Trigonometric Interpolation

In many applications we need to recover periodic function from its data, say \( f(t+T) = f(t), T > 0 \) is the period.

Def: for new integer \( T_n \) is the set of trig polynomial

\[
p(t) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kt + b_k \sin kt) \quad T = 2\pi
\]

with coefficients \( a_j, b_j \).

Theorem: Given 2n+1 distinct points in the interval \([0, 2\pi]\) and 2n+1 values \( y_0, \ldots, y_{2n} \in \mathbb{R} \) there is unique polynomial \( p_n(t) \in T_n \) that interpolates the data, i.e.,

\[
p_n(t_j) = y_j, \quad j = 0, \ldots, 2n
\]

Explicit formula is

\[
p_n(t) = \sum_{k=0}^{2n} y_k T_k(t)
\]

\[
T_k(t) = \prod_{\substack{i=0 \\ i \neq k}}^{2n} \frac{\sin \frac{t-t_i}{2}}{\sin \frac{t_k-t_i}{2}}
\]

Complex setting is much better.
Complex Setting of Trigonometric Interpolation

Define \[ E_n(x) = e^{ixn} = (e^i)^n \] trig poly of degree \[ n \]
\[ e^{ix} = \cos x + i\sin x \]

Define the set
\[ T_n = \{ E_0(x), E_1(x), \ldots, E_{N-1}(x) \} \text{ on } [0, 2\pi) \]
real complex

Define the interpolation points (equidistant)
\[ x_j = t_j = \frac{2\pi}{N} \cdot j, \quad j = 0, 1, \ldots, N-1 \]

Interpolation Problem: Find \( p(x) \in T_n \),

1. \[ p(x) = \sum_{k=0}^{N-1} c_k E_k(x), \quad c_k \in \mathbb{C} \text{ - complex numbers} \]
   such that
2. \[ p\left(\frac{2\pi}{N} \cdot j\right) = f\left(\frac{2\pi}{N} \cdot j\right), \quad j = 0, \ldots, N-1 \]

Now we shall derive an explicit formula for the coefficients \( c_k \) of the interpolant (1)
from the interpolation conditions (2).
Some simple facts. Consider $L^2(0, 2\pi)$

$$\langle f, g \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x) \overline{g(x)} \, dx$$

$$\|f\|_2 = \left( \frac{1}{2\pi} \int_0^{2\pi} |f(x)|^2 \, dx \right)^{1/2}$$

One can easily see that

$$\langle E_n, E_n \rangle = \delta_{mn} = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases}$$

Next, we define a discrete inner product

$$\langle f, g \rangle_N = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{g(x_j)}$$

and check the properties

(a) $\langle f, f \rangle_N > 0$

(b) $\langle f, g \rangle_N = \overline{\langle g, f \rangle_N}$

One can easily see that $\langle E_n, E_n \rangle_N = \delta_{nm}$. Now introduce the matrix

$$W = \begin{pmatrix} E_0 & E_1(x_0) & \cdots & E_{N-1}(x_0) \\ \vdots & \ddots & \ddots & \vdots \\ E_0(x_{N-1}) & E_1(x_{N-1}) & \cdots & E_{N-1}(x_{N-1}) \end{pmatrix}$$

If $x = e^{\frac{2\pi i}{N}}$, then

$$w_{jk} = \lambda^{jk}$$

$W \in \mathbb{C}^{N \times N}$
Because of the property \( \langle E_n, E_m \rangle = 0 \) we have

\[
W^T W = NI
\]

where \( I \) is the identity matrix in \( \mathbb{C}^{N \times N} \).

Now, let

\[
c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix}, \quad f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}
\]

Then the interpolation problem (1), (2) becomes

\[
p(x) = f(x) = \sum_{k=0}^{N-1} c_k E_k(x) = f(x) \quad j = 0, 1, \ldots, N-1
\]

could be written in matrix form

(4) \[
Wc = f
\]

Now multiply (4) by \( W^T \) from left to get

\[
W^T Wc = W^T f \quad \Rightarrow \quad c = \frac{1}{N} W^T f
\]

Thus we get the formula

\[
W^T = \begin{bmatrix} -E_0(x_0) & -E_0(x_1) & \cdots & -E_0(x_{N-1}) \\ -E_1(x_0) & -E_1(x_1) & \cdots & -E_1(x_{N-1}) \\ \vdots & \vdots & \ddots & \vdots \\ -E_{N-1}(x_0) & -E_{N-1}(x_1) & \cdots & -E_{N-1}(x_{N-1}) \end{bmatrix}
\]

\[
c_k = \langle W^T f, E_k \rangle = \frac{1}{N} \sum_{j=0}^{N-1} E_k(x_j) f(x_j)
\]

\[
= \frac{1}{N} \langle f, E_k \rangle
\]

Thus we get the formula

(5) \[
p(x) = \sum_{k=0}^{N-1} c_k E_k(x), \quad c_k = \langle f, E_k \rangle \]
Trigonometric Interpolation (Continue, FFT)

The problem of finding \( p(x) = \sum_{k=0}^{N-1} c_k E_k(x) \), where \( E_k(x) = e^{ixk} \)

s.t. \( p(x_j) = f(x_j) \), \( x_j = \frac{2\pi j}{N}, j = 0, ..., N-1 \)

has a unique solution which is expressed in the following closed form

\[
C = \frac{1}{N} \overline{W}^T f \quad \text{or equivalently} \quad c_k = \frac{1}{N} \sum
\]

or equivalently

\[
c_k = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{E}_k(x_j) = \frac{1}{N} \langle f, E_k \rangle
\]

\[
p(x) = \sum_{k=0}^{N-1} c_k E_k(x) - \text{Fornier transform}
\]

Straighforward implementation of these formula will produce the coefficients \( c_k, k=0, ..., N-1 \) for \( N^2 \) long arithmetic operations.
From the formula (5) we see that if we precompute the matrix $W$ we find each coefficient $c_k$ by just $N$ multiplications and $N$-additions. So for all $c_k$ we need $N^2$ multiplications.

However there is a clever implementation of this process that needs only $N \log_2 N$ multi/div.

Let us have a comparison:

$$
\begin{align*}
N & \quad N^2 & \quad N \log_2 N \\
1024 & = 2^{10} & \approx 10^6 & \approx 10^4 \\
16384 & = 2^{14} & \approx 2.7 \times 10^8 & \approx 2.8 \times 10^5 \\
\end{align*}
$$

Now this is done? It is done by recursion.

Consider $N = 2n$ (in general, $N = 2^m$).

**Theorem:** Let $p(x)$ & $q(x)$ be trigonometric polynomials of degree $\leq n-1$ s.t. for $x_j = \frac{2\pi j}{2n}$, $j = 0, 1, \ldots, 2n-1$ interpolate the data $f(x_j)$, $j = 0, 1, \ldots, 2n$ in the following way:

$$
\begin{align*}
p(x_j) &= f(x_j) \\
q(x_j) &= f(x_{j+1}) \\
j = 0, \ldots, n-1
\end{align*}
$$

i.e. $p(x)$ interpolates the data with even index and $q(x)$ interpolates the data with odd index.
\[ \text{Ex} \quad \begin{array}{cccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & x & x & x & x & x & x & x & x & x \\
\end{array} \]

\[ \begin{align*}
0 & - p(x) \\
x & - q(x) \\
\end{align*} \]

Then the polynomial \( P(x) \) of degree \( 2n-1 \) that interpolates \( f \) at \( x_0, x_1, \ldots, x_{2n-1} \) is

\[ P(x) = \frac{1}{2} \left( 1 + e^{i \pi x} \right) p(x) + \frac{1}{2} \left( 1 - e^{i \pi x} \right) q(x + \frac{\pi}{n}) \]

We just need to check whether \( p(x_j) = f(x_j) \) for \( j = 0, \ldots, 2n-1 \).

**Check** First consider even points \( x_j = \frac{2j + 2}{2n} = \frac{2j_n}{n} \)

\[ e^{i \pi x_j} = e^{i \frac{2j_n}{n}} = e^{\frac{2j}{n}} = 1 \]

\[ P(x_j) = \frac{1}{2} \left( 1 + 1 \right) p(x_j) = p(x_j) = f(x_j) \] \( \forall x \)

Next consider odd points \( x_{j+1} = \frac{2j + 2 + \pi}{2n} = \frac{2j + \pi}{n} \)

\[ e^{i \pi x_j} = e^{i \left( 2j + \pi \right)} = e^{i \pi} = -1 \]

\[ P(x_{j+1}) = \frac{1}{2} \left( 1 + (-1) \right) q(x_j) = f(x_{j+1}) \] by construction

Thus the polynomial is computed using two polynomials of degree \( n \) times smaller. This gives the general idea for the computations of the interpolation coefficients.
Theorem: Let \( p(x) = \sum_{j=0}^{n-1} \alpha_j E_j(x) \), \( q(x) = \sum_{j=0}^{n-1} \beta_j E_j(x) \)
and \( P(x) = \sum_{j=0}^{2n-1} \gamma_j E_j(x) \). If \( \alpha_j, \beta_j \) are available, then \( \gamma_j \) are computed by

\[
\gamma_j = \frac{1}{2} \alpha_j + \frac{1}{2} e^{i \frac{\pi}{n}} \beta_j \quad j = 0, 1, \ldots, n-1
\]

\[
\gamma_{j+n} = \frac{1}{2} \alpha_j - \frac{1}{2} e^{i \frac{\pi}{n}} \beta_j
\]

Proof: Indeed, \( E_n(x)E_j(x) = E_{nj}(x) \) we have

\[
e^{inx} e^{ijx} = e^{i(nx+j)x}
\]

Rewrite the above formula:

\[
P(x) = \frac{1}{2} \left( 1 + E_n(x) \right) \sum_{j=0}^{n-1} \alpha_j E_j(x) + \frac{1}{2} \left( 1 - E_n(x) \right) \sum_{j=0}^{n-1} \beta_j E_j(x) e^{i \frac{\pi}{n} j}
\]

\[
P(x) = \frac{1}{2} \sum_{j=0}^{n-1} \left( \alpha_j E_j(x) + \beta_j e^{i \frac{\pi}{n} j} E_j(x) \right)
\]

\[
+ \frac{1}{2} \sum_{j=0}^{n-1} \left( \alpha_j E_{nj}(x) - \beta_j e^{i \frac{\pi}{n} (n+j)} E_{nj}(x) \right) = \sum_{j=0}^{2n-1} \gamma_j E_j(x)
\]

By comparing the coefficients in front of \( E_j(x) \), we get the desired formulas (1).
Now how you work with these and what is the operation count?

First, you precompute the numbers \( \frac{1}{2} e^{-i \frac{2\pi}{n}} = c \).

Then you see that to find the coefficients \( d_j \) you need to:

1. Multiply \( d_j \times \frac{1}{2} \) for \( j = 0, \ldots, n-1 \) \( n \) times.
2. Multiply \( b_j \times c \) for \( j = 0, \ldots, n-1 \) \( n \) times.

Total, \( 2n \) long operations.

Let us denote by \( R(2n) \) the cost of computing the coefficients of the polynomial of degree \( 2n \). Then obviously the count of long arithmetic operations is

\[
R(2n) = 2R(n) + 2n
\]

the cost of computing two polynomials of degree \( n \).

Now take \( N = 2^m \) \( m = \log_2 N \)

\[
R(2^m) = 2R(2^{m-1}) + 2^m = 2 \left[ 2R(2^{m-2}) + 2^{m-1} \right] + 2^m = 2^2 R(2^{m-2}) + 2 \cdot 2^m = 2^m R(1) + m \cdot 2^m \approx m 2^m \approx \left\lceil \frac{N \log_2 N}{2} \right\rceil
\]