GEORGIA TECH

MATH, PHYSICS & COMPUTING

MATH 4803BDU,BDG, CS4803QIC, PHYS4803A

QUANTUM INFORMATION & QUANTUM COMPUTING

Fast Fourier transform :

The *fast Fourier transform* (FFT) is an example of fast algorithm used by classical computers. If N is an integer, the Fourier transform of a vector $|z\rangle \in \mathbb{C}^N$ with components $(z_0, \cdots, z_{N-1})$ is given by

$$|\tilde{z}\rangle \;=\; (\mathcal{F}_N|z\rangle)_k \;=\; \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} e^{2\imath\pi k \cdot l/N}\, z_l\,. \tag{1}$$

Whenever $N = 2^n$, the numerical computation of $\mathcal{F}_N$ becomes faster due to the structure of the matrix of $\mathcal{F}_N$ that will be investigated below. For simplicity, whenever $N = 2^n$ let F_n be the matrix

$$F_n \;=\; \sqrt{N}\,\mathcal{F}_N \qquad\qquad N \;=\; 2^n\,. \tag{2}$$

1. Give the explicit expression of the matrices of F_1 and F_2.
2. Give the formula for F_n (see eq. (1)). What is the dimension of the matrix F_n ?
3. By decomposing the sum over l into the sums over l' whenever $l = 2l'$ or $l = 2l' + 1$, show that F_n can be expressed in term of F_{n-1}.
4. More precisely, show that the answer of the question (3.) above can be expressed as

$$F_n \;=\; \begin{bmatrix} \mathbf{1}_{2^{n-1}} & D_n \\ \mathbf{1}_{2^{n-1}} & -D_n \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} & 0 \\ 0 & F_{n-1} \end{bmatrix} \cdot [P_n]\,, \tag{3}$$

where $\mathbf{1}_L$ is the identity matrix of dimension L, D_n is the diagonal matrix of dimension 2^{n-1} with diagonal elements $1, \lambda, \lambda^2, \lambda^3, \cdots, \lambda^{(2^{n-1}-1)}$ respectively, if $\lambda = e^{2\imath\pi/2^n}$, and $[P_n]$ is the matrix of the operator

$$P_n \;:\; \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ c_{2^n-1} \end{bmatrix} \longrightarrow \begin{bmatrix} c_0 \\ c_2 \\ \vdots \\ c_{2^n-2} \\ c_1 \\ c_3 \\ \vdots \\ c_{2^n-1} \end{bmatrix}$$

5. To transfer this computation easily on a computer, it is convenient to change the coordinate labels as follows : with each integer $k \in \{0, 1, 2, \cdots, 2^n - 1\}$, is associated its *dyadic* decomposition $k = \epsilon_0 + 2\epsilon_1 + \cdots + 2^{n-1}\epsilon_{n-1}$ where the ϵ_r's take on values 0 or 1. If $\underline{\epsilon} = (\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1})$, then any sum over k is equivalent to summing over all possible $\underline{\epsilon}$'s. Show that $\underline{\epsilon}$ takes on 2^n different values.
   Show also that the operator P_n above can be expressed as
   $$P_n \;:\; c(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) \mapsto c(\epsilon_{n-1}, \epsilon_0, \epsilon_1, \cdots, \epsilon_{n-2}) \qquad \textit{(cyclic permutation of the } \epsilon_r\textit{'s)},$$
   where $c(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) = c_k$ if $k = \sum_r 2^r \epsilon_r$.
6. The main idea of the FFT is to iterate the formula (3), namely expressing F_{n-1} in term of F_{n-2}, then in term of F_{n-3}, *etc.* all the way down to F_1. In order to proceed, show that
   (a) the matrix $F_{n-1} \oplus F_{n-1}$, which appears in the middle of formula (3), does not change the last digit ϵ_{n-1} ;
   (b) iterating j times, F_{n-j} occurs through a direct sum $F_{n-j} \oplus \cdots \oplus F_{n-j}$ containing 2^j terms ;
   (c) this last sum does not modifies the digits $\epsilon_{n-j}, \epsilon_{n-j+1}, \cdots, \epsilon_{n-1}$.
7. Applying the formula (3) to F_{n-1} implies using $\tilde{P}_{n-1} = P_{n-1} \oplus P_{n-1}$. Iterating, this gives $\tilde{P}_{n-j} = P_{n-j} \oplus \cdots \oplus P_{n-j}$ (2^j terms).
   (a) Compute the action of $\tilde{P}_{n-1}$, then $\tilde{P}_{n-2}$.
   (b) Deduce what is the action of $\tilde{P}_{n-j}$ for all j's.
   (c) Prove that the product $\hat{P}_n = \tilde{P}_2 \tilde{P}_3 \cdots \tilde{P}_n$ corresponds to the transformation
   $$\hat{P}_n \;:\; c(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) \mapsto c(\epsilon_{n-1}, \epsilon_{n-2}, \cdots, \epsilon_2, \epsilon_1)\,.$$
   (*Hint :* use (6c.))
8. Let c be the vector giving the initial data, namely the vector that is to be Fourier transformed. Let $c(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1})$ denote its components. Then let y_0 denote the vector $\hat{P}_n c$, given by inverting the order of the digits.
   (a) Show that the application of F_1 (1st step), gives the vector y_1 in the form
   $$\begin{aligned} y_1(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) &= y_0(0, \epsilon_1, \cdots, \epsilon_{n-1}) + (-1)^{\epsilon_1} y_0(1, \epsilon_1, \cdots, \epsilon_{n-1}) \\ &= \sum_{\eta=0,1} (-1)^{\eta\epsilon_0} y_0(\eta, \epsilon_1, \cdots, \epsilon_{n-1}) \end{aligned}$$
   (b) Using the remark made in (6c.), show that iterating the left part of the formula (3), gives a sequence $y_0, y_1, \cdots, y_n$ of vectors defined recursively by
   $$y_{k+1}(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) = \sum_{\eta=0,1} e^{\frac{2\imath\pi\eta(\epsilon_0 + 2\epsilon_1 + \cdots + 2^k\epsilon_k)}{2^{k+1}}} y_k(\epsilon_0, \cdots, \epsilon_{k-1}, \eta, \epsilon_{k+1}, \cdots, \epsilon_{n-1})$$
   so that y_n is the result.
   (c) Show that the number of operations (multiplications) is of the order of $n2^n = N \ln_2 N$. Compare with the number of multiplications N^2 required by applying directly the formula (1). Compare these two numbers whenever $n = 20$.