

MODIFIED FILON-CLENSHAW-CURTIS RULES FOR OSCILLATORY INTEGRALS WITH A NONLINEAR OSCILLATOR*

HASSAN MAJIDIAN[†]

Abstract. Filon-Clenshaw-Curtis (FCC) rules rank among the rapid and accurate quadrature rules for computing oscillatory integrals. In the implementation of the FCC rules, when the oscillator of the integral is nonlinear, its inverse has to be evaluated at several points. In this paper we suggest an approach based on interpolation, which leads to a class of modifications of the original FCC rules in such a way that the modified rules do not involve the inverse of the oscillator function. In the absence of stationary points, two reliable and efficient algorithms based on the modified FCC (MFCC) rules are introduced. For each algorithm, an error estimate is verified theoretically and then illustrated by some numerical experiments. Also, some numerical experiments are carried out in order to compare the convergence speed of the two algorithms. In the presence of stationary points, an algorithm based on composite MFCC rules on graded meshes is developed. An error estimate is derived and illustrated by some numerical experiments.

Key words. Filon-Clenshaw-Curtis rule, oscillatory integral, nonlinear oscillator, stationary point, graded mesh

AMS subject classifications. MSC 65D30, MSC 65T40

1. Introduction.

Consider integrals of the form

$$(1.1) \quad I_k^{[a,b]}(f, g) := \int_a^b f(x) \exp(ikg(x)) \, dx,$$

where $f \in L^1[a, b]$, $k > 0$, and $g \in C^m[a, b]$ for some positive integer m . If g does not oscillate rapidly in $[a, b]$, then the integrand in (1.1) oscillates violently for larger values of k . This class of integrals contains a large portion of highly oscillatory integrals appearing in many areas of science and engineering, e.g., in Fourier series and transforms, special functions, high-frequency acoustic scattering, etc.; see, e.g., [5, 19] and the references therein. Because of their wide applications, computing oscillatory integrals of the form (1.1) has been the subject of much research in the last two decades.

Filon-type methods are among the most efficient ones for computing (1.1), and they have a long history. While highly accurate methods based on steepest descent (see, e.g., [7, 12, 13]) need some manual calculations regarding the steepest decent paths in the complex plane, the Filon-type methods do not require complex calculus and can be performed automatically by computers provided that the moments can be computed to desired accuracy. The idea, in general, is to replace the amplitude function f by an interpolation polynomial p_n and consider $I_k^{[a,b]}(p_n, g)$ as an approximation to $I_k^{[a,b]}(f, g)$. This idea originated from [10], where Louis Napoleon George Filon (1875–1937) applied it to the Fourier integral $\int_a^b f(x) \sin kx \, dx$.

After the work of Filon, many papers on his method appeared, but the 2005 paper [14] proved to be a milestone in the history of research on Filon-type methods. In that work, the asymptotic expansion of $I_k^{[a,b]}(f, g)$ by negative powers of k was studied, and a generalization of Filon's method was developed. The paper has been the motivation for a lot of successive research.

A successful implementation of a Filon-type method rests on the ability to compute the moments $I_k^{[a,b]}(x^m, g)$. For the linear oscillator $g(x) = x$, the moments can be calculated by

*Received September 1, 2019. Accepted February 10, 2021. Published online on April 7, 2021. Recommended by Miroslav Pranić.

[†]Group of Multidisciplinary Studies, Faculty of Encyclopedia Studies, Institute for Humanities and Cultural Studies, PO Box 14155-6419, Tehran, Iran (H.Majidian@ihcs.ac.ir).

the following identity:

$$(1.2) \quad I_k^{[a,b]}(x^m, x) = \frac{1}{(-ik)^{m+1}} [\Gamma(1+m, -ika) - \Gamma(1+m, -ikb)],$$

where Γ is the incomplete Gamma function [1]. However, formula (1.2) is useless mainly because interpolation by the polynomial basis $1, x, x^2, \dots$ is in general unstable. For a more complex oscillator g , Olver [18] considered a g -dependent basis instead of the usual polynomial basis x^j . Then the moments can be written in closed forms, again in terms of incomplete Gamma functions. Another approach, proposed in [29], is based on the transformation $\tau = g(x)$. The method enables one to compute the moments by the identity (1.2) when g has no stationary points in $[a, b]$; otherwise if g has a single stationary point, similar identities for $I_k^{[0,y]}(x^m, x^j)$ are employed (cf. [1, 14]).

Filon-Clelshaw-Curtis (FCC) rules, beside other advantages, enable one to compute the moments stably and rapidly without a need to deal with (incomplete) Gamma functions, whose evaluation is not trivial. The $(N + 1)$ -point FCC rule can be described as follows: Consider the oscillatory integral

$$(1.3) \quad I_k(f) := \int_{-1}^1 f(x) \exp(ikx) dx.$$

In the $(N + 1)$ -point FCC rule, the amplitude function f is interpolated at Clenshaw-Curtis points $t_{j,N} := \cos(j\pi/N)$, $j = 0, \dots, N$, by the polynomial

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

where $T_n(s) = \cos(n \arccos(s))$ are the Chebyshev polynomials of the first kind and $\sum {}''$ means that the first and the last terms of the sum are to be halved. By the discrete orthogonality of the cosine functions, the coefficients $\alpha_{n,N}(f)$ can be written as

$$(1.4) \quad \alpha_{n,N}(f) = \frac{2}{N} \sum_{j=0}^N {}'' \cos(jn\pi/N) f(t_{j,N}), \quad n = 0, \dots, N.$$

If f in (1.3) is replaced by $Q_N f$, then the $(N + 1)$ -point FCC rule is obtained as

$$(1.5) \quad I_{k,N}(f) = \sum_{n=0}^N {}'' \alpha_{n,N}(f) \omega_n(k),$$

where the weights (modified moments) $\omega_n(k) = I_k(T_n)$, $n \geq 0$, can be computed recursively thanks to the three-term recurrence relation for the Chebyshev polynomials T_n . As a generalization of the original Clenshaw-Curtis rules [6], this idea has been developed gradually by employing results of many earlier researchers, mainly Piessens's contributions [20, 21, 22, 23, 24] and Sloan's works [27, 28]. By a simple affine-like change of variables, the FCC rules (1.5) can also be applied to

$$(1.6) \quad I_k^{[a,b]}(f) := \int_a^b f(x) \exp(ikx) dx.$$

The recent papers of Domínguez et al. [8, 9] contain important developments and results for the FCC rules. In [9], beside an error bound in terms of both k and N , a stable algorithm

with complexity of $\mathcal{O}(N \log N)$ was proposed for computing the weights. The algorithm can be employed for computing the general integral (1.1) in the following way (cf. [8]): If g has no stationary points in $[a, b]$, then we can assume that $g'(x) > 0$, $x \in [a, b]$, without loss of generality. In this case, the change of variables $\tau = g(x)$ reduces the integral to $I_k^{[g(a), g(b)]}(F)$ with $F(\tau) = ((f/g') \circ g^{-1})(\tau)$, $\tau \in [g(a), g(b)]$; otherwise, if g has some stationary points in $[a, b]$, a similar change of variables reduce the integral to a finite number of integrals of the form (1.6) such that each integral has a single singularity at one of the endpoints and can be computed efficiently by composite FCC rules on graded meshes [8].

Statement of the problem (or the motivation for this paper). In the implementation of the algorithm proposed in [8], it is required to compute g^{-1} at several points. Expressing $g^{-1}(x)$ in closed form, however, is not always possible. For example, consider the function $g(x) = x - \sin(x)$. Thus, one has to resort to approximation methods, e.g., Newton iterations. This, however, imposes extra errors and computational costs.

In this paper, we follow the idea of [29] and propose reliable and efficient methods for computing (1.1), which are based on the FCC rules but without the need to compute $g^{-1}(x)$ at any point. The modified FCC rules, as developed in this paper, play key roles when exponential transformations are applied to the oscillatory integrals in order to treat singularities (see [17] for details).

The authors of [15] have also used the ideas in [29] and suggest composite Filon-type rules which do not require the evaluation of g^{-1} , but they do not benefit from the advantages of the FCC rule. Our idea in this paper is, however, essentially different in that it is based on the FCC rules and exploits their fast and stable construction (cf. [9]) as well as their error estimates in terms of k , N , and the Sobolev regularity of f (cf. [8]). These advantages are gained by employing special interpolations such that the g -dependent interpolation nodes of [29] approximately coincide with the Clenshaw-Curtis points.

The plan of this paper is as follows: In Section 2, which contains the main ideas, we describe a class of quadrature rules for computing (1.1) in the absence of stationary points (i.e., $g(x) \neq 0$, for any $x \in [a, b]$). The rules can be considered modifications of the FCC rules [8] which circumvent the necessity of computing $g^{-1}(x)$ by imposing an interpolation process. Thus, an interpolation error is added to the total error. In order to decrease the interpolation error, two efficient interpolation methods are employed, based on which two algorithms are proposed. The next two sections deal with these algorithms, their error estimates, and some numerical experiments. In Section 4, we compare the convergence speed of the two algorithms. In Section 5, the case when g has a finite number of stationary points in $[a, b]$ is studied. We show how the modified FCC rules can be applied in this situation. An error estimate is provided, and some numerical results are given. Finally, we state a conclusion.

Some general remarks. Throughout the paper, C and C' stand for generic constants independent of k , and their values may differ from place to place. Their dependence on other parameters (if exist) is declared in each section. Note also that the space of m -times continuously differentiable functions is denoted by the common notation $C^m[a, b]$. Also, in all of the numerical experiments, the reference values of the sample integrals have been computed by Mathematica to at least 10 digits higher than the machine precision. All the numerical experiments are performed in MATLAB R2015b on the author's personal laptop.¹

2. The general idea. Consider the integral (1.1) with g having no stationary points in $[a, b]$. Then, we can assume that $g'(x) > 0$, $x \in [a, b]$, without loss of generality. By this assumption, the integral is changed into an integral with the linear oscillator. Indeed, by the

¹Intel Core i7-8550 CPU with a clock speed between 1.80 GHz and 1.99 GHz with 8GB of RAM.

change of variables $\tau = g(x)$, we obtain

$$\begin{aligned}
 I_k^{[a,b]}(f, g) &:= \int_{g(a)}^{g(b)} F(\tau) \exp(ik\tau) \, d\tau = l \exp(ikc) I_{\tilde{k}}(\tilde{F}) \\
 (2.1) \qquad \qquad &= l \exp(ikc) \int_{-1}^1 \tilde{F}(\tau) \exp(i\tilde{k}\tau) \, d\tau,
 \end{aligned}$$

where

$$\begin{aligned}
 (2.2) \qquad F(\tau) &= ((f/g') \circ g^{-1})(\tau), \quad \tau \in [g(a), g(b)] \\
 c &= \frac{g(b) + g(a)}{2}, \quad l = \frac{g(b) - g(a)}{2}, \\
 \tilde{k} &= lk, \quad \tilde{F}(\tau) = F(c + l\tau).
 \end{aligned}$$

The efficient and robust algorithm of [9] for constructing FCC rules can now be applied to (2.1). However, evaluating $g^{-1}(x)$ for some x is required in this process, and this is not desirable. For example, it may be impossible to express $g^{-1}(x)$ in closed form. In this situation, we would have to apply Newton iterations, whose convergence is not guaranteed. Even if they converge rapidly, this approach imposes extra errors and computational costs.

In the following, we develop a modification of the method in such a way that the need for computing $g^{-1}(x)$ is relaxed. The approach can then be employed in the presence of stationary points by means of composite rules on graded meshes as developed in [8]. The details are discussed in Section 5.

2.1. The modified Filon-Clelland-Curtis (MFCC) rule. Employing the $(N + 1)$ -point FCC rule (1.5) for computing $I_{\tilde{k}}(\tilde{F})$ (which essential is (2.1)) necessitates the evaluation of \tilde{F} at the Clenshaw-Curtis points. We describe a method for the accurate approximation of \tilde{F} at the Clenshaw-Curtis points in which g^{-1} does not have to be evaluated at any point.

Let N be a positive integer and consider the $N + 1$ Clenshaw-Curtis points $t_{0,N}, \dots, t_{N,N}$. Let $\ell_1 : [-1, 1] \rightarrow [a, b]$ and $\ell_2 : [-1, 1] \rightarrow [g(a), g(b)]$ be the onto affine-like mappings. By definition, $\tilde{F} = F \circ \ell_2$. For a large integer $N' > 1$, choose a set of arbitrary points $-1 = u_0 < \dots < u_{N'} = 1$, and define

$$(2.3) \qquad d_j := \ell_2^{-1}(g(\ell_1(u_j))), \quad j = 0, \dots, N'.$$

Then, $-1 = d_0 < \dots < d_{N'} = 1$. One can compute $\tilde{F}(d_j)$ without the need to compute d_j itself and to evaluate the inverse of g at any point. Indeed,

$$(2.4) \qquad \tilde{F}(d_j) = F(\ell_2(d_j)) = \frac{f(\ell_1(u_j))}{g'(\ell_1(u_j))}.$$

Now, we can approximate $\tilde{F}(t_{j,N})$ by interpolation of \tilde{F} at the nodes $d_0, \dots, d_{N'}$. Denote by $\tilde{Q}_{N'}$ the interpolating projection corresponding to some method of interpolation at the points $-1 = d_0 < \dots < d_{N'} = 1$, i.e.,

$$\tilde{Q}_{N'} \tilde{F}(d_j) = \frac{f(\ell_1(u_j))}{g'(\ell_1(u_j))}, \quad j = 0, \dots, N'.$$

Then,

$$\tilde{Q}_{N'} \tilde{F}(t_{j,N}) \approx \tilde{F}(t_{j,N}), \quad j = 0, \dots, N.$$

If $\tilde{k} \geq 1/2$, i.e., $k \geq 1/(g(b) - g(a))$, then the $(N + 1)$ -point MFCC rule is defined by

$$(2.5) \quad I_{k,N}^{[a,b]}(f, g) := l \exp(ikc) I_{\tilde{k},N}(\tilde{Q}_{N'} \tilde{F}) = l \exp(ikc) \sum_{n=0}^N \alpha_{n,N}(\tilde{Q}_{N'} \tilde{F}) \omega_n(\tilde{k}),$$

where the coefficients are expressed by (1.4), i.e.,

$$\alpha_{n,N}(\tilde{Q}_{N'} \tilde{F}) = \frac{2}{N} \sum_{j=0}^N \cos(jn\pi/N) \tilde{Q}_{N'} \tilde{F}(t_{j,N}), \quad n = 0, \dots, N.$$

Indeed, the vector $(\alpha_{n,N}(\tilde{Q}_{N'} \tilde{F}))_{n=0}^N$ is the discrete cosine transform (DCT) of type I of $(\tilde{Q}_{N'} \tilde{F}(t_{j,N}))_{j=0}^N$, which can be computed by the FFT algorithm in $\mathcal{O}(N \ln N)$ flops. Also, the weights $\omega_n(\tilde{k})$ are stably and rapidly computed by the two-phase algorithm [9].

In summary, the MFCC rule for computing (1.1) is defined as follows:

- If $\tilde{k} \geq 1/2$, then

$$I_k^{[a,b]}(f, g) = l \exp(ikc) I_{\tilde{k}}(\tilde{F}) \approx l \exp(ikc) I_{\tilde{k},N}(\tilde{F}) \approx l \exp(ikc) I_{\tilde{k},N}(\tilde{Q}_{N'} \tilde{F}).$$

- If $\tilde{k} < 1/2$, then the integral $I_{\tilde{k}}(\tilde{F})$ and thus $I_k^{[a,b]}(f, g)$ is no longer oscillatory and can be computed by the standard Clenshaw-Curtis rule:

$$I_{k,N}^{[a,b]}(f, g) := I_{0,N}(H_k), \quad H_k(x) := f(x) \exp(ikg(x)).$$

2.2. Error analysis. The accuracy of an MFCC rule is actually affected by the error of the involved interpolation method. In this section, we study how the interpolation error influences the total error of an MFCC rule. The results are then employed to choose efficient interpolation methods such that the total error of the MFCC rule decays rapidly.

For any integer $m \geq 0$ and a function φ defined on $[a, b]$, consider the weighted seminorm

$$\|\varphi\|_{H_w^m[a,b]} := \left\{ \int_a^b \frac{|\varphi^{(m)}(x)|^2}{\sqrt{(b-x)(x-a)}} dx \right\}^{1/2},$$

as introduced in [8]. Clearly, the Hilbert space $H_w^m[a, b]$ induced by $\|\cdot\|_{H_w^m[a,b]}$ contains $C^m[a, b]$.

LEMMA 2.1. *Let C_N and E_N be $(N + 1) \times (N + 1)$ matrices with the entries*

$$(2.6) \quad C_N(n + 1, j + 1) = (2/N) \cos(jn\pi/N), \quad n, j = 0, \dots, N,$$

and

$$E_N(n + 1, j + 1) = \begin{cases} 1/2 & \text{if } n = j \in \{0, N\}, \\ 1 & \text{if } n = j \in \{1, \dots, N - 1\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, $A_N := \sqrt{N/2} E_N^{1/2} C_N E_N^{1/2}$ is a symmetric orthogonal matrix.

Proof. Clearly, A_N is symmetric since C_N and E_N are so. On the other hand, $C_N E_N$ is nothing but the matrix of the DCT of type I. Thus, $(C_N E_N)^2 = (2/N) I_N$, hence,

$$A_N^\top A_N = \frac{N}{2} E_N^{1/2} C_N E_N C_N E_N E_N^{-1/2} = I_N. \quad \square$$

For all $r \in [0, 2]$, define

$$\rho(r) = \begin{cases} r, & 0 \leq r \leq 1, \\ (5r - 3)/2, & 1 \leq r \leq 2. \end{cases}$$

The following theorem provides an error estimate for the MFCC rules (2.5):

THEOREM 2.2. *Assume that f and g are smooth enough such that $F \in H_w^m[g(a), g(b)]$ for some integer $m \geq 1$. Then, for any $r \in [0, 2]$ with $m > \rho(r)$, the error of the $(N + 1)$ -point MFCC rule with $N \geq m - 1$ is estimated as follows:*

$$(2.7) \quad \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[a,b]}(f, g) \right| \leq Ck^{-r}h^{m+1-r}N^{-m+\rho(r)} + C'k^{-1}\sqrt{N} \sup_{\tau \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'}\tilde{F})(\tau) \right|,$$

where $h := b - a$ is the length of the integration interval and C, C' are constants independent of N and N' .

Proof. In the worst case when $I_k^{[a,b]}(f, g)$ is oscillatory, i.e., $\tilde{k} \geq 1/2$, the total error is bounded by

$$(2.8) \quad \left| I_k^{[a,b]}(f, g) - l \exp(ikc) I_{\tilde{k},N}(\tilde{Q}_{N'}\tilde{F}) \right| \leq \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[g(a),g(b)]}(F) \right| + l \left| I_{\tilde{k},N}(\tilde{F} - \tilde{Q}_{N'}\tilde{F}) \right|.$$

(If $\tilde{k} < 1/2$, then the second term on the right-hand-side of (2.8) drops.) By assumption, $F \in H_w^m[g(a), g(b)]$. According to [8, Theorem 2.6],

$$(2.9) \quad \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[g(a),g(b)]}(F) \right| \leq C\sigma_{m,N}k^{-r}l^{m+1-r}N^{-m+\rho(r)},$$

where C is a constant independent of m and N , and

$$\sigma_{m,N} = \begin{cases} \prod_{j=1}^{m-1} \left(\frac{N+1}{N+1-j} \right), & m > 1, \\ 1, & m = 1. \end{cases}$$

For a fixed m , one can see that $\sigma_{m,N} \rightarrow 1$ as $N \rightarrow \infty$. Hence, $C\sigma_{m,N}$ is bounded by a constant independent of N , and thus by the mean value theorem for g ,

$$(2.10) \quad \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[g(a),g(b)]}(F) \right| \leq Ck^{-r}l^{m+1-r}N^{-m+\rho(r)} \leq Ck^{-r}h^{m+1-r}N^{-m+\rho(r)},$$

where C is independent of N .

On the other hand,

$$(2.11) \quad \left| I_{\tilde{k},N}(\tilde{F} - \tilde{Q}_{N'}\tilde{F}) \right| = |\boldsymbol{\omega}_N^\top \mathbf{C}_N \boldsymbol{\delta}_N|,$$

where C_N is defined by (2.6) and ω_N, δ_N are column vectors defined by

$$\begin{aligned} \omega_N &= [\omega_0(\tilde{k})/2, \omega_1(\tilde{k}), \dots, \omega_{N-1}(\tilde{k}), \omega_N(\tilde{k})/2]^\top, \\ \delta_N &= \left[\frac{(\tilde{F} - \tilde{Q}_{N'}\tilde{F})(t_{0,N})}{2}, (\tilde{F} - \tilde{Q}_{N'}\tilde{F})(t_{1,N}), \dots, \right. \\ &\quad \left. \dots, (\tilde{F} - \tilde{Q}_{N'}\tilde{F})(t_{N-1,N}), \frac{(\tilde{F} - \tilde{Q}_{N'}\tilde{F})(t_{N,N})}{2} \right]^\top. \end{aligned}$$

(See [9, Remark 2.1]). By the Cauchy-Schwarz inequality, equation (2.11) yields

$$(2.12) \quad \left| I_{\tilde{k},N}(\tilde{F} - \tilde{Q}_{N'}\tilde{F}) \right| \leq \|\omega_N\|_2 \|C_N \delta_N\|_2.$$

The well-known asymptotic expansion of highly oscillatory integrals in the absence of stationary points (see, e.g., [14]) yields $|I_{\tilde{k}}(T_n)| = \mathcal{O}(\tilde{k}^{-1})$, and thus,

$$(2.13) \quad \|\omega_N\|_2 \leq C' l^{-1} k^{-1} \sqrt{N}.$$

Note that $\|T_n\|_\infty \leq 1$ for all n , so C' in (2.13) can be assumed to be independent of N and N' . On the other hand, Lemma 2.1 implies that

$$(2.14) \quad \|C_N E_N\|_2 \leq \left\| E_N^{-1/2} \right\|_2 \left\| E_N^{1/2} C_N E_N^{1/2} \right\|_2 \left\| E_N^{1/2} \right\|_2 = \frac{2}{\sqrt{N}}.$$

The upper bound (2.12) can be estimated by (2.13) and (2.14):

$$(2.15) \quad \begin{aligned} \left| I_{\tilde{k},N}(\tilde{F} - \tilde{Q}_{N'}\tilde{F}) \right| &\leq \|\omega_N\|_2 \|C_N \delta_N\|_2 \leq \|\omega_N\|_2 \|C_N E_N\|_2 \|\mathbf{y}_N\|_2 \\ &\leq \frac{2}{\sqrt{N}} \|\omega_N\|_2 \|\mathbf{y}_N\|_2 \leq C' l^{-1} k^{-1} \sqrt{N} \sup_{\tau \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'}\tilde{F})(\tau) \right|, \end{aligned}$$

where $\mathbf{y}_N := E_N^{-1} \delta_N$. The result now follows by (2.8), (2.10), and (2.15). \square

The following corollary is immediately obtained from Theorem 2.2:

COROLLARY 2.3. *Assume that f and g are smooth enough such that $F \in H_w^{N+1}[g(a), g(b)]$ for some integer $N \geq 1$. Then, for any $r \in [0, 2]$, the error of the $(N + 1)$ -point MFCC rule is estimated as follows:*

$$(2.16) \quad \begin{aligned} \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[a,b]}(f, g) \right| &\leq C k^{-r} h^{N+2-r} \frac{N^{-1+\rho(r)}}{N! 2^N} \|g'\|_\infty^N \\ &\quad + C' k^{-1} \sqrt{N} \sup_{\tau \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'}\tilde{F})(\tau) \right|, \end{aligned}$$

where $h := b - a$, C, C' are constants independent of N, N' , and $\|g'\|_\infty = \sup_{x \in [a,b]} |g'(x)|$.

Proof. Note that

$$\sigma_{N+1,N} = \frac{N^N}{N!} \left(1 + \frac{1}{N} \right)^N$$

and $(1 + 1/N)^N$ is bounded for all $N \geq 1$. Thus, by the mean value theorem for g , equation (2.9) with $m = N + 1$ is adjusted as

$$\begin{aligned} \left| I_k^{[a,b]}(f, g) - I_{k,N}^{[g(a),g(b)]}(F) \right| &\leq C k^{-r} \left(\frac{h \|g'\|_\infty}{2} \right)^{N+2-r} \frac{N^{-1+\rho(r)}}{N!} \\ &= C k^{-r} h^{N+2-r} \frac{N^{-1+\rho(r)}}{N! 2^N} \|g'\|_\infty^N. \quad \square \end{aligned}$$

As it is seen by (2.7) or (2.16), the total error of the MFCC rule is bounded by the sum of two terms: The first term is nothing but the error of the FCC rule, studied in detail in [8, 9]. It does not depend on N' and decays rapidly as k or N grow. The second term is associated with the interpolation process, whose error may be quite large due to the Runge effect. In the sequel, we present two strategies in order to treat the latter problem and obtain error bounds which decay rapidly by increasing N or N' .

3. The case of no stationary points. One popular remedy for treating the Runge phenomenon lies in splines. In this section, we present a method of integration based on *composite* MFCC rules on uniform meshes. Without loss of generality, we consider the integral (1.1) over the interval $[0, 1]$. Recall from the previous section the assumption that $g'(x) > 0$, $x \in [0, 1]$.

3.1. Algorithm I. Take a large integer M and divide the integration interval $[0, 1]$ into M subintervals of equal lengths h :

$$x_n := nh, \quad n = 0, \dots, M, \quad h := 1/M.$$

Then,

$$g(0) = g(x_0) < \dots < g(x_M) = g(1).$$

Assume that $f \in C^{N+1}[0, 1]$ and $g \in C^{N+2}[0, 1]$ for some integer $N \geq 1$. Then it is easy to see that F , as defined by (2.2), lies in $C^{N+1}[g(0), g(1)]$. In each panel $[x_{n-1}, x_n]$, we employ the $(N + 1)$ -point MFCC rule (2.5) with $N' = N$, $u_j = t_{N-j, N}$, and \tilde{Q}_N being the Lagrange interpolating projection at the points $-1 = d_0 < \dots < d_N = 1$. The accuracy of this kind of composite MFCC rules may be studied as follows.

3.1.1. Error analysis. Take an arbitrary subinterval $[x_{n-1}, x_n]$, and for a simplification of the notation, set $[a, b] := [x_{n-1}, x_n]$. We divide our discussion into two parts: first we show that d_0, \dots, d_N are ‘good’ interpolation points for small $[a, b]$ in the sense that the corresponding Lebesgue constant grows only logarithmically with N provided that M increases accordingly with N . In the second part, the total error of the composite MFCC rule is estimated by Corollary 2.3 and an error bound for the interpolant $\tilde{Q}_N F$ associated with each panel.

Estimating the Lebesgue constant. By the mean value theorem, (2.3) yields

$$(3.1) \quad d_i - d_j = \frac{b - a}{g(b) - g(a)} g'(\eta_{ij})(t_{N-i, N} - t_{N-j, N}),$$

for any $i, j \in \{0, \dots, N\}$ and some $\eta_{ij} \in (a, b)$. Since $d_0 = t_{N, N} = -1$ and $d_N = t_{0, N} = 1$, equation (3.1) implies that the points d_0, \dots, d_N can be considered ‘perturbed Clenshaw-Curtis points’ when M is sufficiently large. This is because g' is positive and uniformly continuous on $[a, b]$, so

$$\frac{b - a}{g(b) - g(a)} g'(\eta_{ij}) \simeq 1,$$

by the mean value theorem.

The Lebesgue constant for a set Π_N of $N + 1$ interpolation points in an interval \mathcal{I} is defined by

$$\max_{x \in \mathcal{I}} \sum_{i=0}^N |L_i(x)|,$$

where L_i is the $(i + 1)$ -st Lagrange cardinal polynomial for the set Π_N . It is well-known that the Lebesgue constant determines how relatively close the maximum error of an interpolation is to that of the best approximation. Denote by Λ_N and $\tilde{\Lambda}_N$ the Lebesgue constants associated with the Clenshaw-Curtis points $t_{N,N}, \dots, t_{0,N}$ and the perturbed Clenshaw-Curtis points d_0, \dots, d_N , respectively. We show that $\tilde{\Lambda}_N \leq e\Lambda_N$ provided that M increases accordingly with N .

Consider the change of variables $\tau = \ell_2^{-1}(g(\ell_1(x)))$. Then (3.1) implies that the $(i + 1)$ -st Lagrange cardinal polynomial \tilde{L}_i associated with the points d_0, \dots, d_N can be written as follows:

$$(3.2) \quad \tilde{L}_i(\tau) = \prod_{\substack{j=0 \\ j \neq i}}^N \frac{\tau - d_j}{d_i - d_j} = \prod_{\substack{j=0 \\ j \neq i}}^N \frac{g'(\eta_j(\tau))}{g'(\eta_{ij})} \frac{x - t_{N-j,N}}{t_{N-i,N} - t_{N-j,N}},$$

where $\eta_j(\tau) \in (a, b)$ for all $j = 0, \dots, N$ and $\tau \in [-1, 1]$. Positivity and continuity of g' on the compact interval $[0, 1]$ imply that there exists $h_N > 0$ such that for all subintervals $[a, b]$ with $b - a < h_N$,

$$(3.3) \quad 1 - \frac{1}{N} < \frac{g'(\eta_j(\tau))}{g'(\eta_{ij})} < 1 + \frac{1}{N}, \quad \tau \in [-1, 1], \quad i, j = 0, \dots, N.$$

Thus, (3.2) with (3.3) implies that

$$(3.4) \quad (1 - 1/N)^N \sum_{i=0}^N |L_i(x)| < \sum_{i=0}^N |\tilde{L}_i(\tau)| < (1 + 1/N)^N \sum_{i=0}^N |L_i(x)|,$$

where L_i is the $(i + 1)$ -st Lagrange cardinal polynomial for the Clenshaw-Curtis points. Since $\tau = \ell_2^{-1}(g(\ell_1(x)))$ is a bijective mapping on $[-1, 1]$, taking the maximum over $x \in [-1, 1]$ (or over $\tau \in [-1, 1]$) in (3.4) yields

$$(3.5) \quad (1 - 1/N)^N \Lambda_N < \tilde{\Lambda}_N < (1 + 1/N)^N \Lambda_N.$$

Since $(1 + 1/N)^N$ is bounded, (3.5) implies that the Runge effect does not occur in the interpolation of \tilde{F} at the points d_0, \dots, d_N , and the error is comparable to that of Chebyshev interpolation provided that $M \geq 1/h_N$, i.e., M is taken large enough in relation to N .

The total error bound. Clearly, the interpolating error is bounded by

$$\sup_{x \in [-1, 1]} \left| (\tilde{F} - \tilde{Q}_N \tilde{F})(x) \right| \leq (2l)^{N+1} \max_{x \in [g(a), g(b)]} \left| F^{(N+1)}(x) \right|.$$

This is because $F \in C^{N+1}[g(a), g(b)]$, and hence $\tilde{F} \in C^{N+1}[-1, 1]$. Now, by the mean value theorem for g , we have

$$\sup_{x \in [-1, 1]} \left| (\tilde{F} - \tilde{Q}_N \tilde{F})(x) \right| \leq C' h^{N+1},$$

for N large enough. Here, C' is a generic constant independent of M but dependent on N . Since $\tilde{F} \in C^{N+1}[-1, 1] \subseteq H_w^{N+1}[-1, 1]$, Corollary 2.3 can now be applied to estimate the total error on an arbitrary panel $[a, b]$:

$$(3.6) \quad \left| I_k^{[a,b]}(f, g) - l \exp(ikc) I_{k,N}(\tilde{Q}_N \tilde{F}) \right| \leq C k^{-r} h^{N+2-r} \frac{N^{-1+\rho(r)}}{N! 2^N} \|g'\|_\infty^N + C' k^{-1} h^{N+1},$$

TABLE 3.1

Application of Algorithm I to the model integral $I_{100}^{[0,1]}(f, g)$ with f and g defined by (3.8): Absolute error with its decay rate.

M	$N = 1$		$N = 2$		$N = 3$	
	error	rate	error	rate	error	rate
2	1.87×10^{-4}	—	1.27×10^{-5}	—	2.22×10^{-6}	—
4	2.72×10^{-5}	2.8	3.87×10^{-6}	1.7	4.43×10^{-7}	2.3
8	3.42×10^{-5}	-0.3	8.11×10^{-8}	5.6	3.50×10^{-8}	3.7
16	3.98×10^{-5}	-0.2	1.40×10^{-7}	-0.8	1.41×10^{-9}	4.6
32	3.69×10^{-6}	3.4	2.37×10^{-9}	5.9	1.66×10^{-11}	6.4
64	8.25×10^{-7}	2.2	1.25×10^{-10}	4.2	7.41×10^{-13}	4.5

where C is independent of M and N . If one takes the sum over all the subintervals $[x_{n-1}, x_n]$ and takes into account that $F \in C^{N+1}[g(0), g(1)]$, then an error bound for Algorithm I is obtained by (3.6):

$$(3.7) \quad Ck^{-r}h^{N+1-r} \frac{N^{-1+\rho(r)}}{N!2^N} \|g'\|_{\infty}^N + C'k^{-1}h^N.$$

REMARK 3.1. Note that C' in the error bound (3.7) may grows with N . Thus, the recommended strategy in Algorithm I is to keep N fixed at a moderate value and increase M until the required accuracy is achieved. This can, in addition, suppress the Runge effect as discussed above.

A few words about the complexity of the algorithm is also necessary. The $(N + 1)$ -point FCC rule can be stably computed with $\mathcal{O}(N \ln N)$ flops [9]. The Lagrange interpolation can also be performed stably by an improved formula, the so-called “first form of the barycentric interpolation formula” (see [3, 11, 26]). The evaluation of the formula at any point involve costs of order $\mathcal{O}(N^2)$, so the evaluation of all the values $\tilde{Q}\tilde{F}(t_{j,N})$, $j = 0, \dots, N$, requires $\mathcal{O}(N^3)$ flops. Since this cost should be paid for each panel, and there exist M panels, Algorithm I requires $\mathcal{O}(MN(N^2 + \ln N))$ flops.

3.1.2. Numerical experiments. Throughout this section, we consider the model integral $I_k^{[0,1]}(f, g)$ with

$$(3.8) \quad f(x) = \frac{x^{4.5}}{1+x^2}, \quad g(x) = \sqrt{x^2 + 3x + 4},$$

for some $k > 0$. It can be seen that $g'(x) > 0$, $x \in [0, 1]$, $f \in C^4[0, 1]$, and $g \in C^\infty[0, 1]$. We employ Algorithm I and illustrate the error estimate (3.7).

Experiment 1. Here we study the case when k and N are kept fixed and $h \rightarrow 0$. According to the error bound (3.7), the convergence of the method is of order $\mathcal{O}(h^N)$, which can be illustrated by the following example. Let $k = 100$. Since $f \in C^4[0, 1]$ and $g \in C^\infty[0, 1]$, we have that $F \in C^4[g(0), g(1)]$. Thus, the error bound (3.7) is valid if the parameter N does not exceed 3. The absolute error with its decay rate, as $h \rightarrow 0$, is given in Table 3.1 for $N = 1, 2, 3$. As mentioned in Section 3.1, the distribution of the interpolation nodes in each panel is so that the corresponding Lebesgue constant is rather small. Thus, the convergence rate will be higher than what we expect. For coarser grids (corresponding to larger h), instead the convergence rate of $\mathcal{O}(h^N)$ is not to be expected. Considering all these facts together, the numerical results of Table 3.1 are in agreement with the theory.

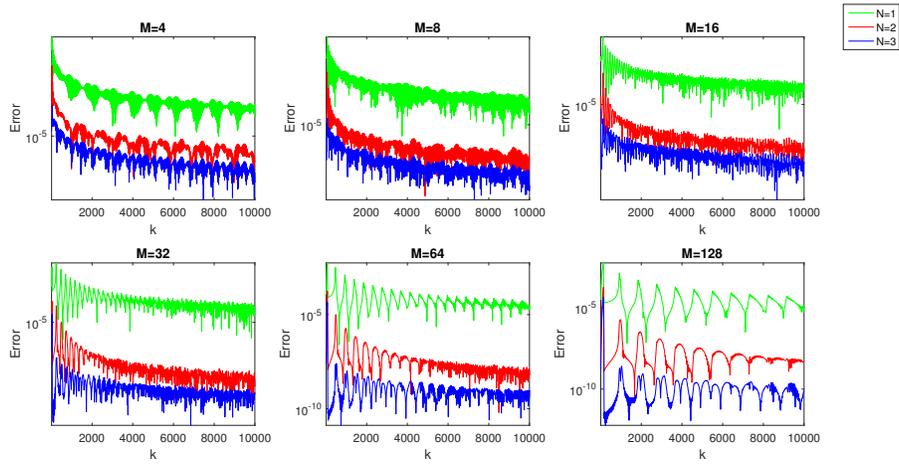


FIG. 3.1. Absolute error scaled by k of Algorithm I with some M and N , when applied to the model integral $I_k^{[0,1]}(f, g)$ with (3.8).

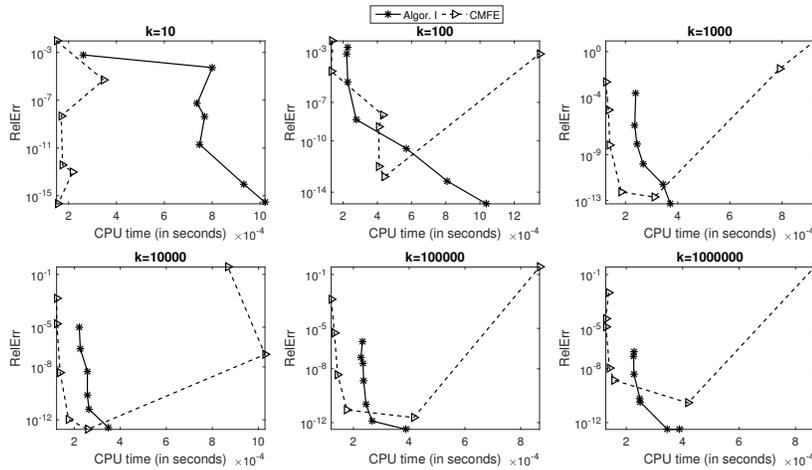


FIG. 3.2. Algorithm I (solid line) and the CMFE rule of [15] with exponential order of convergence (dashed line) when applied to the integral (3.9): Relative error vs. the execution time (in seconds).

Experiment 2. The error bound (3.7) suggests that for a fixed parameters N and M , the maximum rate of convergence is $\mathcal{O}(k^{-1})$ as $k \rightarrow \infty$. In this experiment, we illustrate this rate by a numerical example. For each $M \in \{4, 8, 16, 32, 64, 128\}$ and $N \in \{1, 2, 3\}$, we apply Algorithm I to the model integral $I_k^{[0,1]}(f, g)$ with (3.8) when k varies in a wide band from 10 to 10^4 . The absolute error is scaled by k , and the scaled error for each N and M is plotted as a function of k (Figure 3.1). As it is seen, the error for each M and N does not deteriorate as k increases, and this observation is in agreement with the maximum convergence order $\mathcal{O}(k^{-1})$ suggested by the theory.

Experiment 3. Here, we compare the convergence rate of Algorithm I with the composite Filon-type rules [15] for the integrals $I_k^{[0,1]}(f, g)$ with smooth f and g . For a given integer $n \geq 1$, consider the graded mesh $x_0 = 0, x_j = k^{(j-1)/(n-1)-1}, j = 1, \dots, n$. For

the j -th panel $[x_{j-1}, x_j]$, define the parameters $M_j := \lceil \max\{|g'(x_{j-1})|, |g'(x_j)|\} \rceil$ and $N_j := n(n-1)/(n+1-j)$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x . Then, apply Algorithm I to the integral over $[x_{j-1}, x_j]$ with $N_j + 1$ Clenshaw-Curtis points and a uniform mesh with M_j panels. This is a special kind (with Clenshaw-Curtis points) of the CMFE rule proposed in [15] for $I_k^{[0,1]}(f, g)$ with smooth f . In the absence of harmful rounding, the rule converges exponentially. More precisely, when $n \rightarrow \infty$, the error of the rule scales as $\mathcal{O}((n-1)^{-1/2}k^{-n-1})$ provided that

$$(n-1)(\ln(n-2+e)-1) \geq \ln k.$$

We apply Algorithm I and the CMFE rule, as described above, to the following sample integral for $k = 10^d$, $d = 1, \dots, 6$:

$$(3.9) \quad \int_0^1 \exp(ik(\sin(\pi x/2) + 2x)/3) \, dx.$$

Figure 3.2 displays the relative error versus the execution time (in seconds) for each case. In this example, the CMFE rule is in general faster than Algorithm I because it needs a smaller number of subintervals to reach a given accuracy. This is due to the special graded meshes which result in the exponential order of convergence of the CMFE rule. When higher accuracy is required, however, it may exhibit unstable behavior due to rounding errors in the first few panels. Indeed, the panels in a narrow neighborhood of 0 become too small for larger k . For example, one can see that the first panel is $[0, 1/k]$, which in turn is partitioned by a uniform mesh of size M_1 , and in each subpanel, $N_1 := n-1$ interpolation points are involved. Thus, one has to evaluate functions at a group of very close points, and this may deteriorate the accuracy due to rounding. In the sample integral (3.9), f is constant; if it were not so, then this effect might be even more apparent.

3.2. Algorithm II. In this section, we follow [4] and employ an efficient interpolation method which does not allow for the Runge effect. In addition, one can always increase its accuracy as high as desired without increasing the computational cost.

Consider the integral (1.1) on an arbitrary interval $[a, b]$ (not necessarily of small length). Here, we do not divide the integration interval $[a, b]$ into smaller subintervals as in Algorithm I. Instead, we consider the $(N+1)$ -point MFCC rule on the whole interval and approximate the values $\tilde{F}(t_{j,N})$ by interpolating \tilde{F} at a limited number of points, selected from $\{d_0, \dots, d_{N'}\}$, in such a way that they surround $t_{j,N}$.

For a large integer N' , choose a set of arbitrary points $-1 = u_0 < \dots < u_{N'} = 1$ and consider the points $-1 = d_0 < \dots < d_{N'} = 1$ as defined by (2.3). Choose a positive integer $s < N'$ and fix it throughout the process. For each $j = 0, \dots, N$, select an s -tube $\mathcal{N}_{n,s} = (d_n, \dots, d_{n+s-1})$ such that $d_n \leq t_{j,N} \leq d_{n+s-1}$. Note that n depends (not uniquely) on j . Thus, such a selection is not necessarily unique, while it is always possible since $s < N'$. In [4], some selecting strategies are introduced, and in [16] a MATLAB code for a certain selecting strategy has been provided. Now, consider the approximation

$$\tilde{F}(t_{j,N}) \approx p_{n,s}(t_{j,N}), \quad j = 0, \dots, N,$$

where $p_{n,s}$ is the Lagrange interpolation polynomial of \tilde{F} at the points of $\mathcal{N}_{n,s}$. This means that $p_{n,s}$ is of at most degree $s-1$, and it may change with j .

For computing $p_{n,s}(t_{j,N})$, we recommend the so-called “first form of the barycentric interpolation formula” [26] that is a reformulation of the Lagrange interpolation polynomials. In [11] it has been proved that the formula is backward stable for any set of interpolation points.

Based on (2.5), one can introduce the rule

$$(3.10) \quad I_k^{[a,b]}(f, g) \approx l \exp(ikc) \sum_{n=0}^N \alpha_{n,N}(p_{n,s}) \omega_n(\tilde{k}),$$

where $(\alpha_{n,N}(p_{n,s}))_{n=0}^N$ is the DCT of type I of $(p_{n,s}(t_{j,N}))_{j=0}^N$.

3.2.1. Error analysis. If $f \in C^s[a, b]$ and $g \in C^{s+1}[a, b]$, it follows that $\tilde{F} \in C^s[-1, 1]$, and thus

$$(3.11) \quad \max_{0 \leq n \leq N} |(\tilde{F} - p_{n,s})(t_{j,N})| \leq C' \lambda^s,$$

where

$$\lambda = \max_{1 \leq j \leq N'} |d_j - d_{j-1}|,$$

and C' is independent of N and N' , but dependent of s (see, e.g., [4]).

On the other hand, $F \in C^s[g(a), g(b)] \subseteq H_w^s[g(a), g(b)]$. Thus, for any $r \in [0, 2]$, $s > \rho(r)$, and $N \geq s - 1$, one can obtain the total error of the rule (3.10) by Theorem 2.2 and (3.11) as

$$(3.12) \quad Ck^{-r}(b-a)^{s+1-r} N^{-s+\rho(r)} + C'k^{-1} \lambda^s \sqrt{N},$$

where C is independent of N, N' .

One can easily see from (3.1) that λ decreases with the rate of $\mathcal{O}(1/N')$ as $N' \rightarrow \infty$. Therefore, if we take $N' \geq kN$, the error bound (3.12) is reduced to

$$(3.13) \quad Ck^{-r}(b-a)^{s+1-r} N^{-s+\rho(r)} + C'k^{-s-1} N^{-s+1/2}.$$

The discussion about the complexity of the method is the same as that of Algorithm I. Since the total interpolation process requires $\mathcal{O}(Ns^2)$ flops, the whole algorithm is performed at the cost of $\mathcal{O}(N(\ln N + s^2))$. In order to increase the accuracy of the interpolation, one needs to increase N' only, while the parameter s is usually kept fixed at a moderate value. In practice, one should compute only Ns members of $\{d_0, \dots, d_{N'}\}$, not all of them. Thus, the cost of computation does not grow with N' . Also, the $(N + 1)$ -point FCC rule converges rapidly as N increases, so by a moderate value of N , one can reach a rather high accuracy. Therefore, the integral (1.1) can be accurately approximated by Algorithm II at a rather low cost. We carry out a set of numerical experiments in Section 4 to illustrate our claim here.

3.2.2. Numerical experiments. Throughout this section, we consider the model integral $I_k^{[-1,1]}(f, g)$ with

$$(3.14) \quad f(x) = \frac{x-1}{1+x^2}, \quad g(x) = \sqrt{x^2 + 3x + 4},$$

and some k . It is easy to see that $g'(x) > 0$, for $x \in [-1, 1]$, and $f, g \in C^s[-1, 1]$ for all $s \geq 1$. Thus, Algorithm II can be applied. The aim of this section is to illustrate the convergence estimate (3.13).

Experiment I. If k is kept fixed, then the error bound (3.13) decays with the maximum rate of $\mathcal{O}(N^{-s+1/2})$ as N increases. We consider the assumed model integral with $k = 100$ and employ the algorithm with $s = 2, 3, 4, 5$, some increasing N , and $N' = kN$. The absolute error with its decay rate is given in Table 3.2. As it is seen, the experimental convergence rate for each s is in agreement with the theoretical one.

TABLE 3.2

Application of Algorithm II with $N' = 100N$ to the model integral $I_{100}^{[-1,1]}(f, g)$ with f and g defined by (3.14): Absolute error with its decay rate.

N	$s = 2$		$s = 3$		$s = 4$		$s = 5$	
	error	rate	error	rate	error	rate	error	rate
2	6.42×10^{-4}	—	6.42×10^{-4}	—	6.42×10^{-4}	—	6.42×10^{-4}	—
4	5.42×10^{-4}	0.2	5.42×10^{-4}	0.2	5.42×10^{-4}	0.2	5.42×10^{-4}	0.2
8	9.91×10^{-5}	2.5	9.91×10^{-5}	2.5	9.91×10^{-5}	2.5	9.91×10^{-5}	2.5
16	2.93×10^{-6}	5.1	2.93×10^{-6}	5.1	2.93×10^{-6}	5.1	2.93×10^{-6}	5.1
32	1.48×10^{-8}	7.6	1.77×10^{-9}	10.7	1.73×10^{-9}	10.7	1.73×10^{-9}	10.7
64	1.21×10^{-9}	3.6	1.71×10^{-12}	10.0	2.22×10^{-15}	19.6	1.17×10^{-15}	20.5

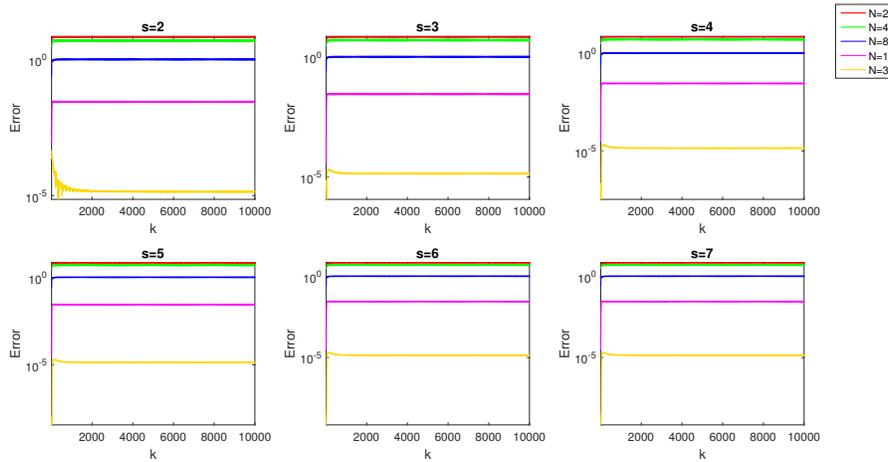


FIG. 3.3. Absolute error scaled by k^2 of Algorithm II with some s and N , when applied to the model integral $I_k^{[-1,1]}(f, g)$ with (3.14).

Experiment 2. If N , N' , and s are kept fixed, then the error estimate (3.13) suggests that the rate of convergence as $k \rightarrow \infty$ is not higher than $\mathcal{O}(k^{-2})$ provided that $N' \geq kN$. Here, we consider the model integral $I_k^{[-1,1]}(f, g)$ with (3.14) when k varies in a wide band from 10 to 10^4 . Algorithm II with $s = 2, \dots, 7$ and $N = 2, 4, 8, 16, 32$ is applied while $N' = kN$. The absolute error is scaled by k^2 , and the scaled error for each s and N is plotted as a function of k (see Figure 3.3). The horizontal trend of the scaled error, corresponding to each pair (s, N) , illustrates the maximum convergence rate of $\mathcal{O}(k^{-2})$.

4. Comparisons. In this section, we compare Algorithms I and II to see which one achieves a given accuracy faster in practice. Our answer to this question is based on some numerical results accompanied by a rough theoretical discussion.

Recall from Section 3.1 that Algorithm I requires $\mathcal{O}(MN(N^2 + \ln N))$ flops. From the error bound (3.7), one can see that a rather large N may reduce the error effectively even if $h < 1$ is not too small. This is because N is appearing in the exponents. Also, it is not recommended to use a very large N since the complexity of the algorithm increases rapidly. Thus, it is preferable to choose moderate values for N while M should be rather small.

Similarly, we recall from Section 3.2 that the number of required flops for Algorithm II is $\mathcal{O}(N(\ln N + s^2))$. Note that the exponents in the error bound (3.13) may not be as large

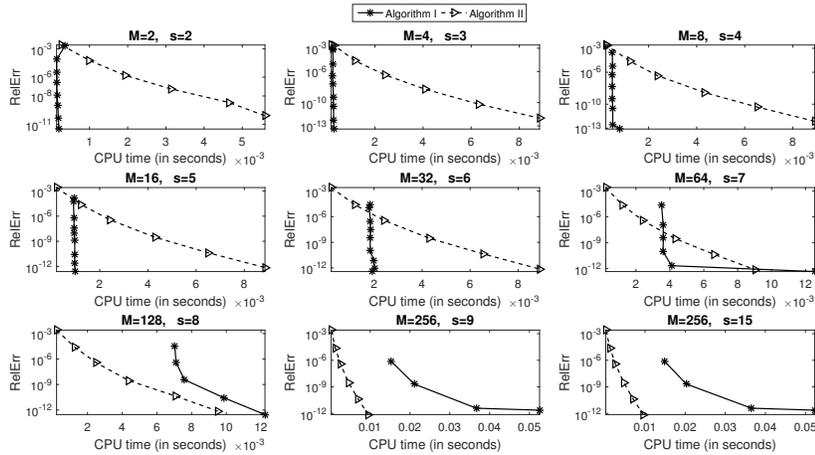


FIG. 4.1. Average CPU time (in seconds) vs. the relative error of Algorithms I and II when applied to the model integral (4.1). The number M of subintervals in Algorithm I and the interpolation parameter s in Algorithm II are kept fixed in each plot while the accuracy increases with growing N . In Algorithm II, $N' = kN$.

as those in the error bound (3.7) because in practice, s usually takes on rather small values; larger values increase the complexity of the algorithm rapidly.

Based on the discussion above, our guess is that Algorithm I may be faster than Algorithm II provided that M is rather small. In the following, we carry out a set of numerical experiments to support this guess.

Experiment 1. Recalling the model integral in Section 3.2.2 with $k = 1000$, consider the integral

$$(4.1) \quad \int_{-1}^1 \frac{x-1}{1+x^2} \exp\left(1000i\sqrt{x^2+3x+4}\right) dx.$$

In Algorithm I, M is kept fixed, and the accuracy increases by letting $N \rightarrow \infty$ (Figure 4.1). The algorithm for each set of the parameters $\{M, N\}$ is executed 10 times, and the average of the CPU times (in seconds) versus the relative error is displayed as a solid line. In Algorithm II, s is kept fixed and the accuracy increases by letting $N \rightarrow \infty$ while $N' = kN$. The algorithm for each set of the parameters $\{s, N\}$ is executed 10 times, and the average of the CPU times (in seconds) versus the relative error is displayed by a dashed line. Figure 4.1 consists of nine subplots for some increasing values of M (for Algorithm I) and s (for Algorithm II). As it is seen, Algorithm I is more accurate only when M is not too large.

Experiment 2. Here, we repeat Experiment 1 now for the model integral of Section 3.1.2 with $k = 100$, i.e.,

$$(4.2) \quad \int_0^1 \frac{x^{4.5}}{1+x^2} \exp\left(100i\sqrt{x^2+3x+4}\right) dx.$$

Note that the amplitude function is of class $C^4[0, 1]$. Thus, the error bound (3.7) of Algorithm I is valid when $N \leq 3$, and the error bound (3.12) of Algorithm II is valid when $s \leq 4$. Regardless of the error bounds, we employ both of the algorithms with larger N and s (Figure 4.2). Here again, as in Experiment 1, it is seen that Algorithm I is more accurate only when M is not too large.

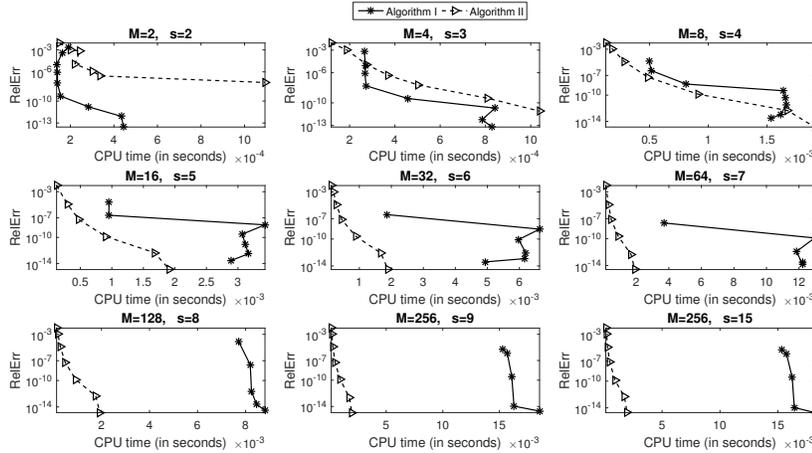


FIG. 4.2. Average CPU time (in seconds) vs. the relative error of Algorithms I and II when applied to the model integral (4.2). The number M of subintervals in Algorithm I and the interpolation parameter s in Algorithm II are kept fixed in each plot while the accuracy increases with growing N . In Algorithm II, $N' = kN$.

5. The case of the presence of stationary points. Consider the integral (1.1), and assume that $g'(x)$ vanishes at a finite number of points in $[a, b]$. Recall that a function φ has a stationary point of order $n \geq 1$ at $\xi \in [a, b]$ if

$$\varphi'(\xi) = \dots = \varphi^{(n)}(\xi) = 0, \quad \varphi^{(n+1)}(\xi) \neq 0.$$

Assume that g has a finite number of stationary points in $[a, b]$. Then, we can always divide $[a, b]$ into subintervals such that g has a single stationary point in each subinterval located at one of its endpoints. Thus, without loss of generality, we consider the integral $I_k^{[0,1]}(f, g)$, where g has a single stationary point of order n at 0 and $g'(x) > 0$ for any $x \in (0, 1]$.

Under the change of variables $\tau = g(x)$, the integral is reduced to $I_k^{[g(0), g(1)]}(F)$, with F defined by (2.2). Again by the change of variables $\tau = g(0) + 2l\hat{x}$, with $l = (g(1) - g(0))/2$, the integral is transformed into one over the standard interval $[0, 1]$:

$$I_k^{[g(0), g(1)]}(F) = 2l \exp(ikg(0)) I_{2lk}^{[0,1]}(\widehat{F}),$$

where

$$\widehat{F}(\hat{x}) = F(g(0) + 2l\hat{x}).$$

By [8, Theorem 4.1], if f and g are smooth enough, then $\widehat{F} \in C_{\beta}^m[0, 1]$, for $\beta = -n/(n+1)$ and some m , depending on the degrees of smoothness of f and g . Recall from [8] that for any $\beta < 0$, $C_{\beta}^m[0, 1]$ is defined as the space of all $\varphi \in C(0, 1]$ such that

$$\|\varphi\|_{m,\beta} := \max \left\{ \sup_{x \in (0,1]} \left| x^{j-\beta} \varphi^{(j)}(x) \right|, \quad j = 0, \dots, m \right\} < \infty.$$

Now the composite algorithm, introduced in [8], can be employed for computing $I_{2lk}^{[0,1]}(\widehat{F})$. The algorithm uses the classical graded mesh

$$(5.1) \quad \Pi_{M,q} := \left\{ x_j := \left(\frac{j}{M} \right)^q : j = 0, 1, \dots, M \right\},$$

for some parameter $q > 1$ sufficiently large. Then in each panel $[x_{j-1}, x_j]$, the $(N + 1)$ -point FCC rule is applied.

Our proposed algorithm here is almost the same. The only difference is that we employ the $(N + 1)$ -point MFCC rule in each panel. In the following we show how the $(N + 1)$ -point MFCC rule may be applied, and how it affects the total error of the algorithm.

5.1. The composite MFCC rules on graded meshes. We divide the integration interval $[0, 1]$ by the graded mesh $\Pi_{M,q}$ into M panels, and in each panel we use the $(N + 1)$ -point MFCC rule (2.5) with $N' = N$ and $u_j = t_{N-j,N}$, i.e., $N + 1$ Clenshaw-Curtis points. As it is seen, the only difference of the algorithm here from Algorithm I lies in the partitioning $0 = x_0 < \dots < x_M = 1$: uniform meshes for Algorithm I and graded meshes for the current algorithm. The following theorem provides an error bound for the composite MFCC rules on graded meshes.

THEOREM 5.1. *Let f and g be smooth enough such that $\widehat{F} \in C_\beta^{N+1}[0, 1]$, and let g have a single stationary point at 0 of order $n \geq 1$. Let $0 \leq r < 1 + \beta$ with $\beta = -n/(n + 1)$, and choose*

$$(5.2) \quad q > (N + 1 - r)/(\beta + 1 - r).$$

Then, the composite MFCC rule on the graded mesh (5.1) has the following error bound provided that M is large enough:

$$(5.3) \quad Ck^{-r}M^{-N-1+r}\|\widehat{F}\|_{N+1,\beta} + C'k^{-1}\sqrt{N}M^{-N},$$

where C and C' are constants independent of k and M .

Proof. Theorem 2.2 implies that the total error of a composite MFCC rule is the sum of the error of the composite FCC rule and the sum over all the panels of the maximum interpolation errors multiplied by $C'k^{-1}\sqrt{N}$. The error of the composite FCC rule is estimated by [8, Theorem 3.6] as

$$(5.4) \quad Ck^{-r}M^{-N-1+r}\|\widehat{F}\|_{N+1,\beta}.$$

The largest panel of $\Pi_{M,q}$, i.e., the last one, is $[(1 - 1/M)^q, 1]$, and it is rather small only if M is large enough. For the second term of the error, the results in [25] or [2, Section 8.3], and the inequalities (3.5) for small subintervals imply that the maximum interpolation error in each panel is of order $\mathcal{O}(M^{-N-1})$. Note that, since $\beta < 0$, the integration over the first panel $[x_0, x_1]$ is simply approximated by 0 (see [8, equation (3.5)]), hence, the interpolation error in the first panel vanishes. Thus, the second term of the bound for the total error can be written as

$$C'k^{-1}\sqrt{N}(M - 1)M^{-N-1},$$

and this together with (5.4) yields the result. \square

5.2. Numerical experiments. We try to illustrate the error estimate (5.3) by a set of numerical experiments. Consider the following sample integral

$$(5.5) \quad \int_0^1 \frac{x-1}{1+x^2} \exp(ikx^4) dx,$$

with some $k > 0$. Clearly 0 is the only stationary point of the oscillator function, and it is of order 3. So, $\beta = -3/4$. Also, the amplitude and the oscillator functions are so smooth that $\widehat{F} \in C_\beta^{N+1}[0, 1]$ for any $N > 0$. Thus, Theorem 5.1 is applicable.

TABLE 5.1

Application of the composite MFCC rules with the graded meshes $\Pi_{M,q}$ to the integral (5.5) with $k = 1000$: Absolute error with its decay rate.

M	$N = 2$		$N = 4$		$N = 6$		$N = 8$	
	error	rate	error	rate	error	rate	error	rate
32	9.89×10^{-3}	—	3.27×10^{-2}	—	2.95×10^{-1}	—	1.50×10^2	—
64	5.99×10^{-4}	4.0	8.80×10^{-5}	8.5	3.02×10^{-4}	9.9	2.81×10^{-3}	15.7
128	5.35×10^{-5}	3.5	8.25×10^{-6}	3.4	2.78×10^{-6}	6.8	1.06×10^{-6}	11.4
256	4.77×10^{-6}	3.5	1.15×10^{-7}	6.2	1.16×10^{-8}	7.9	1.17×10^{-9}	9.8
512	1.99×10^{-6}	1.3	6.45×10^{-9}	4.2	2.62×10^{-11}	8.8	6.05×10^{-13}	10.9

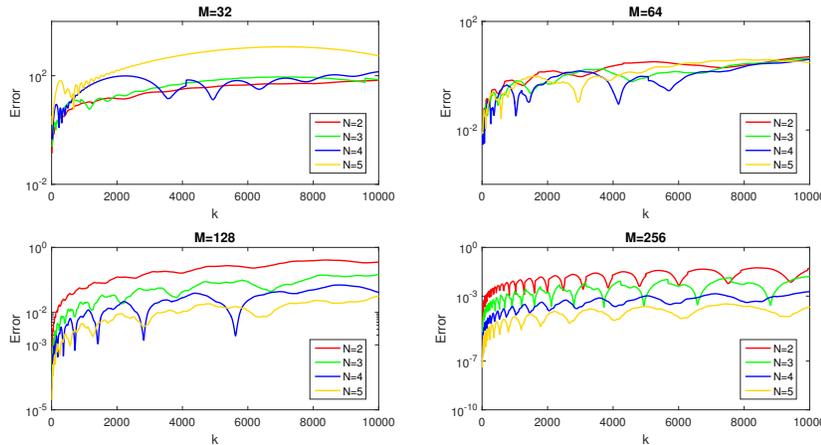


FIG. 5.1. Absolute error scaled by k as a function of k for the composite MFCC rule on the graded mesh (5.1) with $q = \lfloor (N + 1)/(\beta + 1) \rfloor + 1$ when applied to the model integral (5.5).

Experiment 1. If M increases while the other parameters are kept fixed, then Theorem 5.1 implies that the convergence rate is at least of order $\mathcal{O}(M^{-N+1})$. In order to illustrate this result, consider the sample integral (5.5) with $k = 1000$. We employ the composite MFCC rule on the graded mesh (5.1) with $q = \lfloor (N + 1)/(\beta + 1) \rfloor + 1$ for some N and M . By $\lfloor x \rfloor$ we mean the largest integer not greater than x . Then, condition (5.2) is satisfied. The absolute error with its decay rate, as M grows, is given in Table 5.1.

Experiment 2. If $k \rightarrow \infty$ and the other parameters remain fixed, then the error estimate (5.3) implies that the rule converges with the rate of $\mathcal{O}(k^{-1})$. In order to illustrate this rate, we again consider the sample integral (5.5) with k varying in a wide band from 10 to 10^4 . For $M = 32, 64, 128, 256$ and $N = 2, 3, 4, 5$, we employ the composite MFCC rule with the graded mesh (5.1) and $q = \lfloor (N + 1)/(\beta + 1) \rfloor + 1$. In Figure 5.1, for each pair (M, N) , the absolute error scaled by k is displayed as a function of k .

6. Conclusions. We have introduced a general method, which can generate various modifications of the FCC rules and their composite versions. The modified rules can be applied to the oscillatory integral (1.1), while they do not require the inverse of g explicitly. The main tool that allows us to design such modifications is interpolation; all our modifications correspond to a certain interpolation method.

When g has no stationary points in $[a, b]$, two algorithms based on MFCC rules have been introduced. For each one, an error estimate has been given theoretically, and then illustrated

by some numerical experiments. Overall, we found both the algorithms reliable and efficient for any regular integral (1.1) where $g'(x) \neq 0$, $x \in [a, b]$.

When g has a finite number of stationary points in $[a, b]$, it is possible to develop composite rules based on MFCC rules and graded meshes. Similarly to the previous case, an error estimate has been given theoretically and then illustrated by some numerical experiments. Such rules are reliable only for finely graded meshes. Especially, when the grading parameter q in $\Pi_{M,q}$ is rather large, the number M of panels should be large enough. Thus, such rules are not recommended when β is too close to -1 because in this case the panels become too small in a narrow neighborhood of 0. Then, evaluating functions at the interpolation points in such panels may result in large rounding error. This is the same effect as happens for the composite FCC rules (see [8, Section 4.1]).

REFERENCES

- [1] M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, vol. 55 of National Bureau of Standards Applied Mathematics Series, Washington, D.C., 1964.
- [2] K. E. ATKINSON, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, Cambridge, 1997.
- [3] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517.
- [4] O. BRUNO AND D. HOCH, *Numerical differentiation of approximated functions with limited order-of-accuracy deterioration*, SIAM J. Numer. Anal., 50 (2012), pp. 1581–1603.
- [5] S. N. CHANDLER-WILDE, I. G. GRAHAM, S. LANGDON, AND E. A. SPENCE, *Numerical-asymptotic boundary integral methods in high-frequency acoustic scattering*, Acta Numer., 21 (2012), pp. 89–305.
- [6] C. W. CLENSHAW AND A. R. CURTIS, *A method for numerical integration on an automatic computer*, Numer. Math., 2 (1960), pp. 197–205.
- [7] A. DEAÑO AND D. HUYBRECHS, *Complex Gaussian quadrature of oscillatory integrals*, Numer. Math., 112 (2009), pp. 197–219.
- [8] V. DOMÍNGUEZ, I. G. GRAHAM, AND T. KIM, *Filon-Cleenshaw-Curtis rules for highly oscillatory integrals with algebraic singularities and stationary points*, SIAM J. Numer. Anal., 51 (2013), pp. 1542–1566.
- [9] V. DOMÍNGUEZ, I. G. GRAHAM, AND V. P. SMYSHLYAEV, *Stability and error estimates for Filon-Cleenshaw-Curtis rules for highly oscillatory integrals*, IMA J. Numer. Anal., 31 (2011), pp. 1253–1280.
- [10] L. N. G. FILON, *On a quadrature formula for trigonometric integrals*, Proc. Roy. Soc. Edinburgh, 49 (1928), pp. 38–47.
- [11] N. J. HIGHAM, *The numerical stability of barycentric Lagrange interpolation*, IMA J. Numer. Anal., 24 (2004), pp. 547–556.
- [12] D. HUYBRECHS AND S. OLVER, *Superinterpolation in highly oscillatory quadrature*, Found. Comput. Math., 12 (2012), pp. 203–228.
- [13] D. HUYBRECHS AND S. VANDEWALLE, *On the evaluation of highly oscillatory integrals by analytic continuation*, SIAM J. Numer. Anal., 44 (2006), pp. 1026–1048.
- [14] A. ISERLES AND S. P. NØRSETT, *Efficient quadrature of highly oscillatory integrals using derivatives*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 461 (2005), pp. 1383–1399.
- [15] Y. MA AND Y. XU, *Computing highly oscillatory integrals*, Math. Comp., 87 (2018), pp. 309–345.
- [16] H. MAJIDIAN, *Creating stable quadrature rules with preassigned points by interpolation*, Calcolo, 53 (2016), pp. 217–226.
- [17] H. MAJIDIAN, *Stable application of Filon-Cleenshaw-Curtis rules to singular oscillatory integrals by exponential transformations*, BIT Numer. Math., 59 (2019), pp. 155–181.
- [18] S. OLVER, *Moment-free numerical approximation of highly oscillatory integrals with stationary points*, European J. Appl. Math., 18 (2007), pp. 435–447.
- [19] ———, *Numerical Approximation Of Highly Oscillatory Integrals*, PhD. Thesis, University of Cambridge, Cambridge, 2008.
- [20] R. PIESSENS, *Computing integral transforms and solving integral equations using Chebyshev polynomial approximations*, J. Comput. Appl. Math., 121 (2000), pp. 113–124.
- [21] R. PIESSENS AND M. BRANDERS, *The evaluation and application of some modified moments*, BIT Numer. Math., 13 (1973), pp. 443–450.
- [22] ———, *Numerical solution of integral equations of mathematical physics, using Chebyshev polynomials*, J. Comput. Phys., 21 (1976), pp. 178–196.

- [23] ———, *Computation of Fourier transform integrals using Chebyshev series expansions*, *Computing*, 32 (1984), pp. 177–186.
- [24] R. PIESSENS AND F. POLEUNIS, *A numerical method for the integration of oscillatory functions*, *BIT Numer. Math.*, 11 (1971), pp. 317–327.
- [25] J. R. RICE, *On the degree of convergence of nonlinear spline approximation*, in *Approximations with Special Emphasis on Spline Functions* (Proc. Sympos. Univ. of Wisconsin, Madison, Wis., 1969), I. J. Schoenberg, ed., Academic Press, New York, 1969, pp. 349–365.
- [26] H. RUTISHAUSER, *Vorlesungen über numerische Mathematik*, Birkhäuser, Basel, Stuttgart, 1976; English translation, *Lectures on Numerical Mathematics*, Birkhäuser, Boston, 1990.
- [27] I. H. SLOAN AND W. E. SMITH, *Product-integration with the Clenshaw-Curtis and related points. Convergence properties*, *Numer. Math.*, 30 (1978), pp. 415–428.
- [28] I. H. SLOAN AND W. E. SMITH, *Product integration with the Clenshaw-Curtis points: implementation and error estimates*, *Numer. Math.*, 34 (1980), pp. 387–401.
- [29] S. XIANG, *Efficient Filon-type methods for $\int_a^b f(x)e^{i\omega g(x)} dx$* , *Numer. Math.*, 105 (2007), pp. 633–658.