1 Szenario

In diesem Tutorial wird auf die Diskrete Form der Fouriertransformation eingegangen. Eine geschlossene Lösung der Fouriertransformation ist auf dem PC nicht immer leicht handzuhaben, und oftmals stehen nur eine Anzahl von Messwerten bereit mit deren Hilfe die Funktion rekonstruiert wird. Da es eine Vielzahl von Anwendungsgebieten für die Fouriertransformation gibt (Bildbearbeitung, Filter, etc..) gibt man sich mit der Standardimplementierung, die eine quadratische Laufzeit besitzt nicht zufrieden.

2 Mathematische Grundlagen

Die diskrete Fouriertransformation ist definiert als:

$$F_k = \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi ijk}{N}}$$

Der exponentielle Anteil $e^{\frac{2\pi ik}{n}}$ kann als komplexe einheitswurzel aufgefasst werden, mit Hilfe deren speziellen Eigenschaften auch eine derartige Auswertung der Fast-Fourier-Transformation(FFT) möglich. Eine n-te Einheitswurzel w_k im komplexen errechnet man wie folgt:

$$w_k = e^{\frac{2\pi ik}{n}}$$

k läuft dabei von 0 bis n. Die FFT ergibt sich aus der DFT nun, indem zunächst eine separate Betrachtung der geraden und der ungeraden Terme erfolgt :

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} f_j w^{jk} \\ &= \sum_{j=0}^{N-1} f_j w^{j2k} + \sum_{j=0}^{N-1} f_j w^{j(2k+1)} \end{aligned}$$

Für den geraden Anteil kann man nun wie folgt Verfahren. Sei $m = N/2$:

$$\begin{aligned} F_{2k} &= \sum_{j=0}^{N-1} f_j w^{j2k} \\ &= \sum_{j=0}^{m-1} f_j w^{j2k} + \sum_{j=0}^{m-1} f_{j+m} w^{(j+m)2k} \\ &= \sum_{j=0}^{m-1} [f_j + f_{j+m}] w^{j2k} \end{aligned}$$

Der letzte Schritt ergibt sich durch die Eigenschaften der Einheitswurzeln wie folgt:

$$w^{(j+m)2k} = w^{j2k+m2k} = w^{j2k} \cdot w^{m2k} = w^{j2k} \cdot w^{Nk} = w^{j2k}$$

Da $w^{Nk} = \left(w^N\right)^k = 1^k = 1$ Setzt man nun noch $v = w^2$:

$$F_{2k} = \sum_{j=0}^{m-1} [f_j + f_{j+m}] v^{jk}$$

Die Fouriertransformation von $F_2 k$ kann also durch die Fouriertransformation des Vektors $[f_j + f_{j+m}]_{j=0}^{m-1}$ berechnet werden. Der rekursive Charackter wird hier also deutlich. Ähnlich kann man auch mit den ungeraden Indizes verfahren:

$$\begin{aligned} F_{2k+1} &= \sum_{j=0}^{N-1} f_j w^{j(2k+1)} \\ &= \sum_{j=0}^{m-1} f_j w^{j(2k+1)} + \sum_{j=0}^{m-1} f_{j+m} w^{(j+m)(2k+1)} \\ &= \sum_{j=0}^{m-1} w^j [f_j - f_{j+m}] w^{j2k} \end{aligned}$$

Der letzte Schritt ergibt sich auch hier über die Eigenschaften der komplexen Einheitswurzeln:

$$w^{(j+m)(2k+1)} = w^{j2k} \cdot w^{Nk} \cdot w^i \cdot w^{N/2} = -w^i \cdot w^{j2k}$$

Da $\left(w^{N/2}\right)^2 = w^N = 1$ muss $w^{N/2} = \pm 1$. Allerdings entfällt $+1$ als mögliche Lösung, da das der definition der Einheitswurzeln widerspricht.

Auch hier zeichnet sich der rekursive Charackter ab. Da die Vektoren nun immer weiter in gerade und ungerade Indizes unterteilt werden, so müssen diese am Ende wieder richtig zusammengesetzt werden, dies kann Beispielsweise wie Folgt erreicht werden:

Algorithmus 1: Prozedur Shuffle

Eingabe : Das vertauschte Array der FFT $a[]$, ein Index In , die Länge des Arrays n

Ausgabe : ein neu angeordnetes Array $a[]$

$m = n/2$;
Complex $b[m]$;
for $i = 0$ **to** m **do**
  $b[i] = a[In + i]$
for $i = 0$ **to** m **do**
  $a[In + i + i + 1] = a[In + i + m]$
for $i = 0$ **to** m **do**
  $a[In + i + i] = b[i]$

Eine Beispielhafte Ausführung des FFT für $n = 8$ zeigt Figur 1:

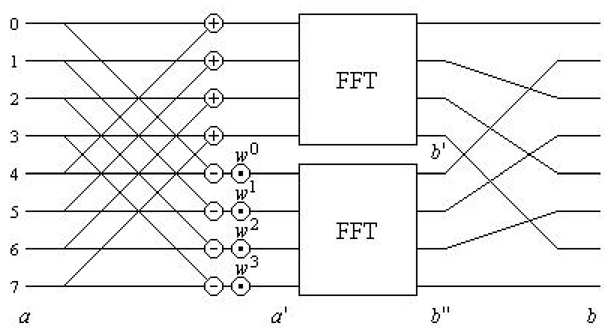

Abbildung 1: Die Fouriertransformation für $n = 8$

Nachdem nun die Grundlagen aufgeklärt wurden kann der FFT-Algorithmus beschrieben werden:

Algorithmus 2: Der FFT-Algorithmus

Eingabe : Ein komplexes Array a , die Länge n, der Startindex In
Ausgabe : Das Array a Fouriertransformiert

```
Complex z, w, h;
w = e^(2πi/n);
if n > 1 then
    m = n/2;
    z = 1;
    for i = In to In + m do
        h = a[i] − a[i − m];
        a[i] = a[i] + a[i + m];
        a[i + m] = h · z;
        z = z · w;
    FFT(a,m,In);
    FFT(a,m,In+m);
    Shuffle(a,n,In);
```

3 Die Inverse Fast-Fourier-Transformation

Es wird uns hier sehr leicht gemacht, denn die Inverse diskrete Fouriertransformation ist definiert als:

$$F_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \cdot e^{\frac{-2\pi ijk}{N}}$$

Es ist also die selbe Prozedur wie die FFT, mit der Ausnahme, dass das Rückgabearray mit dem Faktor $\frac{1}{N}$ skaliert ist, und das anstatt der Einheitswurzel w^j die inverse Einheitswurzel w^{-j} für die Transformation genutzt wird.

4 Beispiel

Die Fouriertransformation soll nun mit einem Beispiel abgerundet werden. Gesucht ist die Fouriertransformation des Arrays a$[1, 1, 1, 1]$. Am leichtesten per Hand berechnen lässt sich die Fouriertransformation über die Fouriermatrix, dies ist eine Matrix W_n der Form:

$$W_n = \begin{pmatrix} w_n^{0\cdot 0} & \dots & w_n^{0\cdot(n-1)} \\ \vdots & \ddots & \vdots \\ w_n^{(n-1)\cdot 0} & \dots & w_n^{(n-1)(n-1)} \end{pmatrix}$$

Mit den Einheitswurzeln $w_n = e^{\frac{2\pi i}{n}}$. Die diskrete Fouriertransformation ergibt sich nun über:

$$F_k = a \cdot W_n$$

In unserem Beispiel ist $n = 4$ und damit ist unsere Einheitswurzel $w_n = e^{\frac{i\pi}{2}} = \cos \pi/2 + i \sin \pi/2 = i$. Wir erhalten dann für die Fouriermatrix:

$$W_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Damit errechnen wir die Fouriertransformierte durch eine Matrix-Vektor-Multiplikation und erhalten den Vektor $\hat{a}[4, 0, 0, 0]$. Dies soll nun auch mit Hilfe der Bibliothek gezeigt werden:

```
Complex c(1,0);
Complex c_arr[4]={c,c,c,c};

FFT(c_arr,4,0); //[4,0,0,0]
```

Die inverse Fouriertransformation soll nun den Ausgangsvektor a$[1, 1, 1, 1]$ zurückliefern:

```
IFFT(c_arr,4,0);
```

Eine Ausgabe macht deutlich, das der Ausgangsvektor a$[1, 1, 1, 1]$ wiederhergestellt wurde. Wir haben nun eine Fouriertransformation die in $\mathcal{O}(n \log n)$ läuft. Opfern mussten wir dafür, dass dieser Algorithmus nur für Arrays der Länge 2^n anwendbar ist. Es muss also die nächste Binärzahl oberhalb des Nyquist-Abtastkriteriums als Länge für das Array gewählt werden, um eine Korrektheit zu gewährleisten.