Stability and error estimates for Filon–Clenshaw–Curtis rules for highly oscillatory integrals

Güneş Akbaş

0.1 Introduction

In this section, our aim is to explain evaluating integral by some type of discrete Fourier transform for more specific DCT-I and interpolation points are Clenshaw–Curtis points. Then we build up basic interpolation theory and its properties. Our aim is to compute integrals like:

$$\int_{-1}^{1} f(s)e^{iks}ds$$

$N+1$ nods with trigonometric interpolation. Nods are in the interval $[-1,1]$, $cos(\frac{j\pi}{N})$, j=0,1,..., N.

$$f(s) \sim Q_N f(s)$$

$$I_k(f) := \int_{-1}^{1} f(s)e^{iks}ds$$

Actually, our intervals need not be $[-1,1]$. Take for example $\int_a^b g(x)e^{ikx}dx$ Change the variable $x = \alpha + \delta s$, $s \in [-1,1]$.

$$\alpha = \frac{b+a}{2} \qquad \delta = \frac{b-a}{2}$$

$$a \le x \le b \implies a \le \delta s \le b \implies a \le \frac{b+a}{2} + \Big(\frac{b-a}{2}\Big)s \le b$$

$$\implies \frac{a-b}{2} \le \Big(\frac{b-a}{2}\Big)s \le \frac{b-a}{2} \implies -1 \le s \le 1$$

and $g(x) = g(\alpha + \delta s) = f(s)$ we define f as that.

$$e^{iks} = e^{ik\delta s}, \begin{cases} x = \alpha + \delta s \\ dx = \delta ds \end{cases} \tag{1}$$

$$\text{Thus} \quad \int_a^b g(x)e^{iks}dx = \int_{-1}^{1} f(s)e^{iks}e^{ik\delta s}\delta ds$$

$$I_{\delta k}(f) := \int_{-1}^{1} f(s)e^{ik\delta s}dse^{ik\delta} \qquad (s \text{ and } \delta \text{ are constants. })$$

so back to our desired integral
$\int_{-1}^{1} f(s)e^{iks}ds$, we will approximate f(s) with Chebyshev polynomials.

$f(s) \sim Q_N f(s) = \sum_{n=0}^{N}{}'' \alpha_{n,N} T_n(s)$
($\sum''$ means that first and the last terms in the sum are to be halved)

and $T_n(s) = cos(n \arccos(s))$ (First kind of Chebyshev polynomials)

We know that by least squares theory
$\langle f - Q_N f(s), T_n'(s) \rangle = 0$ for n'=0,...,N

so ,

$$\langle f(s), T_n'(s) \rangle - \langle Q_N f(s), T_n'(s) \rangle = 0$$

$$\langle f(s), T_n'(s) \rangle = \langle Q_N f(s), T_n'(s) \rangle = \langle \sum_{n=0}^{N}{}'' \alpha_{n,N} T_n(s), T_n'(s) \rangle$$

$$= \sum_{n=0}^{N} \alpha_{n,N} \langle T_n(s), T_n'(s) \rangle = \alpha_{n,N} = \langle f(s), T_n(s) \rangle$$

Of course to do that our inner product should be defined as orthonormal according to Chebyshev polynomials of first kind to each other

$\langle T_j(s), T_k(s) \rangle = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}$
but we approximate with finite elements (N+1), normally inner-product for Chebyshev polynomials of the first kind defined as:

$$\langle f, g \rangle = \int_{-1}^{1} f(x)g(x)\frac{dx}{\sqrt{1-x^2}}$$

instead of this we use a discrete version (a pseudo-inner-product)

$$\langle T_j, T_k\rangle_N = \frac{2}{N}\sum_{n=0}^{N}{}'' T_j(\cos\frac{n\pi}{N})T_k(\cos\frac{n\pi}{N})$$

$$= \begin{cases} 1 & \text{if } j = k \in \{1, 2, ...N-1\} \\ 2 & \text{if } j = k \in \{0, N\} \\ 0 & \text{if otherwise} \end{cases}$$

$$I_k(f) = \int_{-1}^{1} f(s)e^{iks}ds$$

$$f(s) \sim Q_N f(s)$$

$$I_k(f) \sim I_{k,N}(f) = \int_{-1}^{1} Q_N f(s)e^{iks}ds = \int_{-1}^{1} \sum_{n=0}^{N}{}'' \alpha_{n,N}(f)T_n(s)e^{iks}ds$$

$$= \sum_{n=0}^{N}{}'' \alpha_{n,N}(f)\int_{-1}^{1} T_n(s)e^{iks}ds$$

so we define weight functions as:

$$\omega_n(k) := \int_{-1}^{1} T_n(s)e^{iks}ds$$

, $n \geq 0$

Let C_N be $(N+1) \times (N+1)$ matrix with entries

$$C_{n,j} = (2/N)\cos(jn\pi/N) \quad n, j = 0, ...N$$

give attention for entries $\cos(jn\pi/N) = T_j(n/N)$

This means j'th basis with n'th interpolation point.

$\frac{2}{N}$ is normalizer for base.(We defined our pseudo-inner-product).

Then define:

$$\boldsymbol{\alpha}_N(f) = [\alpha_{0,N}(f), \alpha_{1,N}(f), ...\alpha_{N,N}(f)]^T$$

$$\boldsymbol{f}_N = [f(t_{0,N})/2, f(t_{1,N})..., f(t_{N-1,N}), f(t_{N,n})/2]^T$$

by

$$\alpha_{n,N} = \frac{2}{N}\sum_{n=0}^{N}{}'' \cos(\frac{jn\pi}{N})f(t_{j,N})$$

we can easily see that:

$$\boldsymbol{\alpha}_N(f) = C_N \boldsymbol{f}_N$$

This is a dicrete cosine transform(of type I)
It can be computed by Fast Fourier Transform in $\mathcal{O}(N\log(N))$time
Moreover C_N is symmetric $C_{n,j} = C_{j,n} \implies C_N = C_N^T$
We may write
$I_{k,N}(f) = \sum_{n=0}^{N}{}'' \alpha_{n,N}\omega_n(k) = \boldsymbol{\omega}_n^T \boldsymbol{\alpha}_N(f)$

$$\boldsymbol{\omega}_N = [\omega_0(k)/2, \omega_1(k), ..., \omega_N - 1(k), \omega_N(k)/2]^T$$

then
$\boldsymbol{\omega}_N^T \boldsymbol{\alpha}_N(f) = \boldsymbol{\omega}_N^T C_N \boldsymbol{f}_N = (C_N^T \boldsymbol{\omega}_N)^T \boldsymbol{f}_N = (C_N \boldsymbol{\omega}_N)^T \boldsymbol{f}_N$
Thus, if we precompute $C_N \boldsymbol{\omega}_N$ we don't need further calls for FFT.

0.2 Error for our Interpolation

In this section, we try to get an error bound for our target integral and our quadrature by cosine transform and existing bound for even and periodic functions.
We know that $I_{k,N} = \int_{-1}^{1} Q_N(f)(s)e^{iks}ds$ and $I_k = \int_{-1}^{1} f(s)e^{iks}ds$

f on [-1,1] with cosine transform $f_c(\theta) = f(\cos\theta)$ for $\theta \in \mathbb{R}$

$$f_c(\theta) \quad \text{even}, 2\pi \text{ periodic function}$$

$$(Q_N f)_c(s) = \sum_{n=0}^{N}{}'' \alpha_{n,N} T_n(\cos s)$$

$$f : [-1,1] \to \mathbb{R} \quad , f_c : \mathbb{R} \to \mathbb{R} \quad , t_{j,N} = \cos\frac{\pi j}{N} \quad \implies f(\cos\frac{\pi j}{N})$$

$$f(\cos\frac{\pi j}{N}) = f_c(\frac{\pi j}{N})$$

$$f_c(0), f_c(\frac{\pi}{N}), f_c(\frac{2\pi}{N}) ..., f_c(\pi)$$

$$f_c \sim (Q_N f)_c = Q_N f(\cos\theta)$$

$$= \sum_{n=0}^{N}{}'' \alpha_{n,N} T_n(\cos\theta) \quad (T_n(\cos\theta) = \cos n\theta)$$

We see that $\quad (Q_N f)_c \in span\{1, \cos\theta, \cos 2\theta, ... \cos N\theta\}$

This is the even trigonometric polynomial of degree N that interpolates f_c at N+1 equally spaced points. There is error analysis for such interpolants: $0 \le \mu \le \nu$ and $\nu \ge \nu_0 > 1/2$ there is a constant $C_{\nu_0.\mu}$ such that:

$$\|f_c - (Q_N f)_c\|_{H^\mu} \le C_{\nu_0.\mu} N^{\mu-\nu} \|f_c\|_{H^\nu} \quad \text{for all } N \ge 2$$

where norm is defined as:

$$\|\varphi\|^2 := |\hat{\varphi}(0)|^2 + \sum_{m=0} |m|^{2\nu} |\hat{\varphi}(m)|^2 \quad , \hat{\varphi}(m) := \frac{1}{2\pi} \int_{-\pi}^{\pi} \varphi(\theta)\ e^{-im\theta} d\theta$$

when $\quad \nu = 0 \quad \|.\|_{H^\nu}$ is equivalent to the $\quad L^2(-\pi,\pi)$norm.

$$L^2(-\pi,\pi) \quad \text{norm} \quad \|\varphi\|^2_{L^2(-\pi,\pi)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\varphi(x)|^2 dx$$

by Parseval's identity::

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\varphi(x)|^2 dx = \sum_{n=-\infty}^{\infty} |\hat{\varphi}(m)|^2$$

From this we are looking for our main error bound :

Theorem 2.2 : for r=0,1,2 and for all $\nu \ge \nu_0 > \max\{1/2, p(r)\}$ there exist $C_{\nu_0} > 0$ such that for all $k > 0$

$$|I_k(f) - I_{k,N}(f)| \le C_{\nu_0} (\frac{1}{k^r}) (\frac{1}{N})^{\nu - p(r)} \|f_c\|_{H^\nu} \quad (2.7)$$

where ,

$$p(r) = \begin{cases} 0, & r = 0 \\ 1, & r = 1 \\ 7/2, & r = 2 \end{cases}$$

besides (2.7) holds also for $\nu = r = 1$
Proof:Define even 2π-periodic error function

$$e_N := f_c - (Q_N f)_c \quad \text{(Chebyshev points include the end points} \pm 1)$$

$$f_c(\theta) = f(\cos\theta) \quad (Q_N f)_c(\theta) = (Q_N f)(\cos\theta)$$

We know that:

$$Q_N f(t_{j,N}) = f(t_{j,N}) \quad t_{j,N} = \cos\frac{j\pi}{N} \qquad \text{j=0,1,...,N}$$

$$(Q_N f)(\cos\frac{j\pi}{N}) = (Q_N f)_c(\frac{j\pi}{N}) = f(\cos\frac{j\pi}{N}) = f_c(\frac{j\pi}{N}) \quad \text{for} \quad j = 0 \quad \text{and} \quad j = N$$

$$(Q_N f)(0) = f_c(0) \quad \text{and} \quad (Q_N f)_c(\pi) = f_c(\pi)$$

Hence ,

$$e_N(0) = e_N(\pi) = 0$$

$$I_k(f) - I_{k,N}(f) = \int_{-1}^{1} (f - Q_N f) e^{iks} ds$$

with cosine transform $f(\cos\theta) = f_c(\pi)$ $s = \cos\theta$ $ds = -\sin\theta d\theta$
$-1 \leq s \leq 1$
$-1 \leq \cos\theta \leq 1 \rightarrow \arccos -1 \geq \theta \geq \arccos 1$
Hence,

$$e^{iks\cos\theta} \sin\theta d\theta = \int_0^\pi e_N(\theta) e^{ik\cos\theta} \sin\theta d\theta$$

$$\int_0^\pi e_N(\theta) e^{ik\cos\theta} \sin\theta d\theta = -\int_0^\pi e'_N(\theta) e^{ik\cos\theta} \frac{(-1)}{ik} d\theta + \frac{(-1)}{ik} e_N(\theta) e^{ik\cos\theta} \Big|_0^\pi$$

and

$$\frac{(-1)}{ik} e_N(\theta) e^{ik\cos\theta} \Big|_0^\pi = 0$$

(Remember that $e_N(0) = e_N(\pi)$)
so

$$\int_0^\pi e_N(\theta)e^{ik\cos\theta}\sin\theta d\theta = \frac{1}{ik}\int_0^\pi e'_N(\theta)e^{ik\cos\theta}d\theta$$

theorem 2.2 proof: ... for r=0

$$\|f_c - (Q_N f)_c\|_{H^\mu} \leq C_{\nu_0.\mu}N^{\mu-\nu}\|f_c\|_{H^\nu} \quad \text{for all} \quad N \geq 2$$

$$\text{when} \quad \mu = 0 \quad \text{we take} \quad r = \mu$$

$$\|f_c - (Q_N f)_c\|_{H^0} \leq C_0\left(\frac{1}{N}\right)^\nu \|f_c\|_{H^0}$$

We know that bu parseval's identity this norm equivalent $L^2(-\pi,\pi)$ norm

$$\|f_c - (Q_N f)_c\|_{H^0} = \|f_c - (Q_N f)_c\|_{L^2(-\pi,\pi)} = \left(\frac{1}{2\pi}\int_{-\pi}^{\pi}|f_c - (Q_N f)_c|^2 d\theta\right)^{\frac{1}{2}}$$

By Hölder's inequality:

$$\int_S |f(x)g(x)|\mathrm{d}x \leq \left(\int_S |f(x)|^p\ \mathrm{d}x\right)^{\frac{1}{p}}\left(\int_S |g(x)|^q\ \mathrm{d}x\right)^{\frac{1}{q}}$$

$$p = q = 2$$

$$f_c - (Q_N f)_c = e_N(\theta) \quad \text{choose} \quad e_N(\theta) = f \quad \text{and} \quad g(x) = 1$$

$$\int_{-\pi}^{\pi}|e_N(\theta)|\mathrm{d}\theta \leq \left(\int_{-\pi}^{\pi}|f(x)|^2\ \mathrm{d}\theta\right)^{\frac{1}{2}}\left(\int_{-\pi}^{\pi}|1|^2\ \mathrm{d}\theta\right)^{\frac{1}{2}}$$

$$\text{so we know that} \quad |e_N(\theta)| \geq 0$$

$$\int_0^\pi |e_N(\theta)|\mathrm{d}\theta \leq \int_{-\pi}^{\pi}|e_N(\theta)|\mathrm{d}\theta$$

$$\int_0^\pi |e_N(\theta)||1||1|\mathrm{d}\theta \geq \int_0^\pi \left|e_N(\theta)e^{iks\cos\theta}\sin\theta\right|\mathrm{d}\theta$$
$$= |I_k(f) - I_{k,N}(f)|$$

For $r = 2$ a bit more work is required. First observe that $e'_N(\theta) = -(f - Q_N f)(\cos\theta)\sin\theta$, so $e'_N(\theta)$ also vanishes at 0 and π. Hence, we can introduce the function

$$\varphi_N(\theta) = \frac{e'_N(\theta)}{\sin\theta} = -(f - Q_N f)'(\cos\theta)$$

into (2.10) and then perform another integration by parts to obtain

$$\begin{aligned} I_k(f) - I_{k,N}(f) &= \frac{1}{k^2}\left[\varphi_N(\theta)\exp(\mathrm{i}k\cos\theta)|_0^\pi - \int_0^\pi \varphi'_N(\theta)\exp(\mathrm{i}k\cos\theta)\mathrm{d}\theta\right] \\ &= \frac{1}{k^2}\left[E_1 - E_2\right]. \end{aligned}$$

L'Hôpital's rule shows that $\varphi_N(0) = e''_N(0)$ and $\varphi_N(\pi) = -e''_N(\pi)$ and so, using the Sobolev embedding theorem and (2.5) again,

$$|E_1| \leqslant |e''_N(0)| + |e''_N(\pi)| \leqslant C \, \|e_N\|_{H^3} \leqslant C_{v_0} \frac{1}{N^{v-3}} \, \|f_{\mathrm{c}}\|_{H^v}$$

for all $v \geqslant v_0 \geqslant 3$. To estimate E_2 we write e_N as a cosine series

$$e_N(\theta) = \sum_{m=1}^{\infty} \widehat{e}_N(m) \cos m\theta$$

where

$$\hat{e}_N(m) = \begin{cases} \frac{1}{\pi} \int_0^\pi e_N(\theta) \mathrm{d}\theta, & m = 0, \\ \frac{2}{\pi} \int_0^\pi e_N(\theta) \cos m\theta \mathrm{d}\theta, & m \geqslant 1. \end{cases}$$

Hence,

$$\varphi_N(\theta) = -\sum_{m=1}^{\infty} m \hat{e}_N(m) \frac{\sin m\theta}{\sin\theta}$$

Then with σ denoting the bounded $\mathcal{C}^\infty$ function $\sigma(\theta) := (\sin\theta)/\theta$ we have $\sigma(\theta) \geqslant 2/\pi$ for $\theta \in [-\pi/2, \pi/2]$

$$\textbf{Proof of:} \quad \frac{\sin\theta}{\theta} \geq \frac{2}{\pi} \quad \text{for all} \quad \theta \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$$

Let

$$f(x) = \frac{\sin x}{x} \Longrightarrow f'(x) = \frac{g(x)}{x^2} \text{ with } \quad g(x) = x\cos x - \sin x$$

and

$$g'(x) = -x \sin x \leq 0$$

For $x \in \left[0, \frac{\pi}{2}\right)$, we have $g'(x) \leq 0$, then g is decreasing whereas $g(0) = 0$. Thus $g(x) \leq 0$ on this interval. As a result, $f'(x) \leq 0$ too, hence f is decreasing on $\left[0, \frac{\pi}{2}\right]$. That is

$$\frac{\sin x}{x} = f(x) \geq f(\pi/2) = \frac{2}{\pi} \quad \text{for} \quad 0 \leq x \leq \pi/2$$

so for $\theta \in [0, \pi/2]$ and $m \geqslant 1$,

$$\left| \left(\frac{\sin m\theta}{\sin\theta} \right)' \right| = \left| m \left(\frac{\sigma(m\theta)}{\sigma(\theta)} \right)' \right| = \left| m^2 \frac{\sigma'(m\theta)}{\sigma(\theta)} - m \frac{\sigma(m\theta)\sigma'(\theta)}{\sigma^2(\theta)} \right| \leqslant C m^2$$

for some constant C. Moreover, writing

$$\frac{\sin m\theta}{\sin\theta} = (-1)^{m-1} \frac{\sin m(\theta - \pi)}{\sin(\theta - \pi)}$$

allows us to extend (2.14) to $\theta \in [0, \pi]$. Therefore,

$$|E_2| \leqslant \pi \, \|\varphi'_N\|_{L^\infty(0,\pi)} \leqslant C \sum_{m=1}^{\infty} m^3 \, |\widehat{e}_N(m)| \, .$$

To complete the estimate on E_2 we recall the elementary estimates

$$\sum_{m=1}^{N} m^6 < \frac{(N+1)^7}{7} \quad \text{and} \quad \sum_{m=N+1}^{\infty} \frac{1}{m^{1+\alpha}} < \frac{1}{\alpha N^{\alpha}}, \quad \alpha > 0.$$

Then, splitting the sum (2.15) for $m \leqslant N$ and $m \geqslant N+1$, using the Cauchy-Schwarz inequality, (2.5) and (2.6), we deduce for all $v \geqslant v_0 > 7/2$

$$\begin{aligned}
|E_2| &\leqslant C \left\{ \left[\sum_{m=1}^{N} m^6 \right]^{1/2} \left[\sum_{m=1}^{N} |\widehat{e}_N(m)|^2 \right]^{1/2} + \left[\sum_{m=N+1}^{\infty} \frac{1}{m^{2v-6}} \right]^{1/2} \left[\sum_{m=N+1}^{\infty} m^{2v} |\widehat{e}_N(m)|^2 \right]^{1/2} \right\} \\
&\leqslant C \left\{ N^{-1(-7/2)} \|e_N\|_{H^0} + N^{-1(v-7/2)} \|e_N\|_{H^v} \right\} \\
&\leqslant C_{v_0} N^{-1(v-7/2)} \|f_c\|_{H^v}
\end{aligned}$$

with C denoting a generic constant. Combining the estimates for E_1 and E_2 yields the result.
The estimate of Theorem 2.2 is not optimal when k is small. However, it can easily be extended as in the following corollary.
COROLLARY 2.3 Under the conditions of Theorem 2.2 , for $r = 0, 1, 2$ and for all $v \geqslant v_0 > \max\{1/2, \rho(r)\}$, there exists $C_{10} > 0$

$$|I_k(f) - I_{k,N}(f)| \leqslant C_{v_0} \min\left\{1, k^{-r}\right\} N^{-1(v-\rho(r))} \|f_c\|_{H^v}$$

Proof. The result is clear from Theorem 2.2 when $k \geqslant 1$. When $k < 1$ it follows from (2.9) that

$$|I_k(f) - I_{k,N}(f)| \leqslant \sqrt{\pi} \|e_N\|_{L^2(0,\pi)}$$

which yields the result.

0.3 Accurate computation of weights

In this section, we deduce the algorithm by using basic analysis methods to weight functions and divide the algorithm for $n > k$ and $n \leq k$

$$\omega_n(k) := \int_{-1}^{1} T_n(s) e^{iks} ds$$

$T_n(s)$ is Chebyshev Polynomials of first kind
$U_n(s)$ is Chebyshev Polynomials of second kind
We know that:

$$2T_n(s) = U_n(s) - U_{n-2}(s)$$

and

$$U_n(s) = \frac{1}{n+1} T'_{n+1}(s)$$

$$\int_{-1}^{1} 2T_n(s) e^{iks} ds = 2\omega_n(k) = \int_{-1}^{1} [U_n(s) - U_{n-2}(s)] e^{iks} ds$$

$$= \int_{-1}^{1} U_n(s)e^{iks} - \int_{-1}^{1} U_{n-2}(s)e^{iks}ds$$

$$= \int_{-1}^{1} \frac{1}{n+1}T'_{n+1}(s)e^{iks} - \int_{-1}^{1} \frac{1}{n-1}T'_{n-1}(s)e^{i}ksds = \rho_{n+1}(k) - \rho_{n-1}(k)$$

$$\rho_n(k) := \int_{-1}^{1} U_{n-1}(s)e^{iks}ds = \frac{1}{n}\int_{-1}^{1} T'_n(s)e^{iks}ds$$

$n \geq 1$
By integration by parts:

$$\omega_n(k) = \int_{-1}^{1} T_n(s)e^{iks}ds = \frac{-1}{ik}\int_{-1}^{1} T'_n(s)e^{iks}ds + \frac{T_n(s)e^{iks}}{ik}\Big|_{-1}^{1}$$

$$\omega_n(k) = \frac{-\rho_n(k)n}{ik} + T_n(s)e^{iks}\Big|_{-1}^{1}$$

$$\gamma_n(k) := T_n(s)\frac{e^{iks}}{ik}\Big|_{-1}^{1} = T_n(1)\frac{e^{ik.1}}{ik} - T_n(-1)\frac{e^{-ik.1}}{ik}$$

$$\cos n \arccos 1 \frac{e^{ik}}{ik} - \cos n \arccos -1 \frac{e^{-ik}}{ik} = \frac{1.e^{ik}}{ik} - (-1)^n\frac{e^{-ik}}{ik} = \frac{e^{ik} - (-1)^n e^{-ik}}{ik}$$

$$= \frac{\cos k + i\sin k - ((-1)^n(\cos k - i\sin k))}{ik}$$

$$= \begin{cases} \frac{2\sin k}{k} & \text{even n} \\ \frac{2\cos k}{ik} & \text{odd n} \end{cases}$$

we said that:

$$\omega_n(k) = \frac{-\rho_n(k).n}{ik} + T_n(s)\frac{e^{iks}}{ik}\Big|_{-1}^{1}$$

$$= \gamma_n(k) - \frac{n}{ik}\rho_n(k)$$

$$n \geq 1$$

$$2\omega_n(k) = 2\gamma_n(k)\frac{-2n}{ik}\rho_n(k) = \rho_{n+1}(k) - \rho_{n-1}(k)$$

$$\Rightarrow 2\gamma_n(k) - \frac{2n}{ik}\rho_n(k) = \rho_{n+1}(k) - \rho_{n-1}(k) \quad n \geq 2$$

$$U_0(s) = 1 \quad U_1(s) = 2s \quad \text{so we have}$$

$$\rho_1(k) = \int_{-1}^{1} e^{iks}ds = \gamma_0(k) \quad \rho_2(k) = \int_{-1}^{1} 2se^{iks}ds = 2\gamma_1(k) - \frac{2\gamma_0(k)}{ik}$$

We have starting point for our recursion and recurrence relation for any n so we have an algorithm for phase 1.
Hence, our algorithm is:

Algorithm: for $n \leqslant \min[N, k]$ (first phase)
- Compute

$$\rho_1(k) := \gamma_0(k),$$
$$\rho_2(k) := 2\gamma_1(k) - \frac{2}{ik}\gamma_0(k),$$
$$\rho_{n+1}(k) := 2\gamma_n(k) - \frac{2n}{ik}\rho_n(k) + \rho_{n-1}(k),$$
$$n = 2, \ldots, \min[N, k] - 1.$$

- Set

$$\omega_0(k) = \rho_1(k), \quad \omega_n(k) := \gamma_n(k) - \frac{n}{ik}\rho_n(k), \quad n = 1, 2, \ldots, \min[N, k\}$$

Afterward, we need to form an algorithm for $n > k$, due to stability problem.

$$n_0 = \lceil k \rceil \quad (\lceil \quad \rceil \quad \text{ceiling function})$$
$$M \geq n_0$$

Tridiagonal matrices is used for three-term recurrence relation, to see this we will do one-row and one column matrice multiplication

$$A_M(k) := \begin{bmatrix} \frac{2n_0}{\text{ik}} & 1 & & & \\ -1 & \frac{2(n_0+1)}{\text{ik}} & 1 & & \\ & -1 & \frac{2(n_0+2)}{\text{ik}} & 1 & \\ & \ddots & \ddots & \ddots & \\ & & & & \\ & & & \frac{2(2M-1)}{\text{ik}} & \end{bmatrix}, \quad \mathbf{b}_M(k) := \begin{bmatrix} 2\gamma_{n_0}(k) + \rho_{n_0-1}(k) \\ 2\gamma_{n_0+1}(k) \\ 2\gamma_{n_0+2}(k) \\ \vdots \\ 2\gamma_{2M-1}(k) - \rho_{2M}(k) \end{bmatrix}.$$

Clearly,

$$\boldsymbol{\rho}_M(k) := [\rho_{n_0}(k)\rho_{n_0+1}(k)\rho_{n_0+2}(k), \ldots, \rho_{2M-1}(k)]^{\mathrm{T}}$$

is a solution of

$$A_M(k)\mathbf{x} = \mathbf{b}_M(k)$$

$$A_M(k)_{ij} = \begin{cases} \frac{2(n_0+i-1)}{ik}, & i = j \\ -1 & i-1 = j \\ \ \ 1 & i+1 = j \end{cases} \quad , \mathbf{b}_M(k)_i = 2\gamma_{n_0+i-1}(k)$$

$$\boldsymbol{\rho}_M(k)_i = \rho_{n_0+i-1} \quad i \geq 2M-1$$

$$\text{take i'th row of} \quad A_M(k)$$

$$A_m(k)_{ii} = \frac{2(n_0+i-1)}{ik}, \quad A_M(k)_{i,i-1} = -1, \quad A_m(k)_{i,i+1} = 1$$

$$A_M(k)\boldsymbol{\rho}_M(k) \to A_m(k)_{i,i-1}\boldsymbol{\rho}_M(k)_{i-1} + A_M(k)_{i,i}\boldsymbol{\rho}_M(k)_i + A_M(k)_{i,i+1}\boldsymbol{\rho}_M(k)_{i+1}$$

$$= -\rho_{n_0+i-2}(k) + \frac{2(n_0+i-1)}{ik}\rho_{n_0+i-1}(k) + \rho_{n_0+i}(k) = 2\gamma_{n_0+i-1}(k)$$

THEOREM 3.1 Let M be an integer with $M \geqslant k$ and define

$$p_0(\theta) := \frac{1}{(2M - k\sin\theta)}, \quad p_r(\theta) := p_0(\theta)\frac{\mathrm{d}}{\mathrm{d}\theta}p_{r-1}(\theta), \quad r = 1, 2, \ldots$$

Then

$$\rho_{2M}(k) = 2\mathrm{i}\left[\sum_{r=0}^{J}(-1)^r p_{2r}(0)\sin k + \sum_{r=0}^{J}(-1)^r p_{2r+1}(0)\cos k\right] + R_J(M,k) \quad (3.1)$$

where

$$|R_J(M,k)| \leqslant C_J k M^{-2J-4},$$

and C_J is independent of M and k.

Algorithm: for $k < u \leqslant N$ (second phase)
Set $n_0 = \lceil k \rceil$. - Take $M \geqslant \max\{n_0/2, N/2\}$ sufficiently large and compute $\rho_2 M(k)$ using (3.10). - Construct $A_M(k), \mathbf{b}_M(k)$ as in (3.9) and solve

$$A_M(k)\rho_M(k) = \mathbf{b}_M(k)$$

with

$$\boldsymbol{\rho}_M(k) := [\rho_{n_0}(k)\rho_{n_0+1}(k)\rho_{n_0+2}(k), \ldots, \rho_{2M-1}(k)]^{\mathrm{T}}$$

- Set

$$\omega_n(k) := \gamma_n(k) - \frac{n}{\mathrm{i}k}\rho_n(k), \quad n = n_0, \ldots, N.$$

REMARK 3.2 Note that U_{n-1} is even (respectively odd) when n is odd (respectively even) and so by definition of $\rho_n(k)$ in (3.2), $\rho_n(k)$ is real for odd n and purely imaginary for even n. Hence, defining

$$\breve{\rho}_n = \operatorname{Re}\rho_n + \operatorname{Im}\rho_n$$

we can rewrite (3.4) in real arithmetic,

$$2\breve{\gamma}_n(k) - \frac{2n(-1)^n}{k}\breve{\rho}_n(k) = \breve{\rho}_{n+1}(k) - \breve{\rho}_{n-1}(k), \quad n \geqslant 2$$

The first seven coefficients in the asymptotic expansion (3.10) are given by

$$
\begin{aligned}
p_0(0) &= \frac{1}{2M},\\
p_1(0) &= \frac{k}{(2M)^3},\\
p_2(0) &= \frac{3k^2}{(2M)^5},\\
p_3(0) &= \frac{\left(15k^2-4M^2\right)k}{(2M)^7},\\
p_4(0) &= \frac{\left(105k^2-60M^2\right)k^2}{(2M)^9},\\
p_5(0) &= \frac{\left(945k^4-840k^2M^2+16M^4\right)k}{(2M)^{11}},\\
p_6(0) &= \frac{\left(-12600k^2M^2+1008M^4+10395k^4\right)k^2}{(2M)^{13}}.
\end{aligned}
$$

0.4 An asymptotic expansion

In this section our only aim is to prove the asymptotic expansion of $\rho_2 M(k)$ for large M.
First, note that after applying the change of variables $s=\cos\theta$ in (3.2)

$$\rho_{2M}(k)=\int_{-1}^{1}U_{2M-1}(s)\exp(\mathrm{i}ks)\mathrm{d}s=\int_0^\pi \exp(\mathrm{i}k\cos\theta)\sin(2M\theta)\mathrm{d}\theta,$$

where we have used the fact that

$$U_n(\cos\theta)=\frac{\sin(n+1)\theta}{\sin\theta}$$

Clearly,

$$
\begin{aligned}
\rho_{2M}(k) &= -\frac{\mathrm{i}}{2}\left[\int_0^\pi \exp(\mathrm{i}k\cos\theta)(\exp(2\mathrm{i}M\theta)-\exp(-2\mathrm{i}M\theta))\mathrm{d}\theta\right]\\
&=: -\frac{\mathrm{i}}{2}\left[I_k^+(M)-I_k^-(M)\right],
\end{aligned}
$$

where

$$I_k^\pm(M):=\int_0^\pi \exp\left(\mathrm{i}S_\pm(\theta)\right)\mathrm{d}\theta,\quad S_\pm(\theta):=(\pm 2M\theta+k\cos\theta).$$

(We hide the dependence of $S_\pm$on M and k to simplify forthcoming expressions.) The two following families of smooth functions will be relevant in the sequel:

$$p_0^{\pm}(\theta) := \frac{1}{\mathrm{i}S'_{\pm}(\theta)} = \frac{1}{\mathrm{i}(\pm 2M - k\sin\theta)},$$
$$p_r^{\pm}(\theta) := \frac{1}{\mathrm{i}S'_{\pm}(\theta)} \frac{\mathrm{d}p_{r-1}^{\pm}(\theta)}{\mathrm{d}\theta}, \quad r = 1, 2 \dots$$

Note that $p_r^{\pm}(\theta)$ is real for odd r and purely imaginary for even r. LEMMA 4.1 For all $r \geqslant 0$,

$$p_r^{\pm}(\theta) = \frac{q_r^{\pm}(\theta)}{\mathrm{i}^{r+1}\left(S'_{\pm}(\theta)\right)^{2r+1}}$$

where $q_r^{\pm}(\theta)$ is a trigonometric polynomial in θ defined recursively by

$$q_0^{\pm} \equiv 1, \quad q_{r+1}^{\pm} = \left(q_r^{\pm}\right)' S'_{\pm} - (2r+1) q_r^{\pm} S''_{\pm}, \quad r = 0, 1, 2, \dots$$

In addition

$$p_r^{\pm}(0) = (-1)^r p_r^{\pm}(\pi),$$
$$p_r^{+}(0) = -p_r^{-}(0), \quad p_r^{+}(\pi) = -p_r^{-}(\pi)$$

Proof. Equation (4.3) is clearly true for $r = 0$. Assume that it has been proved up to r. Then

$$\begin{aligned} p_{r+1}^{\pm}(\theta) &= \frac{1}{\mathrm{i}S'_{\pm}(\theta)} \frac{\mathrm{d}}{\mathrm{d}\theta}\left[\frac{q_r^{\pm}(\theta)}{\mathrm{i}^{r+1}\left(S'_{\pm}(\theta)\right)^{2r+1}}\right] \\ &= \frac{1}{\mathrm{i}^{r+2}S'_{\pm}(\theta)}\left[\frac{\left(q_r^{\pm}\right)'(\theta)}{\left(S'_{\pm}(\theta)\right)^{2r+1}} - \frac{(2r+1)q_r^{\pm}(\theta)S''_{\pm}(\theta)}{\left(S'_{\pm}(\theta)\right)^{2r+2}}\right] \\ &= \frac{1}{\mathrm{i}^{r+2}\left(S'_{\pm}(\theta)\right)^{2r+3}}\left[\left(q_r^{\pm}\right)'(\theta)S'_{\pm}(\theta) - (2r+1)q_r^{\pm}(\theta)S''_{\pm}(\theta)\right] \end{aligned}$$

and the first assertion of the Lemma is proved. To obtain relations (4.5) note that

$$S_{\pm}^{(r)}(\theta) = (-1)^{r+1} S_{\pm}^{(r)}(\pi - \theta) = (-1)^r S_{\mp}^{(r)}(-\theta), \quad r = 1, 2$$

Then (4.5) follows easily provided we prove that

$$q_r^{\pm}(\theta) = (-1)^r q_r^{\pm}(\pi - \theta),$$
$$q_r^{+}(\theta) = q_r^{-}(-\theta).$$

To prove (4.7a) we proceed by induction on (4.4). For $r = 0$, (4.7a) is clear since $q_0^{\pm} \equiv 1$. If (4.7a) holds for some r, then by (4.4) and (4.6),

$$\begin{aligned} q_{r+1}^{\pm}(\theta) &= \left(q_r^{\pm}\right)'(\theta)S'_{\pm}(\theta) - (2r+1)q_r^{\pm}(\theta)S''_{\pm}(\theta) \\ &= (-1)^{r+1}\left[\left(q_r^{\pm}\right)'(\pi-\theta)\right](-1)^2 S'_{\pm}(\pi-\theta) - (2r+1)(-1)^r\left[q_r^{\pm}(\pi-\theta)\right](-1)^3 S''_{\pm}(\pi-\theta) \\ &= (-1)^{r+3}\left(\left(q_r^{\pm}\right)'(\pi-\theta)S'_{\pm}(\pi-\theta) - (2r+1)q_r^{\pm}(\pi-\theta)S''_{\pm}(\pi-\theta)\right) \\ &= (-1)^{r+1} q_{r+1}^{\pm}(\pi-\theta). \end{aligned}$$

Similarly, (4.7b) holds for $r=0$. Assuming that (4.7b) holds for r, we observe

$$\begin{aligned}
q_{r+1}^{+}(\theta) &= \left(q_r^{+}\right)'(\theta)S_{+}'(\theta)-(2r+1)q_r^{+}(\theta)S_{+}''(\theta)\\
&= \left[-\left(q_r^{-}\right)'(-\theta)\right]\left(-S_{-}'(-\theta)\right)-(2r+1)\left[q_r^{-}(-\theta)\right](-1)^2S_{-}''(-\theta)\\
&= \left(q_r^{-}\right)'(-\theta)S_{-}'(-\theta)-(2r+1)q_r^{-}(-\theta)S_{-}''(-\theta)\\
&= q_{r+1}^{-}(-\theta),
\end{aligned}$$

and the proof is finished.

COROLLARY 4.2 For all $M\geqslant k$ and $r\geqslant 1$ there exists $C_r>0$ independent of M and k such that

$$\left|p_r^{\pm}(\theta)\right|+\left|\left(p_r^{\pm}\right)'(\theta)\right|\leqslant C_r k M^{-r-2} \text{ for all } \theta\in[0,\pi]$$

Proof. Note that since $q_1^{\pm}(\theta)=k\cos\theta$ one can check easily from (4.4) that k is a common factor in $q_r^{\pm}$for all $r\geqslant 1$. Moreover, for fixed $\theta, q_r^{\pm}(\theta)$ is a polynomial in M and k of (total) degree r and its coefficients are continuous in θ (this can be easily verified from its definition in (4.4)). Hence, there exist constants C_r' such that

$$\left|q_r^{\pm}(\theta)\right|\leqslant C_r'k\sum k^{p-1}M^q=C_r'kM^{r-1}\sum\left(\frac{k}{M}\right)^{p-1}M^{q+p-r},$$

where the sum is over all $p\geqslant 1, q\geqslant 0$ such that $p+q\leqslant r$. Hence, since $M>k$

$$\left|q_r^{\pm}(\theta)\right|\leqslant C_r'kM^{r-1}\quad \text{for all}\,\theta\in[0,\pi]$$

On the other hand,

$$\left|S_{\pm}'(\theta)\right|\geqslant 2M-k\geqslant M$$

Collecting both bounds we conclude

$$\left|p_r^{\pm}(\theta)\right|=\left|\frac{q_r^{\pm}(\theta)}{\left(S_{\pm}'(\theta)\right)^{2r+1}}\right|\leqslant\frac{C_r'kM^{r-1}}{M^{2r+1}}=C_r'kM^{-r-2},\quad \theta\in[0,\pi].$$

The estimate for $\left|\left(p_r^{\pm}\right)'(\theta)\right|$ is a consequence of (4.2 b) since

$$\left|\left(p_r^{\pm}\right)'(\theta)\right|=\left|\mathrm{i}S_{\pm}'(\theta)p_{r+1}^{\pm}(\theta)\right|\leqslant C_{r+1}'(2M+k)kM^{-r-3}\leqslant 3C_{r+1}'kM^{-r-2},\quad \theta\in[0,\pi].$$

THEOREM 4.3 For all $M\geqslant k$ we have

$$\rho_{2M}(k)=-2\left[\sum_{r=0}^{J}p_{2r}^{+}(0)\sin k+\mathrm{i}\sum_{r=0}^{J}p_{2r+1}^{+}(0)\cos k\right]+R_J(M,k)$$

with

$$|R_J(M,k)|\leqslant C_J kM^{-2J-4},$$

and C_J is independent of M and k. Proof. Integrating $I_k^{\pm}(M)$ twice by parts

$$\begin{aligned}
I_k^{\pm}(M) &= \int_0^{\pi} \frac{1}{\mathrm{i}S'_{\pm}(\theta)} \left[\mathrm{i}S'_{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\right] \mathrm{d}\theta = \int_0^{\pi} p_0^{\pm}(\theta) \left[\mathrm{i}S'_{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\right] \mathrm{d}\theta \\
&= p_0^{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\Big|_0^{\pi} - \int_0^{\pi} \frac{\mathrm{d}p_0^{\pm}(\theta)}{\mathrm{d}\theta} \exp\left(\mathrm{i}S_{\pm}(\theta)\right) \mathrm{d}\theta \\
&= p_0^{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\Big|_0^{\pi} - \int_0^{\pi} p_1^{\pm}(\theta) \left[\mathrm{i}S'_{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\right] \mathrm{d}\theta \\
&= \sum_{r=0}^{1} (-1)^r p_r^{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\Bigg|_0^{\pi} + \int_0^{\pi} \frac{\mathrm{d}p_1^{\pm}(\theta)}{\mathrm{d}\theta} \exp\left(\mathrm{i}S_{\pm}(\theta)\right) \mathrm{d}\theta
\end{aligned}$$

Repeating the same argument and using that $S_{\pm}(0) = k, S_{\pm}(\pi) = \pm 2M\pi - k$, we finally obtain

$$\begin{aligned}
I_k^{\pm}(M) &= \sum_{r=0}^{2J+2} (-1)^r p_r^{\pm}(\theta)\exp\left(\mathrm{i}S_{\pm}(\theta)\right)\Bigg|_0^{\pi} - \int_0^{\pi} \frac{\mathrm{d}p_{2J+2}^{\pm}(\theta)}{\mathrm{d}\theta} \exp\left(\mathrm{i}S_{\pm}(\theta)\right) \mathrm{d}\theta \\
&= -\sum_{r=0}^{2J+2} (-1)^r p_r^{\pm}(0) \left[\exp(\mathrm{i}k) + (-1)^{r+1}\exp(-\mathrm{i}k)\right] \\
&\quad - \int_0^{\pi} \frac{\mathrm{d}p_{2J+2}^{\pm}(\theta)}{\mathrm{d}\theta} \exp\left(\mathrm{i}S_{\pm}(\theta)\right) \mathrm{d}\theta,
\end{aligned}$$

where we have now applied (4.5a). Thus, from (4.1),

$$\begin{aligned}
\rho_{2M}(k) &= -\frac{\mathrm{i}}{2}\left[I_k^{+}(M) - I_k^{-}(M)\right] \\
&= \frac{\mathrm{i}}{2} \sum_{r=0}^{2J+2} (-1)^r \left(p_r^{+}(0) - p_r^{-}(0)\right) \left[\exp(\mathrm{i}k) + (-1)^{r+1}\exp(-\mathrm{i}k)\right] \\
&\quad + \frac{\mathrm{i}}{2}\int_0^{\pi} \left[\frac{\mathrm{d}p_{2J+2}^{+}(\theta)}{\mathrm{d}\theta}\exp\left(\mathrm{i}S_{+}(\theta)\right) - \frac{\mathrm{d}p_{2J+2}^{-}(\theta)}{\mathrm{d}\theta}\exp\left(\mathrm{i}S_{-}(\theta)\right)\right] \mathrm{d}\theta \\
&= -2\left[\sum_{r=0}^{J} p_{2r}^{+}(0)\sin k + \mathrm{i}\sum_{r=0}^{J} p_{2r+1}^{+}(0)\cos k\right] + R_J(M,k),
\end{aligned}$$

where we have applied (4.5b) in the last step. If we now define

$$R_J(M,k) := -2p_{2J+2}^{+}(0)\sin k + \frac{\mathrm{i}}{2}\int_0^{\pi} \left[\frac{\mathrm{d}p_{2J+2}^{+}(\theta)}{\mathrm{d}\theta}\exp\left(\mathrm{i}S_{+}(\theta)\right) - \frac{\mathrm{d}p_{2J+2}^{-}(\theta)}{\mathrm{d}\theta}\exp\left(\mathrm{i}S_{-}(\theta)\right)\right] \mathrm{d}\theta,$$

and finally use Corollary 4.2 we obtain

$$|R_J(M,k)| \leqslant C_J k M^{-2J-4}$$

with a suitable constant C independent of M and k. The result is now proven. Proof of Theorem 3.1. It is now a simple consequence of Theorem 4.3 since by (4.2)

$$p_0^+ = \frac{1}{\mathrm{i}} p_0, \quad p_r^+(\theta) = p_0^+(\theta)\frac{\mathrm{d}p_{r-1}^+(\theta)}{\mathrm{d}\theta} = \frac{1}{\mathrm{i}r+1} p_0(\theta)\frac{\mathrm{d}p_{r-1}(\theta)}{\mathrm{d}\theta} = \frac{1}{\mathrm{i}r+1} p_r(\theta),$$

that is,

$$p_r = \mathrm{i}^{r+1} p_r^+$$

0.5 Proof of the stability of the algorithm

Our purpose in this section is to prove the stability of algorithms in both phase 1 and phase 2.
THEOREM 5.1 Let $(\varepsilon_m)_m \subset \mathbb{C}$ with $|\varepsilon_m| \leqslant \varepsilon$ and define

$$\begin{aligned}
\widetilde{\rho}_1(k) &:= \rho_1(k) + \varepsilon_1, \\
\widetilde{\rho}_2(k) &:= \rho_2(k) + \varepsilon_2, \\
\widetilde{\rho}_{n+1}(k) &:= 2\gamma_n(k) - \frac{2n}{\mathrm{i}k}\widetilde{\rho}_n(k) + \widetilde{\rho}_{n-1}(k) + \varepsilon_{n+1}, \quad n = 2, 3, \ldots.
\end{aligned}$$

Then for all $2 < n < k$

$$|\widetilde{\rho}_n(k) - \rho_n(k)| \leqslant \left[1 + \frac{4}{3}\frac{nk^{1/2}}{(k^2-n^2)^{1/4}}\right]\varepsilon$$

Proof. Setting $\delta_n := \widetilde{\rho}_n(k) - \rho_n(k)$, we see that

$$\begin{aligned}
\delta_1 &= \varepsilon_1, \quad \delta_2 = \varepsilon_2, \\
\delta_n &= -\frac{2(n-1)}{\mathrm{i}k}\delta_{n-1} + \delta_{n-2} + \varepsilon_n, \quad n = 3, 4, \ldots,
\end{aligned}$$

$$\begin{array}{rr}
 & -\rho_n(k) = -2\gamma_{n-1}(k) + \frac{2(n-1)}{ik}\rho_{n-1}(k) - \rho_{n-2}(k) \\
+ & \widetilde{\rho}_n(k) = 2\gamma_{n-1}(k) - \frac{2(n-1)}{ik}\widetilde{\rho}_{n-1}(k) + \widetilde{\rho}_{n-2}(k) + \epsilon_n \\
\hline
 & \delta_n = \frac{2(n-1)}{ik}(\widetilde{\rho}_{n-1}(k) - \rho_{n-1}(k)) + (\widetilde{\rho}_{n-2}(k) - \rho_{n-2}(k)) + \epsilon_n
\end{array}$$

$$\delta_n = \frac{2(n-1)}{ik}\delta_{n-1} + \delta_{n-2} + \epsilon_n$$

in matrix notation

$$\begin{bmatrix} \delta_n \\ \delta_{n-1} \end{bmatrix} = \begin{bmatrix} -\frac{2(n-1)}{\mathrm{i}k} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \delta_{n-1} \\ \delta_{n-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_n \\ 0 \end{bmatrix}, \quad n \geqslant 3.$$

Introducing the notation

$$\boldsymbol{\delta}_n := \begin{bmatrix} \delta_n \\ \delta_{n-1} \end{bmatrix}, \quad \varepsilon_2 := \begin{bmatrix} \varepsilon_2 \\ \varepsilon_1 \end{bmatrix}, \quad \boldsymbol{\varepsilon}_n := \begin{bmatrix} \varepsilon_n \\ 0 \end{bmatrix} \quad \text{for } n = 3, 4, \ldots$$

and the matrix

$$D_n := \begin{bmatrix} -\frac{2(n-1)}{\mathrm{i}k} & 1 \\ 1 & 0 \end{bmatrix}$$

we see that δ_n satisfies

$$\boldsymbol{\delta}_n = D_n \boldsymbol{\delta}_{n-1} + \boldsymbol{\varepsilon}_n \text{ for } n \geqslant 3.$$

It is then easily proved by induction that

$$\delta_n = \sum_{j=2}^{n-1} \left[\prod_{i=j}^{n-1} D_{i+1} \right] \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_n \quad \text{ for } n \geqslant 2$$

Proof by induction first base step:
n=3

$$\boldsymbol{\delta}_3 = D_3 \boldsymbol{\delta}_2 + \boldsymbol{\varepsilon}_3$$

$$\begin{bmatrix} \delta_3 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} -\frac{4}{\mathrm{i}k} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \delta_2 \\ \delta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_3 \\ 0 \end{bmatrix}, \quad n \geqslant 3.$$

$$\begin{bmatrix} \frac{-4}{ik}\delta_2 + \delta_1 + \varepsilon_3 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} \delta_3 \\ \delta_2 \end{bmatrix}, \quad n \geqslant 3.$$

so this is true.
Assume n-1 holds

$$\boldsymbol{\delta}_{n-1} = \sum_{j=2}^{n-2} \left[\prod_{i=j}^{n-2} D_{i+1} \right] \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_{n-1}$$

$$D_n \boldsymbol{\delta}_{n-1} + \boldsymbol{\varepsilon}_n = D_n \left(\sum_{j=2}^{n-2} \left[\prod_{i=j}^{n-2} D_{i+1} \right] \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_{n-1} \right) + \boldsymbol{\varepsilon}_n$$

$$\left(\sum_{j=2}^{n-2} D_n \left[\prod_{i=j}^{n-2} D_{i+1} \right] \boldsymbol{\varepsilon}_{\boldsymbol{j}} + D_n \boldsymbol{\varepsilon}_{n-1} \right) + \boldsymbol{\varepsilon}_n$$

$$= \left(\sum_{j=2}^{n-2} \left[\prod_{i=j}^{n-1} D_{i+1} \right] \boldsymbol{\varepsilon}_{\boldsymbol{j}} + D_n \boldsymbol{\varepsilon}_{n-1} \right) + \boldsymbol{\varepsilon}_n$$

when j=n-2 $\quad D_{n-1} D_n \boldsymbol{\varepsilon}_{n-2} \quad$ is the last term and add $\quad D_n \boldsymbol{\varepsilon}_{n-1}$

Hence,

$$\sum_{j=2}^{n-1} \left[\prod_{i=j}^{n-1} D_{i+1} \right] \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_n = \boldsymbol{\delta}_n \quad \text{ for } n \geqslant 3$$

where the sum on the right-hand side vanishes for $n=2$ and

$$\prod_{i=j}^{n-1} D_{i+1} := D_n D_{n-1} \ldots D_j$$

For each $j=2,3,\ldots$ consider the sequence $\left\{\delta_n^{(j)}\right\}_{n=j}^{\infty}$ defined by

$$\boldsymbol{\delta}_j^{(j)} := \boldsymbol{\varepsilon}_j \quad \text{and} \quad \boldsymbol{\delta}_n^{(j)} := \left[\prod_{i=j}^{n-1} D_{i+1}\right] \boldsymbol{\varepsilon}_j, \quad n \geqslant j+1.$$

Then clearly,

$$\boldsymbol{\delta}_n = \sum_{j=2}^{n-1} \boldsymbol{\delta}_n^{(j)} + \boldsymbol{\varepsilon}_n \quad \text{(we just say with different notation in here)}$$

Now we define

$$\boldsymbol{\delta}_n^{(j)} = \left[\begin{array}{c} \delta_n^{(j)} \\ \delta_{n-1}^{(j)} \end{array}\right]$$

Then since $\boldsymbol{\delta}_n^{(j)} = D_n \boldsymbol{\delta}_{n-1}^{(j)}$

$$\left[\begin{array}{c} \delta_n^{(j)} \\ \delta_{n-1}^{(j)} \end{array}\right] = \left[\begin{array}{cc} -\frac{2(n-1)}{\mathrm{i}k} & 1 \\ 1 & 0 \end{array}\right] \left[\begin{array}{c} \delta_{n-1}^{(j)} \\ \delta_{n-2}^{(j)} \end{array}\right] = \left[\begin{array}{c} -\frac{2(n-1)}{ik} \delta_{n-1}^{(j)} + \delta_{n-2}^{(j)} \\ \delta_{n-1}^{(j)} \end{array}\right]$$

it follows that for each $j \geqslant 2$, $\left\{\delta_n^{(j)}\right\}_{n=j}^{\infty}$ satisfies the following difference equation with respect to n :

$$\delta_n^{(j)} = -\frac{2(n-1)}{\mathrm{i}k} \delta_{n-1}^{(j)} + \delta_{n-2}^{(j)}, \quad n \geqslant j+1,$$

with starting conditions

$$\left.\begin{array}{lll} \delta_1^{(2)} := \varepsilon_1, & \delta_2^{(2)} = \varepsilon_2, & \\ \delta_{j-1}^{(j)} := 0, & \delta_j^{(j)} = \varepsilon_j & \text{for } j = 3, 4, \ldots \end{array}\right\}.$$

Consider now the closely related difference equation

$$a_n - \frac{2(n-1)}{x} a_{n-1} + a_{n-2} = 0,$$

and let J_n, Y_n be Bessel functions of the first and second kind, respectively. Then since $\{J_n(x), Y_n(x)\}_n \geqslant 1$ is a fundamental system of solutions for (5.6) (cf. Abramowitz & Stegun, 1964, Chapter 9) the functions

$$\widetilde{J}_n(k) := \mathrm{i}^n J_n(k), \quad \widetilde{Y}_n(k) := \mathrm{i}^n Y_n(k)$$

are independent solutions for (5.4). Hence, for $j \geqslant 3$, the solution of (5.4) may be written

$$\delta_n^{(j)} = \alpha^{(j)}\widetilde{J}_n(k) + \beta^{(j)}\widetilde{Y}_n(k) \quad \text{for } n \geqslant j+1$$

$$\alpha^{(j)}\widetilde{J}_{j-1}(k) + \beta^{(j)}\widetilde{Y}_{j-1}(k) = \quad \delta_{j-1}^{(j)} = 0 \quad \text{we defined as above}$$

$$\alpha^{(j)}\widetilde{J}_j(k) + \beta^{(j)}\widetilde{Y}_j(k) = \delta_j^{(j)} = \varepsilon_j \quad \text{we defined as above}$$

where, $\left(\alpha^{(j)}, \beta^{(j)}\right)$ has to satisfy

$$\begin{bmatrix} \widetilde{J}_{j-1}(k) & \widetilde{Y}_{j-1}(k) \\ \widetilde{J}_j(k) & \widetilde{Y}_j(k) \end{bmatrix} \begin{bmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon_j \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{bmatrix} = \begin{bmatrix} \widetilde{J}_{j-1}(k) & \widetilde{Y}_{j-1}(k) \\ \widetilde{J}_j(k) & \widetilde{Y}_j(k) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \varepsilon_j \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \widetilde{J}_n(k) & \widetilde{Y}_n(k) \end{bmatrix} \begin{bmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{bmatrix} = \alpha^{(j)}\widetilde{J}_n(k) + \beta^{(j)}\widetilde{Y}_n(k) = \delta_n^{(j)}$$

$$= \begin{bmatrix} \widetilde{J}_n(k) & \widetilde{Y}_n(k) \end{bmatrix} \begin{bmatrix} \widetilde{J}_{j-1}(k) & \widetilde{Y}_{j-1}(k) \\ \widetilde{J}_j(k) & \widetilde{Y}_j(k) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \varepsilon_j \end{bmatrix}$$

We know that wen Matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad \text{(Notice that it is the determinant of matrice)}$$

Hence,

$$\begin{bmatrix} \widetilde{J}_{j-1}(k) & \widetilde{Y}_{j-1}(k) \\ \widetilde{J}_j(k) & \widetilde{Y}_j(k) \end{bmatrix}^{-1} = \frac{1}{\widetilde{J}_{j-1}(k)\widetilde{Y}_j(k) - \widetilde{J}_j(k)\widetilde{Y}_{j-1}(k)} \begin{bmatrix} \widetilde{Y}_j(k) & -\widetilde{Y}_{j-1}(k) \\ -\widetilde{J}_j(k) & \widetilde{J}_{j-1}(k) \end{bmatrix}$$

$$\widetilde{J}_{j-1}(k) = \mathrm{i}^{j-1}J_{j-1}(k), \quad \widetilde{Y}_j(k) = \mathrm{i}^j Y_j(k)$$

$$\widetilde{J}_j(k) = \mathrm{i}^j J_j(k), \quad \widetilde{Y}_{j-1}(k) = \mathrm{i}^{j-1} Y_j(k)$$

$$\widetilde{J}_{j-1}(k)\widetilde{Y}_j(k) - \widetilde{J}_j(k)\widetilde{Y}_{j-1}(k) = \mathrm{i}^{j-1}J_{j-1}(k).\mathrm{i}^j Y_j(k) - \mathrm{i}^j J_j(k).\mathrm{i}^{j-1}Y_j(k)$$
$$= \mathrm{i}^{2j-1}(J_{j-1}(k).Y_j(k) - J_j(k).Y_j(k))$$

$$(\mathrm{i}^2)^{j-1}.\mathrm{i}(J_{j-1}(k).Y_j(k) - J_j(k).Y_j(k)) = (-1)^{j-1}\mathrm{i}(J_{j-1}(k).Y_j(k) - J_j(k).Y_j(k))$$

where we have used the identity (see Abramowitz & Stegun, 1964, equation (9.1.16))

$$J_j(k)Y_{j-1}(k) - J_{j-1}(k)Y_j(k) = \frac{2}{\pi k}$$

Finally

$$\frac{(-1)^{j+1}\pi k\mathrm{i}}{2} = \frac{1}{\widetilde{J}_{j-1}(k)\widetilde{Y}_j(k) - \widetilde{J}_j(k)\widetilde{Y}_{j-1}(k)}$$

$$\text{Defining} \quad M_n(k) := \sqrt{J_n(k)^2 + Y_n(k)^2},$$

Notice also:

$$\begin{bmatrix} \widetilde{Y}_j(k) & -\widetilde{Y}_{j-1}(k) \\ -\widetilde{J}_j(k) & \widetilde{J}_{j-1}(k) \end{bmatrix} \begin{bmatrix} 0 \\ \varepsilon_j \end{bmatrix} = \begin{bmatrix} -\widetilde{Y}_{j-1}(k) \\ \widetilde{J}_{j-1}(k) \end{bmatrix} \varepsilon_j$$

and using the Cauchy-Schwarz inequality :

$$(a_1b_1 + a_2b_2)^2 \leq (a_1^2 + a_2^2)(b_1^2 + b_2^2)$$

replace respectful variables:

$$a_1 = \widetilde{J}_n(k) \quad a_2 = \widetilde{Y}_n(k) \qquad b_1 = -\widetilde{Y}_{j-1}(k) \quad b_2 = \widetilde{J}_{j-1}(k)$$

$$\widetilde{J}_n(k)(-\widetilde{Y}_{j-1}(k)) + \widetilde{Y}_n(k)\widetilde{J}_{j-1}(k) \leq \sqrt{(\widetilde{J}_n(k)^2 + \widetilde{Y}_n(k)^2)((-\widetilde{Y}_{j-1}(k))^2 + \widetilde{J}_{j-1}(k)^2)}$$

$$= \sqrt{(\widetilde{J}_n(k)^2 + \widetilde{Y}_n(k)^2)}\sqrt{(\widetilde{Y}_{j-1}(k)^2 + \widetilde{J}_{j-1}(k)^2)} = M_n(k)M_{j-1}(k)$$

$$\text{and} \quad |\varepsilon_j| \leqslant \varepsilon$$

When we all sum up that we have

$$\left|\delta_n^{(j)}\right| = \left|\frac{(-1)^{j+1}\pi k\mathrm{i}}{2}\left[\widetilde{J}_n(k)\widetilde{Y}_n(k)\right]\begin{bmatrix} -\widetilde{Y}_{j-1}(k) \\ J_{j-1}(k) \end{bmatrix}\varepsilon_j\right| \leqslant \frac{\pi k}{2} M_n(k)M_{j-1}(k)\varepsilon \quad \text{for } j \geq 3.$$

Now observe that (see Watson, 1995, Section 13.74)

$$M_n^2(k) \leqslant \frac{2}{\pi}\frac{1}{\sqrt{k^2 - n^2}} \qquad \text{for } k > n > 1/2.$$

$$M_{j-1}^2(k) \leqslant \frac{2}{\pi}\frac{1}{\sqrt{k^2 - (j-1)^2}} \qquad \text{for } k > n \geq j > 1/2$$

$$M_n(k)M_{j-1}(k) \leq \sqrt{\frac{2}{\pi}}\sqrt{\frac{2}{\pi}}\left[\frac{1}{(k^2 - n^2)^{1/4}}\frac{1}{(k^2 - (j-1)^2)^{1/4}}\right]$$

$$\left|\delta_n^{(j)}\right| \leqslant \frac{\pi k}{2}\frac{2}{\pi}\left[\frac{1}{\left(k^2-n^2\right)^{1/4}}\frac{1}{\left(k^2-(j-1)^2\right)^{1/4}}\right]\varepsilon$$

$$\Rightarrow \left|\delta_n^{(j)}\right| \leqslant \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\frac{1}{\left(k^2-(j-1)^2\right)^{1/4}}\right] \quad j=3,4,\ldots,n,$$

$$\left|\delta_n^{(2)}\right| \leqslant \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\frac{1}{\left(k^2-1\right)^{1/4}}+\frac{1}{\left(k^2-4\right)^{1/4}}\right]$$

We know that :

$$\boldsymbol{\delta}_n = \sum_{j=2}^{n-1}\boldsymbol{\delta}_n^{(j)} + \boldsymbol{\varepsilon}_n = \sum_{j=3}^{n-1}\boldsymbol{\delta}_n^{(j)} + \boldsymbol{\delta}_n^2 + \boldsymbol{\varepsilon}_n$$

$$|\boldsymbol{\delta}_n| = \left|\sum_{j=3}^{n-1}\boldsymbol{\delta}_n^{(j)} + \boldsymbol{\delta}_n^2 + \boldsymbol{\varepsilon}_n\right|$$

$$\leq \left|\sum_{j=3}^{n-1}\left(\frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\frac{1}{\left(k^2-(j-1)^2\right)^{1/4}}\right]\right) + \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left(\frac{1}{\left(k^2-1\right)^{1/4}}+\frac{1}{\left(k^2-4\right)^{1/4}}\right) + \boldsymbol{\varepsilon}_n\right|$$

$$= \left|\frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\sum_{j=3}^{n-1}\left(\frac{1}{\left(k^2-(j-1)^2\right)^{1/4}}\right) + \left(\frac{1}{\left(k^2-1\right)^{1/4}}+\frac{1}{\left(k^2-4\right)^{1/4}}\right)\right] + \boldsymbol{\varepsilon}_n\right|$$

Gathering all together and using the first entry of vector identity (5.3), we have

$$\begin{aligned}
|\delta_n| &\leqslant \varepsilon + \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\frac{1}{\left(k^2-1\right)^{1/4}}+\frac{1}{\left(k^2-4\right)^{1/4}}+\sum_{j=2}^{n-2}\frac{1}{\left(k^2-j^2\right)^{1/4}}\right] \\
&\leqslant \varepsilon + \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\frac{1}{\left(k^2-1\right)^{1/4}}+\frac{1}{\left(k^2-(n-1)^2\right)^{1/4}}+\sum_{j=2}^{n-2}\frac{1}{\left(k^2-j^2\right)^{1/4}}\right] \\
&= \varepsilon + \frac{k\varepsilon}{\left(k^2-n^2\right)^{1/4}}\left[\sum_{j=1}^{n-1}\frac{1}{\left(k^2-j^2\right)^{1/4}}\right]
\end{aligned}$$

(The derivation of (5.7) uses the assumption that $n \geqslant 3$.) Now, since $\left(k^2-x^2\right)^{-1/4}$ is an increasing function for $x \in [0,k)$, the Riemann sum in

(5.8) can be bounded as

$$
\begin{aligned}
\sum_{j=1}^{n-1} \frac{1}{\left(k^2-j^2\right)^{1/4}} &\leqslant \int_0^n \frac{\mathrm{d}x}{\left(k^2-x^2\right)^{1/4}} < \frac{1}{k^{1/4}} \int_0^n \frac{\mathrm{d}x}{(k-x)^{1/4}} \\
&= \frac{1}{k^{1/4}}\left[-\frac{4}{3}(k-x)^{3/4}\right]_0^n = \frac{4}{3k^{1/4}}\left[k^{3/4}-(k-n)^{3/4}\right] \\
&= \frac{4k^{1/2}}{3}\left[1-\left(\frac{k-n}{k}\right)^{3/4}\right] < \frac{4k^{1/2}}{3}\left[1-\frac{k-n}{k}\right] \\
&= \frac{4}{3}k^{-1/2}n.
\end{aligned}
$$

Using this bound in (5.8) we finish the proof. Note that for $k >> n$ we derive from this result that $|\widetilde{\rho}_n(k)-\rho_n(k)| \lesssim n\varepsilon$, i.e., the computation of these coefficients is stable. The worst case occurs when n is taken close (or equal) to k. The following result gives a bound on the error in this case. COROLLARY 5.2 Under the same asssumptions as Theorem 5.1 , for all $2 < n \leqslant k$, we have

$$|\widetilde{\rho}_n(k)-\rho_n(k)| \leqslant \left[4+2^{7/4}k^{5/4}\right]\varepsilon.$$

Proof. For $n \leqslant k-1$ Theorem 5.1 implies

$$|\delta_n| \leqslant \left[1+\frac{4}{3}\frac{k^{1/2}(k-1)}{\left(k^2-(k-1)^2\right)^{1/4}}\right]\varepsilon = \left[1+\frac{2^{7/4}}{3}\frac{k^{1/2}(k-1)}{(k-1/2)^{1/4}}\right]\varepsilon \leqslant \left[1+\frac{2^{7/4}k^{5/4}}{3}\right]\varepsilon$$

For $k-1 < n \leqslant k$ we simply note that

$$|\delta_n| \leqslant \varepsilon + \frac{2n}{k}\left|\delta_{n-1}\right| + |\delta_{n-2}| \leqslant \varepsilon + 2\left|\delta_{n-1}\right| + |\delta_{n-2}|$$

from where the result follows. The proof of the Corollary suggests that the recurrence may become unstable when $n > k$. This has been observed by other authors and we illustrate this phenomenon numerically in the final section. Hence, the second phase is introduced to avoid this instability.
PROPOSiTION 5.3 Let $M \geqslant n_0 = \lceil k \rceil$ and $A_M(k)$ and $\mathbf{b}_M(k)$ be defined as in (3.9). If

$$A_M(k)\rho_M(k) = \mathbf{b}_M(k), \quad A_M(k)\widetilde{\boldsymbol{\rho}}_M(k) = \widetilde{\mathbf{b}}_M(k)$$

with

$$\left\|\widetilde{\mathbf{b}}_M(k)-\mathbf{b}_M(k)\right\|_\infty \leqslant \varepsilon,$$

then

$$\|\boldsymbol{\rho}_M(k)-\widetilde{\boldsymbol{\rho}}_M(k)\|_\infty \leqslant \left(\frac{n_0+2}{2}\right)\varepsilon.$$

Proof. Note that

$$(\boldsymbol{\rho}_M(k)-\widetilde{\boldsymbol{\rho}}_M(k)) = A_M^{-1}(k)\left(\widetilde{\mathbf{b}}_M(k)-\mathbf{b}_M(k)\right).$$

Therefore,

$$\|\widetilde{\boldsymbol{\rho}}_M(k) - \boldsymbol{\rho}_M(k)\|_\infty \leqslant \left\|A_M^{-1}(k)\right\|_\infty \left\|\widetilde{\mathbf{b}}_M(k) - \mathbf{b}_M(k)\right\|_\infty$$

and the proof reduces to bounding $\left\|A_M^{-1}(k)\right\|_\infty$. Let then

$$A_M(k) := (I_M + K_M(k))\, D_M(k),$$

where I_M is the identity matrix of order $2M - n_0$ and

$$K_M(k) := \begin{bmatrix} 0 & \frac{\mathrm{i}k}{2(n_0+1)} & & & & \\ -\frac{\mathrm{i}k}{2n0} & 0 & \frac{\mathrm{i}k}{2(n_0+2)} & & & \\ & -\frac{\mathrm{i}k}{2(n_0+1)} & 0 & \frac{\mathrm{i}k}{2(n_0+3)} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & & & & -\frac{\mathrm{i}k}{4(M-1)} & 0 \end{bmatrix}.$$

Note that

$$\|K_M(k)\|_\infty = \frac{(n_0+1)\,k}{n_0\,(n_0+2)} \leqslant \frac{n_0+1}{n_0+2} < 1$$

Therefore,

$$\left\|A_M^{-1}(k)\right\|_\infty \leqslant \left\|D_M^{-1}(k)\right\|_\infty \|{(I_M} + K_M(k))^{-1}\|_\infty \leqslant \frac{k}{2n_0} \frac{1}{1 - \|K_M(k)\|_\infty} \leqslant \frac{n_0+2}{2},$$

and the result is proven. Collecting Corollary 5.2 and Proposition 5.3 we deduce that for the second phase algorithm, we can expect in the worst case that $|\widetilde{\rho}_n(k) - \rho_n(k)| \lesssim k^{9/4}\varepsilon$ for all $n > k$. Since the second phase algorithm is only used for moderate values of k (the greater is k, the greater has to be N to make the second phase of the algorithm necessary) this bound implies the stability of the algorithm for practical computations.
REMARK 5.4 We finish this section by explaining why the implementation of the quadrature rule (1.3) has complexity $\mathcal{O}(N \log N)$. The first step in the implementation is the computation of the coefficients $\{\omega_n(k) : n = 0, \ldots, N\}$. Since the first phase of the Algorithm involves a three-term recurrence relation this requires $\mathcal{O}(\min\{N, k\}) = \mathcal{O}(N)$ operations. If the second phase is required, then we have to solve an additional tridiagonal system of size $2M - \lceil k \rceil$, where M is proportional to N, resulting in an additional $\mathcal{O}(N)$ operations, via the Thomas algorithm for tridiagonal systems. Finally, $I_{k,N}(f)$ in (1.3) may then be computed by applying the discrete cosine transform which, by FFT, requires $\mathcal{O}(N \log N)$ operations.

0.6 Numerical Experiments

Numerical experiments In this section we present some numerical experiments to illustrate the theoretical results presented in this paper.

Experiment 1

In this experiment, we illustrate the stability of the weights $\omega_n(k)$ evaluation method defined in (2.4). These exact values were first computed using analytic formulae and evaluation in high-precision arithmetic.
First, it is used the first algorithm and includes when $n > k$ and observe the instability of the algorithm.
Results:

	$k = 10$	$k = 20$	$k = 40$	$k = 80$
$n = k/2$	0.3358	0.0716	0.0855	0.1518
$n = k$	0.0860	0.6124	0.8056	0.5251
$n = 3k/2$	25.1693	494.6511	$1.9096e + 05$	$8.1384e + 10$
$n = 2k$	$5.4605e + 03$	$5.3239e + 07$	$2.0893e + 15$	$9.1435e + 30$

Matlab code for First Phase of the Algorithm:

```
% Clear all variables and close all figures
clear all
close all

% Define the function f_beta(s)
beta = 1/4;  % Choose the desired value of beta
f_beta = @(s) ((1 + s).^beta) ./ (1 + s.^2);

% Define the function T_n(s)
T_n = @(n, s) cos(n * acos(s));

% Define the number of terms in the summation
N = 24;

% Compute the integral limits
a = -1;
b = 1;

% Integration tolerances
relTol = 1e-8;
absTol = 1e-13;

% Define the values of k
i_values = 0:9;
k_values = 100 * 2 .^ i_values;

% Function to compute the estimates for different
   values of k
estimate_func = @(k) abs(quadgk(@(s) f_beta(s) .* exp
```

```
    (1i * k * s), a, b, 'RelTol', relTol, 'AbsTol',
    absTol, 'MaxIntervalCount', 10000));

% Compute the estimates for different values of k
estimates = arrayfun(estimate_func, k_values);

% Calculate e_k(beta) for different values of k
e_values = log2(estimates(1:end-1) ./ estimates(2:end)
    );

% Display the estimates for different values of k
fprintf('Estimates for different values of k:\n');
for i = 1:numel(k_values)
    fprintf('k = %d: %e\n', k_values(i), estimates(i))
        ;
end

% Display e_k(beta) for different values of k
fprintf('\ne_k(beta) for different values of k:\n');
for i = 2:numel(k_values)
    fprintf('k = %d: %f\n', k_values(i), e_values(i-1)
        );
end
```

Secondarily, we combine with the first algorithm with our second phase algorithm so instability is no more problem for values $n > k$
Here is the results:

	$k = 10$	$k = 20$	$k = 40$	$k = 80$
$n = 2k$	0.093701	0.076427	0.030889	0.020712
$n = 4k$	0.098874	0.08228	0.033482	0.022364

Matlab code for First and second phase of algorithm combined:

```
% Given k and n values
k = 20;
n = 4*k;

% Define the function T_n(s)
T_n = @(n, s) cos(n * acos(s));

% Calculate \widetilde{\omega}_n(k) for a given value
    of k and n
integrand = @(s) T_n(n, s) .* exp(1i * k * s);
omegaintegral = integral(integrand, -1, 1);

% Calculate gamma_n(k) for even and odd n
if mod(n, 2) == 0
```

```
    gamma_n = (2 * sin(k)) / k;
else
    gamma_n = (2 * cos(k)) / (1i * k);
end

% Calculate \rho_1(k)
rho(1) = gamma_n;

% Calculate \rho_2(k)
rho(2) = 2 * gamma_n - (2 / (1i * k)) * gamma_n;

% Calculate \rho_n(k) using the recursive formula
for j = 3:n
    rho(j) = 2 * gamma_n - ((2 * (j-1)) / (1i * k)) *
        rho(j-1) + rho(j-2);
end

% Calculate \omega_0(k) and \omega_n(k)
omega(1) = rho(1);
omega(n) = gamma_n - (n / (1i * k)) * rho(n);

if n < k
    % Calculate the error
    error = abs(omegaintegral - omega(n));
else
    % Compute the values of A_M(k) and b_M(k)
    n0 = ceil(k);
    M = 2 * n;
    A = zeros(M, M);
    b = zeros(M, 1);

    for i = 1:M
        A(i, i) = (2 * (n0 + i - 1)) / (1i * k);

        if i < M
            A(i, i+1) = 1;
            A(i+1, i) = -1;
        end

        if mod(n0 + i - 1, 2) == 0
            b(i) = 2 * sin(k) / (k + n0 + i - 1);
        else
            b(i) = 2 * cos(k) / (1i * k + n0 + i - 1);
        end
    end
```

```
    % Solve the tridiagonal system using Thomas'
        algorithm
    rho = ThomasAlgorithm(A, b);

    % Compute the values of omega_n(k)
    omega = zeros(n+1, 1);
    for i = n0:n
        omega(i) = gamma_n - (i / (1i * k)) * rho(i -
            n0 + 1);
    end

    % Calculate the error
    error = omegaintegral - omega(n);
end
absolute = abs(error);
disp(['Error for k=', num2str(k), ' and n=', num2str(n
    ), ': ', num2str(absolute)]);

function x = ThomasAlgorithm(A, b)
    n = size(A, 1);

    c = zeros(n, 1);
    d = zeros(n, 1);

    % Forward Sweep
    c(1) = A(1, 2) / A(1, 1);
    d(1) = b(1) / A(1, 1);
    for i = 2:n-1
        c(i) = A(i, i+1) / (A(i, i) - A(i, i-1) * c(i
            -1));
        d(i) = (b(i) - A(i, i-1) * d(i-1)) / (A(i, i)
            - A(i, i-1) * c(i-1));
    end

    % Backward Sweep
    x = zeros(n, 1);
    x(n) = (b(n) - A(n, n-1) * d(n-1)) / (A(n, n) - A(
        n, n-1) * c(n-1));
    for i = n-1:-1:1
        x(i) = d(i) - c(i) * x(i+1);
    end
end
```

Experiment 2

In this experiment we study the rate of decay of the error in the

Filon-Clenshaw-Curtis rule for fixed N, as $k \rightarrow \infty$, To do this, for $\beta>0$, define

$$
f_\beta(s):=\frac{(1+s)^\beta}{1+s^2}, \quad s \in[-1,1]
$$

We compute the error in (2.7) with $N=24$ (a 25 -point rule):

$$
E_k\left(f_\beta\right):=\left|\int_{-1}^1 f_\beta(s) \exp (\mathrm{i} k s) \mathrm{d} s-I_{k, 24}\left(f_\beta\right)\right| .
$$

Results are:
Estimates for different values of k:
k = 100: 5.982613e-03
k = 200: 3.482716e-03
k = 400: 1.524485e-03
k = 800: 8.367027e-04
k = 1600: 3.695753e-04
k = 3200: 2.043599e-04
k = 6400: 8.732997e-05
k = 12800: 4.766506e-05
k = 25600: 2.392810e-05
k = 51200: 1.210188e-05
Matlab code for Error estimate:

```
% Clear all variables and close all figures
clear all
close all

% Define the function f_beta(s)
beta = 1/4;  % Choose the desired value of beta
f_beta = @(s) ((1 + s).^beta) ./ (1 + s.^2);

% Define the function T_n(s)
T_n = @(n, s) cos(n * acos(s));

% Define the number of terms in the summation
N = 24;

% Compute the integral limits
a = -1;
b = 1;

% Integration tolerances
relTol = 1e-8;
absTol = 1e-13;
```

```
% Define the values of k
i_values = 0:9;
k_values = 100 * 2 .^ i_values;

% Function to compute the estimates for different
   values of k
estimate_func = @(k) abs(quadgk(@(s) f_beta(s) .* exp
   (1i * k * s), a, b, 'RelTol', relTol, 'AbsTol',
   absTol, 'MaxIntervalCount', 10000));

% Compute the estimates for different values of k
estimates = arrayfun(estimate_func, k_values);

% Calculate e_k(beta) for different values of k
e_values = log2(estimates(1:end-1) ./ estimates(2:end)
   );

% Display the estimates for different values of k
fprintf('Estimates for different values of k:\n');
for i = 1:numel(k_values)
    fprintf('k = %d: %e\n', k_values(i), estimates(i))
        ;
end

% Display e_k(beta) for different values of k
fprintf('\ne_k(beta) for different values of k:\n');
for i = 2:numel(k_values)
    fprintf('k = %d: %f\n', k_values(i), e_values(i-1)
        );
end
```

0.7 Fast Fourier Transform

Fast Fourier Transform is an algorithm that compute coeffients of discrete fourie transform $\mathcal{O}\left(N \log_2 N\right)$ time instead of $\mathcal{O}\left(N^2\right)$ by direct computation.
We will explain simple version of FFT, known as Cooley-Tukey FFT.
Assume that we want to compute discrete fourier coefficients of f for N-point sequence

$$n = 0, 1, 2, ... \frac{N}{2}$$

$$
\begin{aligned}
F_n &= \sum_{j=0}^{N-1} e^{-2\pi i jn/N} f_j \\
&= \sum_{j=0}^{\frac{N}{2}-1} e^{-2\pi i(2j)n/N} f_{2j} + \sum_{j=0}^{\frac{N}{2}-1} e^{-2\pi i(2j+1)n/N} f_{2j+1} \\
&= \sum_{j=0}^{\frac{N}{2}-1} e^{-2\pi i jn/\frac{N}{2}} f_{2j} + e^{-2\pi i n/N} \sum_{j=0}^{\frac{N}{2}-1} e^{-2\pi i jn/\frac{N}{2}} f_{2j+1} \\
&= F_n^e + w^n F_n^o
\end{aligned}
$$

where F^e denotes the DFT of the even components f_{2j}, F^o is the DFT of the odd components f_{2j+1}, and $w = e^{-2\pi i/N}$.

Our observation shows us to compute F_n and $F_{n+\frac{N}{2}}$ at the same time:

$$
\begin{aligned}
F_{n+N/2} &= \sum_{j=0}^{N-1} e^{-2\pi i j(n+N/2)/N} f_j \\
&= \sum_{j=0}^{N-1} e^{-2\pi i jn/N} e^{-2\pi i jN/2N} f_j \\
&= \sum_{j=0}^{N-1} e^{-2\pi i jn/N} e^{-\pi i j} f_j \\
&= \sum_{j=0}^{N-1} e^{-2\pi i jn/N} (-1)^j f_j \\
&= \sum_{j=0}^{N/2-1} e^{-2\pi i(2j)n/N} f_{2j} - \sum_{j=0}^{N/2-1} e^{-2\pi i(2j+1)n/N} f_{2j+1} \\
&= \sum_{j=0}^{N/2-1} e^{-2\pi i jn/\frac{N}{2}} f_{2j} - e^{-2\pi i n/N} \sum_{j=0}^{N/2-1} e^{-2\pi i jn/\frac{N}{2}} f_{2j+1} \\
&= F_n^e - w^n F_n^o
\end{aligned}
$$

So we have :

$$F_n = F_n^e + w^n F_n^o$$

$$F_{n+N/2} = F_n^e + w^n F_n^o$$

Now that the problem has been reduced to computing F_n^e and F_n^o, we can repeat the same argument to reduce the problem to computing $F_n^{ee}, F_n^{eo}, F_n^{oe}$, and F_n^{oo}. Hence, to compute two coeffients normally we have to do 2N operation but with divide and conquer princible we can do N/2 operation for even and odd components so total cost of N operation basically for two components.

Proof of computation cost of FFT

Our proof based on observation above, we apply dividing half for all coeffients. Let's denote the number of operations required to compute the DFT of a sequence of length N as $T(N)$. According to the Cooley-Tukey algorithm, we can express $T(N)$ in terms of the number of operations required to compute the DFTs of the two halves:

$$T(N) = 2T(N/2) + \mathcal{O}(N)$$

Here, the term $2T(N/2)$ represents the computation cost of computing the DFTs of the two halves, and $\mathcal{O}(N)$ accounts for the additional operations required to combine these partial results.

By recursively applying the above equation, we can expand $T(N)$ as follows:

$$T(N) = 2T(N/2)+\mathcal{O}(N) = 2(2T(N/4)+\mathcal{O}(N/2))+\mathcal{O}(N) = 4T(N/4)+2\mathcal{O}(N/2)+\mathcal{O}(N) = \dots$$

Continuing this process, we get:

$$T(N) = 2^kT(N/2^k) + 2^{k-1}\mathcal{O}(N/2^{k-1}) + \dots + 4\mathcal{O}(N/4) + 2\mathcal{O}(N/2) + \mathcal{O}(N)$$

Here, k represents the recursion depth. When the base case of $N = 1$ is reached, we have $N/2^k = 1$, which implies $k = \log_2 N$.

Using this information, we can rewrite the expanded equation for $T(N)$ as:

$$T(N) = 2^kT(N/2^k) + 2^{k-1}\mathcal{O}(N/2^{k-1}) + \dots + 4\mathcal{O}(N/4) + 2\mathcal{O}(N/2) + \mathcal{O}(N)$$

$$= N/2^kT(1) + \mathcal{O}(N) + \dots + \mathcal{O}(N) + \mathcal{O}(N)$$

$$= N/2^kT(1) + k\mathcal{O}(N)$$

we know that: $\quad a\mathcal{O}(N) = \mathcal{O}(aN)$

Simplifying further, we have:

$$T(N) = N/2^{\log_2 N}T(1) + \mathcal{O}(N\log_2 N) = NT(1) + \mathcal{O}(N\log_2 N)$$

Since $T(1)$ represents the computation cost of computing the DFT of a single point, which is a constant, we can write $T(1)$ as C. Thus, the equation simplifies to:

$$T(N) = C \cdot N + \mathcal{O}(N\log_2 N)$$

Therefore, we can conclude that the computation cost of the FFT algorithm is $\mathcal{O}(N\log_2 N)$ since the term $\mathcal{O}(N\log_2 N)$ dominates the expression.