MAS565 Numerical Analysis HW2

20218125 채지석

2.13.(a) Using the identity $\sin\varphi = \frac{1}{2i}(e^{i\varphi} - e^{-i\varphi})$ we have

$$t(x) = \prod_{k=1}^{2n} \sin\frac{x-x_k}{2}$$

$$= \prod_{k=1}^{2n} \frac{1}{2i}\left(e^{i\cdot\frac{x-x_k}{2}} - e^{-i\cdot\frac{x-x_k}{2}}\right)$$

$$= \frac{1}{(2i)^{2n}} \prod_{k=1}^{2n}\left(e^{\frac{ix}{2}}\cdot e^{-\frac{ix_k}{2}} - e^{-\frac{ix}{2}} e^{\frac{ix_k}{2}}\right).$$

For convenience let $z_k := e^{-\frac{ix_k}{2}}$ then $e^{\frac{ix_k}{2}}$ is the complex conjugate of z_k, i.e. $e^{\frac{ix_k}{2}} = \bar{z}_k$, and also a multiplicative inverse, i.e. $e^{\frac{ix_k}{2}} = z_k^{-1}$.

Now, observe that

$$t(x) = \frac{1}{(2i)^{2n}} \prod_{k=1}^{2n}\left(e^{\frac{ix}{2}} e^{-\frac{ix_k}{2}} - e^{-\frac{ix}{2}} e^{\frac{ix_k}{2}}\right)$$

$$= \frac{1}{(-4)^n} \prod_{k=1}^{2n}\left(e^{ix/2} z_k + (-1) e^{(-1)\frac{ix}{2}} z_k^{-1}\right)$$

so if we let $\mathcal{H}$ be the set

$$\mathcal{H} := \{ y = (y_1, \dots, y_{2n}) \in \mathbb{R}^{2n} : y_k \in \{\pm 1\},\ k = 1, \dots, 2n\},$$

expanding the product we get

$$t(x) = \frac{1}{(-4)^n} \sum_{y\in\mathcal{H}} \prod_{k=1}^{2n} y_k e^{y_k\cdot\frac{ix}{2}} z_k^{y_k}.$$

The terms in the sum above can be paired so that each

$y \in \mathcal{H}$ gets paired with $-y \in \mathcal{H}$, to obtain

$$t(\alpha) = \frac{1}{(-4)^n} \sum_{\substack{y \in \mathcal{H} \\ y_1 = 1}} \left(\left(\prod_{k=1}^{2n} y_k e^{y_k \cdot \frac{i\alpha}{2}} z_k^{y_k} \right) + \left(\prod_{k=1}^{2n} (-y_k) e^{-y_k \cdot \frac{i\alpha}{2}} z_k^{-y_k} \right) \right)$$

where the terms in the sum now can be further converted into

$$\left(\prod_{k=1}^{2n} y_k e^{y_k \cdot \frac{i\alpha}{2}} z_k^{y_k} \right) + \left(\prod_{k=1}^{2n} (-y_k) e^{-y_k \cdot \frac{i\alpha}{2}} z_k^{-y_k} \right)$$

$$= \left(\prod_{k=1}^{2n} y_k \right) \left(\prod_{k=1}^{2n} e^{y_k \cdot \frac{i\alpha}{2}} z_k^{y_k} + (-1)^{2n} \prod_{k=1}^{2n} e^{-y_k \cdot \frac{i\alpha}{2}} z_k^{-y_k} \right)$$

$$= \left(\prod_{k=1}^{2n} y_k \right) \left(e^{\frac{i\alpha}{2} \sum_{k=1}^{2n} y_k} \prod_{k=1}^{2n} z_k^{y_k} + e^{-\frac{i\alpha}{2} \sum_{k=1}^{2n} y_k} \prod_{k=1}^{2n} z_k^{-y_k} \right). \qquad \cdots (*)$$

Note that $\sum_{k=1}^{2n} y_k$ is always an even integers, as if ν of y_k's are $+1$ then $(2n-\nu)$ of y_k's are -1 so $\sum_{k=1}^{2n} y_k = \nu - (2n-\nu) = 2\nu - 2n$.

Put $j := \frac{1}{2} \sum_{k=1}^{2n} y_k$ then $j \in \mathbb{Z}$. Also, $z_k^{-y_k} = \overline{z_k^{y_k}}$, so putting $\alpha := \prod_{k=1}^{2n} z_k^{y_k}$ we have $\overline{\alpha} = \prod_{k=1}^{2n} z_k^{-y_k}$. Then $(*)$ can be written as

$$\left(\prod_{k=1}^{2n} y_k \right) \left(e^{ij\alpha} \cdot \alpha + e^{-ij\alpha} \cdot \overline{\alpha} \right).$$

Put $\alpha = a + bi$ for $a = \mathrm{Re}(\alpha)$, $b = \mathrm{Im}(\alpha)$, then

$$\begin{aligned} e^{ij\alpha} \alpha + e^{-ij\alpha} \cdot \overline{\alpha} &= e^{ij\alpha}(a+bi) + e^{-ij\alpha}(a-bi) \\ &= a \cdot (e^{ij\alpha} + e^{-ij\alpha}) + bi(e^{ij\alpha} - e^{-ij\alpha}) \\ &= a(2\cos j\alpha) + bi(2i \sin j\alpha) \\ &= 2a \cos j\alpha - 2b \sin j\alpha. \end{aligned}$$

In conclusion we have

$$t(\alpha) = \frac{1}{(-4)^n} \sum_{\substack{y \in \mathcal{H} \\ y_1 = 1}} \left(\left(\prod_{k=1}^{2n} y_k \right) \left(2a \cos j\alpha - 2b \sin j\alpha \right) \right).$$

As $(-4)^n$, $\prod_{k=1}^{2n} y_k$, a, and b are all real, and since $\cos(-jx) = \cos jx$ and $\sin(-jx) = -\sin jx$, to show that the given statement is true it suffices to show that $|j| \le n$, but this is clear from

$$|j| = \frac{1}{2}\left|\sum_{k=1}^{2n} y_k\right| \le \frac{1}{2}\sum_{k=1}^{2n} |y_k| = n.$$

(b) We have $(2n+1)$ support abscissae, and from part (a) it is clear that each t_j's, and therefore its linear combination $T(x)$, is a trigonometric polynomial of the form

$$\frac{1}{2}A_0 + \sum_{j=1}^{n}\left(A_j \cos(jx) + B_j \sin(jx)\right)$$

where $A_0, A_1, \cdots, A_n, B_1, \cdots, B_n$ are all real. By Theorem 2.3.1.12 which asserts the uniqueness of interpolating trigonometric polynomials, it suffices to show that $T(x_k) = y_k$ holds for all $k = 0, 1, \cdots, 2n$.

Since $0 < |x_j - x_k| < 2\pi$ whenever $j \ne k$, the denominator of $t_j(x)$ is never zero, hence well defined. Furthermore, the fact that $t_j(x_j) = 1$ and $t_j(x_k) = 0$ whenever $j \ne k$ is immediate from the definition of $t_j(x)$. Therefore

$$T(x_k) = \sum_{j=0}^{2n} y_j t_j(x_k) = y_k$$

and we are done.

2.18. (a) The Sande – Tukey method performs fast Fourier transform upon the recursion

$$\begin{cases} f_{r,k}^{(m-1)} = f_{r,k}^{(m)} + f_{r,k+M}^{(m)} \\ f_{r+R,k}^{(m-1)} = \left(f_{r,k}^{(m)} - f_{r,k+M}^{(m)}\right) \varepsilon_m^k \end{cases}, \quad \begin{array}{l} m = n, n-1, \cdots, 1 \\ r = 0, 1, \cdots, R-1 = 2^{n-m}-1 \\ k = 0, 1, \cdots, M-1 = 2^{m-1}-1 \end{array}$$

initiated by

$$f_{0,k}^{(n)} = f_k, \qquad k = 0, 1, \cdots, N-1$$

and terminating with

$$f_{r,0}^{(0)} = N\beta_r, \qquad r = 0, 1, \cdots, N-1.$$

Showing that the factorization

$$T = QSP(D_{n-1}SP)\cdots(D_1SP)$$

holds is equivalent to show that

$$N\beta = QSP(D_{n-1}SP)\cdots(D_1SP)f.$$

However the provided definitions of D_i, $i = 1, \cdots, n-1$, are erroneous, as when $n=2$ if we follow the provided definition of D_1 then D_1 becomes a real matrix, hence the product $QSPD_1SP$ also, while T is nonreal. The revision we propose is to change the definition of $\delta_r^{(l)}$ into

$$\delta_r^{(l)} = \exp(-2\pi i \tilde{r}/2^{n-(l-1)}).$$

Also the definition of the permutation matrix P is ambiguous, if not flawed when one follows the conventional definition. The permutation matrix P must be defined as $P = [p_{ij}]$ where

$$p_{ij} = \begin{cases} 1 & \text{if} \quad i = \xi(j) \\ 0 & \text{otherwise} \end{cases}$$

with zero-based indices, so that

$$(Pf)_j = f_{\xi^{-1}(j)}$$

in order to make the proposed factorization of T valid.

With the modified definitions as above, we claim that the factorization

$$T = Q(SP^n)(P^{-(n-1)}D_{n-1}SP^{n-1})\cdots(P^{-2}D_2SP^2)(P^{-1}D_1SP)$$

actually represents Sande-Tukey method with a specific arrangement of the values of r in each step. More specifically, denote the bit-reversal permutation as τ, and $\varphi_m \in \mathbb{C}^N$ be a vector

$$\varphi_m^T = [f^{(m)}_{\tau(0),0}, f^{(m)}_{\tau(0),1}, \cdots, f^{(m)}_{\tau(0),2^m-1}, f^{(m)}_{\tau(1),0}, \cdots, f^{(m)}_{\tau(1),2^m-1}, \cdots, f^{(m)}_{\tau(2^{n-m}-1),2^m-1}]$$

then we claim that $\varphi_{n-m} = P^{-m}D_mSP^m\varphi_{n-m+1}$, $m = 1, 2, \cdots, n-1$. Fix any m, and denote (temporarily, with abuse of notation) the j^{th} entry of φ_{n-m+1} by f_j. By definition of P, we have

$$P^m\varphi_{n-m+1} = [f_0, f_M, f_{2M}, \cdots, f_{(2R-1)M}, f_1, f_{1+M}, \cdots, f_{1+(2R-1)M}, \cdots, f_{M-1}, f_{M-1+M}, \cdots, f_{M-1+(2R-1)M}].$$

Now if S is multiplied, f_{i+2jM} $(0 \le i < M, 0 \le j < R)$ is paired with $f_{i+2jM+M}$ so that they are added and subtracted. Now according to the (modified) definition the matrix D_m can be expressed in the form

$$D_m = \mathrm{diag}(\underbrace{1, \varepsilon_m^0, \cdots, 1, \varepsilon_m^0}_{R \text{ times.}}, \underbrace{1, \varepsilon_m^1, \cdots, 1, \varepsilon_m^1}_{R \text{ times}}, \cdots, \underbrace{1, \varepsilon_m^{M-1}, \cdots, 1, \varepsilon_m^{M-1}}_{R \text{ times}})$$

hence in $D_m S P^m \varphi_{n-m+1}$ we have $f_{i+2jM} + f_{i+2jM+M}$ and $(f_{i+2jM} - f_{i+2jM+M})\,\varepsilon_m^i$, $0 \le i < M$, $0 \le j < R$. Finally P^{-m} sends $f_{i+2jM} + f_{i+M+2jM}$ and $(f_{i+2jM} - f_{i+M+2jM})\,\varepsilon_m^i$ back to the $(i+2jM)^{th}$ and $(i+M+2jM)^{th}$ entry, respectively. So in summary, $P^{-m} D_m S P^m$ transforms φ_{n-m+1} as

$$\begin{cases} (P^{-m} D_m S P^m \varphi_{n-m+1})_{i+2jM} = f_{i+2jM} + f_{i+M+2jM} \\ (P^{-m} D_m S P^m \varphi_{n-m+1})_{i+M+2jM} = (f_{i+2jM} - f_{i+M+2jM})\,\varepsilon_m^i \end{cases}$$

for $0 \le i < M$, $0 \le j < R$. Reverting to $f_{r,k}^{(m)}$ notation, we have

$$\begin{cases} (P^{-m} D_m S P^m \varphi_{n-m+1})_{i+2jM} = f_{\tau(j),i}^{(n-m+1)} + f_{\tau(j),i+M}^{(n-m+1)} = f_{\tau(j),i}^{(n-m)} \\ (P^{-m} D_m S P^m \varphi_{n-m+1})_{i+M+2jM} = \left(f_{\tau(j),i}^{(n-m+1)} - f_{\tau(j),i+M}^{(n-m+1)}\right)\varepsilon_m^i = f_{\tau(j)+R,i}^{(n-m)} \end{cases}$$

for all $0 \le i < M$, $0 \le j < R$. Now that we have

$$(P^{-m} D_m S P^m \varphi_{n-m+1})^T = \left[f_{\tau(0),0}^{(n-m)}, \cdots, f_{\tau(0),M-1}^{(n-m)}, f_{\tau(0)+R,0}^{(n-m)}, \cdots, f_{\tau(0)+R,M-1}^{(n-m)}, \cdots, f_{\tau(R-1)+R,M-1}^{(n-m)}\right], \qquad \cdots (\ast)$$

and to be precise if we let τ_k to denote a bit-reversal permutation considering the input as a k-bit integer then $\tau_k = \tau_k^{-1}$ and

$$\begin{cases} \tau_{n-m+1}(\tau_{n-m}(j)) = 2j \\ \tau_{n-m+1}(\tau_{n-m}(j)+R) = 2j+1 \end{cases}, \quad j = 0, 1, \cdots, 2^{n-m}-1$$

so $(\ast)$ actually reads as

$$\begin{aligned}(P^{-m} D_m S P^m \varphi_{n-m+1})^T &= \left[f_{\tau_{n-m}(0),0}^{(n-m)}, \cdots, f_{\tau_{n-m}(0),M-1}^{(n-m)}, f_{\tau_{n-m}(0)+R,0}^{(n-m)}, \cdots, f_{\tau_{n-m}(0)+R,M-1}^{(n-m)}, \cdots, f_{\tau_{n-m}(R-1)+R,M-1}^{(n-m)}\right] \\ &= \left[f_{\tau_{n-m+1}(0),0}^{(n-m)}, \cdots, f_{\tau_{n-m+1}(0),M-1}^{(n-m)}, f_{\tau_{n-m+1}(1),0}^{(n-m)}, \cdots, f_{\tau_{n-m+1}(1),M-1}^{(n-m)}, \cdots, f_{\tau_{n-m+1}(2R-1),M-1}^{(n-m)}\right] \\ &= \varphi_{n-m}^T.\end{aligned}$$

With all of the observations made up to this point the definition of D_ℓ can be naturally extended to the case where $\ell = n$, and also the definition of φ_m to when $m=0$. Exact same logic up to this point can be applied to conclude that

$$\varphi_0 = P^{-n} D_n S P^n \varphi_1 .$$

But since

$$\begin{aligned}\varphi_0^T &= [f^{(0)}_{\tau(0),0}, f^{(0)}_{\tau(1),0}, \dots, f^{(0)}_{\tau(N-1),0}] \\ &= [N\beta_{\tau(0)}, N\beta_{\tau(1)}, \dots, N\beta_{\tau(N-1)}]\end{aligned}$$

we have $Q\varphi_0 = N\beta$, and as

$$\begin{aligned}\varphi_n^T &= [f^{(n)}_{\tau(0),0}, f^{(n)}_{\tau(0),1}, \dots, f^{(n)}_{\tau(0),N-1}] \\ &= [f_0, f_1, \dots, f_{N-1}] \\ &= f^T\end{aligned}$$

we observe that

$$\begin{aligned}\beta &= \frac{1}{N} Q\varphi_0 \\ &= \frac{1}{N} Q (P^{-n} D_n S P^n)\varphi_1 \\ &= \dots \\ &= \frac{1}{N} Q (P^{-n} D_n S P^n)(P^{-(n-1)} D_{n-1} S P^{n-1}) \cdots (P^{-1} D_1 S P)\varphi_n \\ &= \frac{1}{N} Q P^{-n} (D_n S P)(D_{n-1} S P) \cdots (D_1 S P) f .\end{aligned}$$

Recalling the definitions, it follows that P^n and D_n are both identity matrices, so in conclusion

$$\beta = \frac{1}{N} Q S P (D_{n-1} S P) \cdots (D_1 S P) f$$

and therefore $T = Q S P (D_{n-1} S P) \cdots (D_1 S P)$.

(c) The Cooley–Tukey method is performed as follows. Given $f = [f_0, f_1, \cdots, f_{N-1}]^T$, we divide them into two groups according to the parity of the index, $[f_0, f_2, \cdots, f_{2(N/2-1)}]^T$ and $[f_1, f_3, \cdots, f_{N-1}]^T$.

We perform fast Fourier transform (recursively) on each $\frac{N}{2}$-vectors, obtaining coefficients of the phase polynomial $\beta_{0,j}^{(n-1)}$, $j = 0, 1, \cdots, \frac{N}{2}-1$ and $\beta_{1,j}^{(n-1)}$, $j = 0, 1, \cdots, \frac{N}{2}-1$. Then we use the relation

$$\begin{cases} 2\beta_{0,j}^{(n)} = \beta_{0,j}^{(n-1)} + \beta_{1,j}^{(n-1)} \varepsilon_n^j \\ 2\beta_{0,j+N/2}^{(n)} = \beta_{0,j}^{(n-1)} - \beta_{1,j}^{(n-1)} \varepsilon_n^j \end{cases}$$

where $\varepsilon_n := e^{-\frac{2\pi i}{2^n}}$ and $j = 0, 1, \cdots, \frac{N}{2}-1$, to obtain the coefficients of the discrete Fourier transform of f. For convenience, we denote T_n to be the matrix denoting the discrete Fourier transform on $N = 2^n$ data points, then for

$$\Delta_{n-1} := \mathrm{diag}(\varepsilon_n^0, \varepsilon_n^1, \cdots, \varepsilon_n^{\frac{N}{2}-1})$$

we have the relation

$$T_n = \frac{1}{2}\begin{bmatrix} I & \Delta_{n-1} \\ I & -\Delta_{n-1} \end{bmatrix}\begin{bmatrix} T_{n-1} & 0 \\ 0 & T_{n-1} \end{bmatrix} P^{-1}$$

where P is the bit-cycling permutation matrix defined in (a).

Let S_ℓ denote a block diagonal matrix

$$S_\ell := \begin{bmatrix} I & \Delta_\ell & & & & & \\ I & -\Delta_\ell & & & & & \\ & & I & \Delta_\ell & & & \\ & & I & -\Delta_\ell & & & \\ & & & & \ddots & & \\ & & & & & I & \Delta_\ell \\ & & & & & I & -\Delta_\ell \end{bmatrix}$$

for $l = 1, 2, \cdots, n-1$. Also, let Π_l be the $2^{l+1} \times 2^{l+1}$ permutation matrix denoting a bit-cycling permutation on $(l+1)$-bit integers, and P_l denote a block diagonal matrix

$$P_l = \begin{bmatrix} \overbrace{\Pi_l^{-1} \quad \Pi_l^{-1} \quad \ddots \quad \Pi_l^{-1}}^{2^{n-l-1}\text{ times}} \end{bmatrix}$$

for $l = 1, \cdots, n-1$. Then by the recursive nature of the Cooley-Tukey method, we have

$$\begin{aligned}
T_n &= \frac{1}{2} S_{n-1} \cdot \begin{bmatrix} T_{n-1} & 0 \\ 0 & T_{n-1} \end{bmatrix} \cdot P_{n-1} \\
&= \frac{1}{2} S_{n-1} \cdot \frac{1}{2} \begin{bmatrix} I & \Delta_{n-2} & & \\ I & -\Delta_{n-2} & & 0 \\ & & I & \Delta_{n-2} \\ 0 & & I & -\Delta_{n-2} \end{bmatrix} \begin{bmatrix} T_{n-2} & & & 0 \\ & T_{n-2} & & \\ & & T_{n-2} & \\ 0 & & & T_{n-2} \end{bmatrix} \begin{bmatrix} \Pi_{n-2}^{-1} & 0 \\ 0 & \Pi_{n-2}^{-1} \end{bmatrix} P_{n-1} \\
&= \frac{1}{2^2} S_{n-1} \cdot S_{n-2} \cdot \begin{bmatrix} T_{n-2} & & & 0 \\ & T_{n-2} & & \\ & & T_{n-2} & \\ 0 & & & T_{n-2} \end{bmatrix} P_{n-2} P_{n-1} \\
&= \cdots \\
&= \frac{1}{2^{n-1}} S_{n-1} S_{n-2} \cdots S_1 \cdot \begin{bmatrix} T_1 & & \\ & \ddots & \\ & & T_1 \end{bmatrix} P_1 P_2 \cdots P_{n-1} \\
&= \frac{1}{2^{n-1}} S_{n-1} S_{n-2} \cdots S_1 \cdot \frac{1}{2} S \cdot P_1 \cdots P_{n-1} \\
&= \frac{1}{N} S_{n-1} S_{n-2} \cdots S_1 \, S \cdot P_1 \cdots P_{n-1}
\end{aligned}$$

where S is the matrix defined in part (a). From that

$$\beta = \frac{1}{N} T f = T_n f \Rightarrow T = N T_n$$

we get the factorization of T as

$$T = S_{n-1} S_{n-2} \cdots S_1 S P_1 P_2 \cdots P_{n-1}.$$

2.23. Note that, for any $x \in \Delta$, we have

$$f(x) = S_{\Delta'}(Y'; x) = S_{\Delta}(Y; x).$$

Thus, not only $S_{\Delta}(Y; \cdot)$ is a spline function for f, it is also a spline function for $S_{\Delta'}(Y'; \cdot)$. Further, if $S_{\Delta}(Y; \cdot)$ and $S_{\Delta'}(Y'; \cdot)$ are spline functions for f satisfying either condition (a) or (b), it is clear that $S_{\Delta}(Y; \cdot)$ is a spline function for $S_{\Delta'}(Y'; \cdot)$ satisfying the respective condition. Meanwhile if $S_{\Delta}(Y; \cdot)$ and $S_{\Delta'}(Y'; \cdot)$ are spline functions for f satisfying condition (c) then

$$S_{\Delta}'(Y; a) = f'(a) = S_{\Delta'}'(Y'; a)$$
$$S_{\Delta}'(Y; b) = f'(b) = S_{\Delta'}'(Y'; b)$$

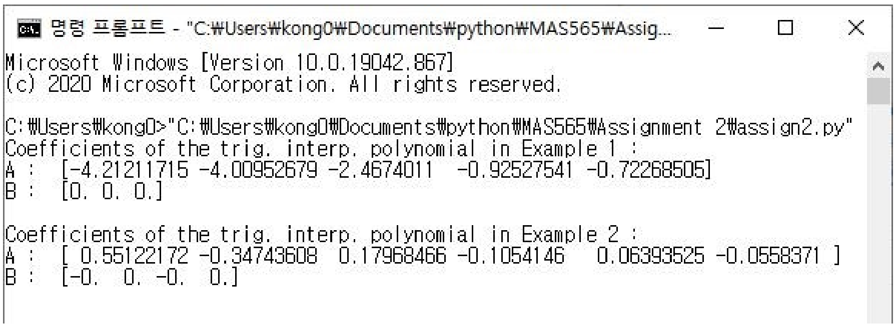
so $S_{\Delta}(Y; \cdot)$ is a spline function for $S_{\Delta'}(Y'; \cdot)$ satisfying condition (c). Therefore, when any of the condition (a), (b), or (c) is satisfied then Theorem 2.4.1.5 asserts that

$$\| S_{\Delta'}(Y'; \cdot) \| \geq \| S_{\Delta}(Y; \cdot) \|.$$

The other inequality is exactly the statement of Theorem 2.4.1.5, so we are done.

Computer Assignment

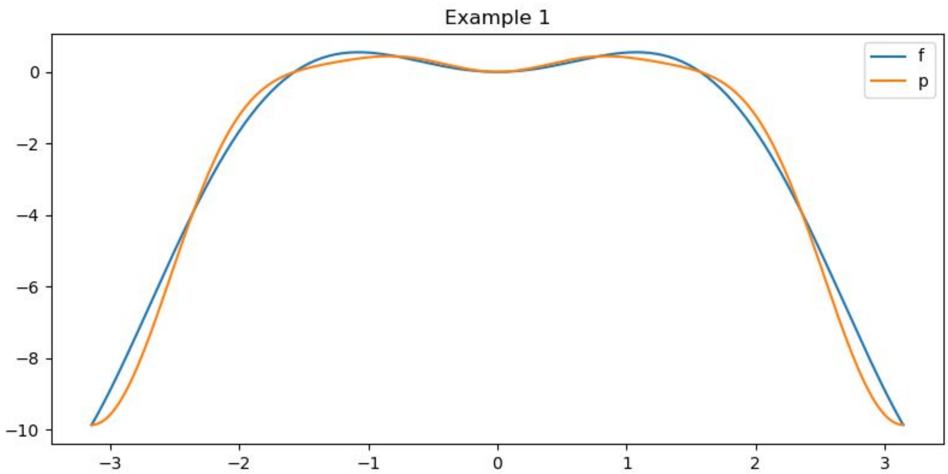
The program which does the required is submitted via KLMS along with this document. Given the two endpoints of the interval `a` and `b`, the step size `h`, and a function `f`, the program computes the trigonometric interpolating polynomial by first computing the coefficients `beta` $= [\beta_k]$ of the phase polynomial using the function `fft` provided by `numpy` package, and then computing the coefficients `A` $= [A_j]$, `B` $= [B_j]$ using the relations between β_k and A_j, B_j. The computed interpolating polynomials are printed out, as the following figure.

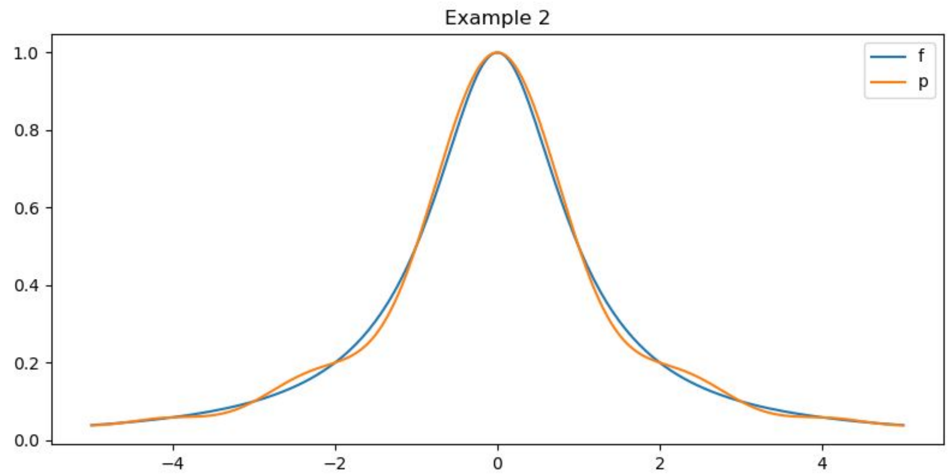


As indicated above, the result say that for example the trigonometric interpolating polynomial for the first function as coefficients

$$\begin{aligned} A_0 &= -4.2121, & A_1 &= -4.0095, & A_2 &= 2.4674, & A_3 &= -0.92528, & A_4 &= -0.72269, \\ & & B_1 &= 0, & B_2 &= 0, & B_3 &= 0. \end{aligned}$$

The original function f and the interpolating polynomial p plotted together, for the two functions given, is shown as in the following figure.





Since both functions given are even functions, we expect that $B_j = 0$ for all j, and indeed that is the case as we can see in the console output.

For the first function the trigonometric interpolating polynomial is nice, but not great. Our guess is that the nature of f, having an M-shaped graph with a very shallow valley, is not well captured with only a few cosine functions. However increasing the number of points in the abscissa shows an improvement in the quality of interpolation.

What is really interesting is the wellness of trigonometric interpolating polynomial interpolating the Runge function. Ordinary polynomials show terrible performance when interpolating the Runge function. But in contrast trigonometric interpolating polynomials well interpolate even the Runge function. Thus, we observe that trigonometric interpolating polynomials have some advantages that ordinary interpolating polynomials do not possess.