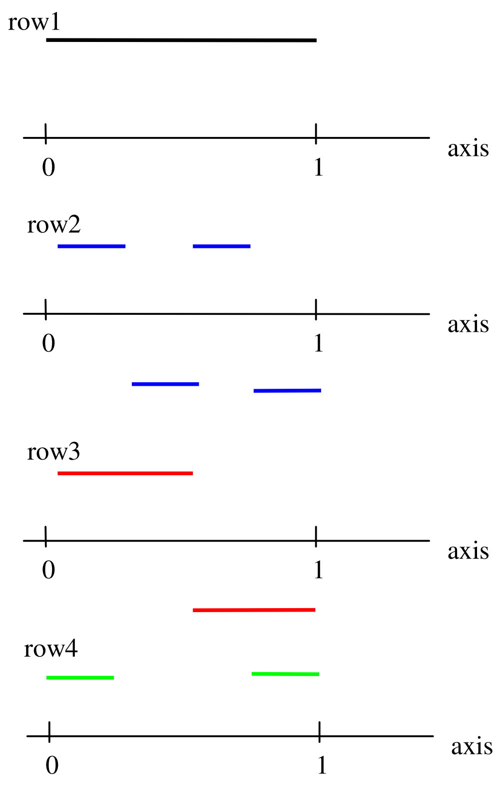
10. Discrete Haar Transform

Let $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, then after multiplying by H_1, the function should be like the following pattern.



And then $H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & - & 1 & - \\ 1 & 1 & - & - \\ 1 & - & - & 1 \end{bmatrix} = \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix}$ where "-" stands for "-1"

After multiplying by H_2, the function should be like as the following pattern.



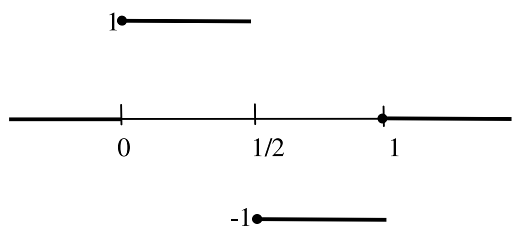
We can find out that the row 4 has the same period with row 3, just shifted from row 3

And

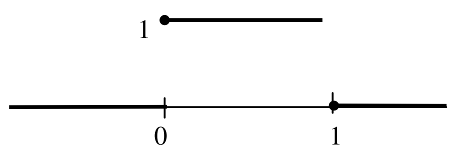
$$H_3 = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{array}\right] \begin{array}{l} \\ \\ \text{row 3} \\ \text{shifted from row 3} \\ \text{row5} \\ \\ \text{shifted from row 5} \\ \\ \end{array}$$

11. Wavelet Transform
    1) Fourier Transform: Decompose data (signal) by frequency (Hz)
    2) Wavelet Transform: Decompose by frequency and also locality

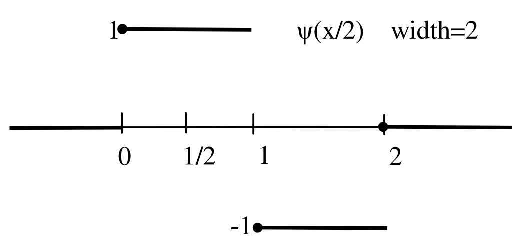
12. Mother wavelet



13. Father wavelet



14. Dilation and Translations
    1) Dilation of mother function



    2) $\psi(x/2^j)$ has width 2^j.