FAST FOURIER TRANSFORM

February, 2007

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 $n = 20$.

FAST FOURIER TRANSFORM

Correction

By definition (see eq. (2,1))

$$(F_n|z\rangle)_k = \sum_{l=0}^{2^n-1} e^{2\imath\pi kl/2^n}\, z_l \,. \tag{4}$$

1. For $n=1$ then $e^{2\imath\pi kl/2} = (-1)^{kl}$, while, for $n=2$, $e^{2\imath\pi kl/4} = (\imath)^{kl}$ so that

$$F_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad F_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \imath & -1 & -\imath \\ 1 & -1 & 1 & -1 \\ 1 & -\imath & -1 & \imath \end{bmatrix}. \tag{5}$$

2. The formula (4) gives the matrix elements of F_n

$$(F_n)_{kl} = e^{2\imath\pi kl/2^n}\,.$$

3. The eq. (4) can be written by separating the sum over l into a sum over $l=2l'$ (with $0 \leq l' \leq 2^{n-1}-1$) and the sum over $l = 2l'+1$. Then $e^{2\imath\pi k(2l')/2^n} = e^{2\imath\pi kl'/2^{n-1}}$ and $e^{2\imath\pi k(2l'+1)/2^n} = e^{2\imath\pi k'/2^n} e^{2\imath\pi kl'/2^{n-1}}$. This gives

$$(F_n|z\rangle)_k = \sum_{l'=0}^{2^{n-1}-1} e^{2\imath\pi kl'/2^{n-1}}\, z_{2l'} = e^{2\imath\pi k/2^n} \sum_{l'=0}^{2^{n-1}-1} e^{2\imath\pi kl'/2^{n-1}}\, z_{2l'+1}\,.$$

To interpret this decomposition let $|z_{od}\rangle$ and $|z_{ev}\rangle$ be the vectors of dimension 2^{n-1} with coordinates $(|z_{od}\rangle)_k = z_{2k+1}$ and $(|z_{ev}\rangle)_k = z_{2k}$ respectively. Remarking that

$$e^{2\imath\pi(k+2^{n-1})l'/2^{n-1}} = e^{2\imath\pi kl'/2^{n-1}}, \qquad e^{2\imath\pi(k+2^{n-1})/2^n} = -e^{2\imath\pi k/2^n},$$

leads to

$$\begin{aligned} (F_n|z\rangle)_k &= (F_{n-1}|z_{ev}\rangle)_k + e^{2\imath\pi k/2^n}(F_{n-1}|z_{od}\rangle)_k\,, \\ & \qquad\qquad 0 \leq k \leq 2^{n-1}-1 \\ (F_n|z\rangle)_{k+2^{n-1}} &= (F_{n-1}|z_{ev}\rangle)_k - e^{2\imath\pi k/2^n}(F_{n-1}|z_{od}\rangle)_k\,. \end{aligned} \tag{6}$$

4. Let D_n be the diagonal matrix of dimension 2^{n-1} with $(D_n)_{kl} = e^{2\imath\pi k/2^n}\delta_{kl}$. Then the previous expression (6) can be written in matrix form as

$$F_n|z\rangle \;=\; \begin{bmatrix} \mathbf{1}_{2^{n-1}} & D_n \\ \mathbf{1}_{2^{n-1}} & -D_n \end{bmatrix} \begin{bmatrix} F_{n-1} & 0 \\ 0 & F_{n-1} \end{bmatrix} \begin{bmatrix} |z_{ev}\rangle \\ |z_{od}\rangle \end{bmatrix}.$$

   If P_n is the operator defined by

$$P_n|z\rangle \;=\; \begin{bmatrix} |z_{ev}\rangle \\ |z_{od}\rangle \end{bmatrix},$$

   the last equation leads to the formula (3).

5. The dyadic decomposition of integers smaller than 2^n gives a one-to-one correspondence between $[0, 2^n - 1]$ and the set $\{0,1\}^{\times n}$ of families $\underline{\epsilon} = (\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1})$ where $\epsilon_r \in \{0,1\}$ is the r-th digit. Since each ϵ_r takes on two values and since there are n such digits, $\underline{\epsilon}$ takes on 2^n values.

   Moreover

$$0 \le k \le 2^{n-1} - 1 \;\Leftrightarrow \epsilon_{n-1} = 0\,, \qquad\qquad 2^{n-1} \le k \le 2^n - 1 \;\Leftrightarrow \epsilon_{n-1} = 1\,.$$

   In particular if $0 \le k \le 2^{n-1} - 1$

$$(P_n|c\rangle)_k \;=\; (P_n|c\rangle)(\epsilon_0, \cdots, \epsilon_{n-2}, 0)\,, \qquad\qquad (P_n|c\rangle)_{k+2^{n-1}} \;=\; (P_n|c\rangle)(\epsilon_0, \cdots, \epsilon_{n-2}, 1)\,.$$

   On the other hand, if $k \le 2^{n-1} - 1$, then $2k = 2\epsilon_0 + 2^2\epsilon_1 + \cdots + 2^{n-1}\epsilon_{n-2}$, so that $c_{2k} = c(0, \epsilon_0, \cdots, \epsilon_{n-2})$. In much the same way, $2k+1 = 1 + 2\epsilon_0 + 2^2\epsilon_1 + \cdots + 2^{n-1}\epsilon_{n-2}$, so that $c_{2k+1} = c(1, \epsilon_0, \cdots, \epsilon_{n-2})$. Therefore in both cases $\epsilon_{n-1} = 0$ and $\epsilon_{n-1} = 1$

$$(P_n|c\rangle)(\epsilon_0, \cdots, \epsilon_{n-2}, \epsilon_{n-1}) \;=\; c(\epsilon_{n-1}, \epsilon_0, \cdots, \epsilon_{n-2})\,. \tag{7}$$

6. (a) A $2^n \times 2^n$ matrix of the form

$$A \;=\; \begin{bmatrix} A_0 & 0 \\ 0 & A_1 \end{bmatrix}, \tag{8}$$

   where the A_i's are $2^{n-1} \times 2^{n-1}$ matrices, is denoted $A_0 \oplus A_1$. In particular it gives

$$\left.\begin{aligned} (A|c\rangle)_k \;&=\; \sum_{l=0}^{2^{n-1}-1} (A_0)_k^l\, c_l\,, \\ (A|c\rangle)_{k+2^{n-1}} \;&=\; \sum_{l=0}^{2^{n-1}-1} (A_1)_k^l\, c_{l+2^{n-1}}\,, \end{aligned}\right. \qquad 0 \le k \le 2^{n-1} - 1\,,$$

where the matrix elements $(A_i)_{kl}$ are written with lower and upper indices $(A_i)_k^l$ instead. Using the previous arguments, this last formula can be expressed in tem of the digits as follows

$$(A|c\rangle)(\epsilon_0,\cdots,\epsilon_{n-2},\epsilon_{n-1}) \;=\; \sum_{\eta_0=0}^{1}\cdots\sum_{\eta_{n-2}=0}^{1}\left(A_{\epsilon_{n-1}}\right)_{\epsilon_0,\cdots,\epsilon_{n-2}}^{\eta_0,\cdots,\eta_{n-2}} c(\eta_0,\cdots,\eta_{n-2},\epsilon_{n-1})\,. \tag{9}$$

In other words, such a matrix does not touch the last digit ϵ_{n-1}. This argument applies in particular to the middle matrix in eq. (3) that is $F_{n-1}\oplus F_{n-1}$.

(b) Applying a second time eq. (3) to each of the two F_{n-1}'s appearing above gives a decomposition of the form

$$F_n \;=\; \begin{bmatrix} \mathbf{1}_{2^{n-1}} & D_n \\ \mathbf{1}_{2^{n-1}} & -D_n \end{bmatrix}\cdot\begin{bmatrix} \begin{bmatrix} \mathbf{1}_{2^{n-2}} & D_{n-1} \\ \mathbf{1}_{2^{n-2}} & -D_{n-1} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \mathbf{1}_{2^{n-2}} & D_{n-1} \\ \mathbf{1}_{2^{n-2}} & -D_{n-1} \end{bmatrix} \end{bmatrix}\cdots$$

$$\cdots\begin{bmatrix} F_{n-2} & 0 & 0 & 0 \\ 0 & F_{n-2} & 0 & 0 \\ 0 & 0 & F_{n-2} & 0 \\ 0 & 0 & 0 & F_{n-2} \end{bmatrix}\cdot\begin{bmatrix} [P_{n-1}] & 0 \\ 0 & [P_{n-1}] \end{bmatrix}\cdot[P_n]\,, \tag{10}$$

Hence, iterating j-times eq. (3) will give, in the middle, the direct sum of 2^j terms $F_{n-j}\oplus\cdots\oplus F_{n-j}$.

(c) Using the argument above, such a matrix does not modify the last j-digits of the coordinates namely the $\epsilon_{n-j},\cdots,\epsilon_{n-1}$.

7. (a) Since $\tilde{P}_{n-1}$ has the structure of the A-matrice (8), it does not affect the last digit. Moreover, it acts as P_{n-1} on the previous digits, so that (see eq. (7))

$$\left(\tilde{P}_{n-1}|c\rangle\right)(\epsilon_0,\cdots,\epsilon_{n-2},\epsilon_{n-1}) \;=\; (|c\rangle)(\epsilon_{n-2},\epsilon_0,\cdots,\epsilon_{n-3},\epsilon_{n-1})\,. \tag{11}$$

Similarly $\tilde{P}_{n-2}$ does no modify $\epsilon_{n-2},\epsilon_{n-1}$ and acts like P_{n-2} on the first $(n-2)$-digits, so that

$$\left(\tilde{P}_{n-2}|c\rangle\right)(\epsilon_0,\cdots,\epsilon_{n-2},\epsilon_{n-1}) \;=\; (|c\rangle)(\epsilon_{n-3},\epsilon_0,\cdots,\epsilon_{n-4},\epsilon_{n-2},\epsilon_{n-1})\,. \tag{12}$$

(b) More generally, the same argument leads to

$$\tilde{P}_{n-j}|c\rangle(\epsilon_0,\cdots,\epsilon_{n-j-1},\epsilon_{n-j},\cdots,\epsilon_{n-1}) = |c\rangle(\epsilon_{n-j-1},\epsilon_0,\cdots,\epsilon_{n-j-2},\epsilon_{n-j},\cdots,\epsilon_{n-1}). \tag{13}$$

(c) Thanks to (13), if $\hat{P}_k = \tilde{P}_{n-k+2} \cdots \tilde{P}_n$, an iteration leads to

$$\begin{aligned}
\hat{P}_n|c\rangle(\epsilon_0, \cdots, \epsilon_{n-1}) &= \hat{P}_{n-1}|c\rangle(\epsilon_1, \epsilon_0, \epsilon_2, \cdots, \epsilon_{n-1}) \\
&= \hat{P}_{n-2}|c\rangle(\epsilon_2, \epsilon_1, \epsilon_0, \epsilon_3, \cdots, \epsilon_{n-1}) \\
&= \cdots \\
&= |c\rangle(\epsilon_{n-1}, \epsilon_{n-2}, \cdots \epsilon_1, \epsilon_0)\,.
\end{aligned}$$

8. (a) Thanks to (5), the matrix elements of F_1 are given by $(F_1)_\epsilon^\eta = (-1)^{\epsilon\eta}$ with $\epsilon, \eta \in \{0,1\}$. Moreover, the first step consists in applying $\tilde{F}_1 = F_1 \oplus \cdots \oplus F_1$ (2^{n-1} factors). Thanks to 6.(c), it does not affect the digits $\epsilon_1, \cdots, \epsilon_{n-1}$. Therefore, if $\tilde{F}_1|y_0\rangle = |y_1\rangle$

$$\begin{aligned}
y_1(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) &= \sum_{\eta=0}^{1} (-1)^{\eta\epsilon_0}\; y_0(\eta, \epsilon_1, \cdots, \epsilon_{n-1}) \\
&= y_0(0, \epsilon_1, \cdots, \epsilon_{n-1}) + (-1)^{\epsilon_1}\; y_0(1, \epsilon_1, \cdots, \epsilon_{n-1})
\end{aligned}$$

(b) Let $\tilde{D}_{n-j}$ denote the direct sum $\mathcal{D}_{n-j} \oplus \cdots \oplus \mathcal{D}_{n-j}$ (2^j terms) where

$$\mathcal{D}_{n-j} = \begin{bmatrix} \mathbf{1}_{2^{n-j-1}} & D_{n-j} \\ \mathbf{1}_{2^{n-j-1}} & -D_{n-j} \end{bmatrix}$$

Then $\tilde{D}_1 = \tilde{F}_1$ and the same argument applied to $|y_k\rangle = \tilde{D}_k \tilde{D}_{k-1} \cdots \tilde{D}_1 |y_0\rangle$ gives

$$y_{k+1}(\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}) = \sum_{\eta=0,1} e^{\frac{2\imath\pi\eta(\epsilon_0 + 2\epsilon_1 + \cdots + 2^k \epsilon_k)}{2^{k+1}}} y_k(\epsilon_0, \cdots, \epsilon_{k-1}, \eta, \epsilon_{k+1}, \cdots, \epsilon_{n-1})$$

since D_n is diagonal (see 4.).

(c) Starting from $|c\rangle$, which contains $N = 2^n$ complex numbers, $|y_0\rangle$ consists simply in relabeling the initial datas. For each set of n-digits $\underline{\epsilon} = (\epsilon_0, \cdots, \epsilon_{n-1})$, $|y_{k+1}\rangle$ is obtained from $|y_k\rangle$ by applying one multiplication by an exponential factor and one addition. Thus for each components of $|y_n\rangle$ requires n multiplications and n additions and n exponential factors. Since there are $N = 2^n$ such components, the computation of the Fourier transform of $|c\rangle$ requires $3nN = 3N \ln_2 N$ operations.

A direct application of matrix multiplication, with the matrix of the Fourier transform, requires for each component, N exponential factors, N multiplications and N additions. Thus it gives $3N^2$ operations instead, namely $N/\ln_2 N$ times the number required in the *FFT*. For $n = 20$, $N = 2^{20} = 1024^2 = 1.05 \times 10^6$. Thus the *FFT* will require $N/\ln_2 N = 5 \times 10^4$ less computer time than the ordinary method!

⋆ ⋆ ⋆