JOURNAL OF NUMERICAL ANALYSIS AND APPROXIMATION THEORY

J. Numer. Anal. Approx. Theory, vol. 53 (2024) no. 1, pp. 132–145, doi.org/10.33993/jnaat531-1380

ictp.acad.ro/jnaat

A COMPARATIVE STUDY OF FILON-TYPE RULES FOR OSCILLATORY INTEGRALS

HASSAN MAJIDIAN[1]

Abstract. Our aim is to answer the following question: "Among the Filon-type methods for computing oscillatory integrals, which one is the most efficient in practice?". We first discuss why we should seek the answer among the family of Filon-Clenshaw-Curtis rules. A theoretical analysis accompanied by a set of numerical experiments reveals that the plain Filon-Clenshaw-Curtis rules reach a given accuracy faster than the (adaptive) extended Filon-Clenshaw-Curtis rules. The comparison is based on the CPU run-time for certain wave numbers (medium and large).

MSC. 65D32, 65Y20.

Keywords. oscillatory integral, Filon-Clenshaw-Curtis rule, extended FCC rule, adaptive FCC+ rule.

1. INTRODUCTION

Consider the oscillatory integral

$$I_k(f) := \int_{-1}^{1} f(x)\mathrm{e}^{\mathrm{i}kx}\,\mathrm{d}x, \tag{1}$$

where f is a given smooth function on $[-1,1]$, and the wave number $k>0$ is rather large. Filon-type methods, in brief, reads as $I_k(p)$, where p is a polynomial approximation of f in $[-1,1]$. In details, the interpolation polynomial p is generally expanded as

$$p(x) = \sum_j \alpha_j q_j(x), \quad x \in [-1,1], \tag{2}$$

where q_j is a polynomial of order j for each j. The scalars α_j, for all j, are computed in the construction phase of p. Thus, the Filon-type method is expanded as

$$I_k(p) = \sum_j \alpha_j m_j, \tag{3}$$

where $m_j := I_k(q_j)$ are the moments to be evaluated.

[1] Department of Multidisciplinary Studies, Faculty of Encyclopedia Studies, Institute for Humanities and Cultural Studies, PO Box 14155-6419, Tehran, Iran, e-mail: h.majidian@ihcs.ac.ir.

Certainly, accuracy of a Filon-type method is proportional to the accuracy of the corresponding polynomial approximation because

$$|I_k(f) - I_k(p)| < 2\|f - p\|_\infty. \tag{4}$$

In practice, what is the best choice of p of a given degree? Even if we restrict our investigation to interpolation, there are a tremendous number of interpolation polynomials approximating a smooth function satisfactory, *e.g.* splines, Hermite-Birkhoff interpolation, and those based on orthogonal polynomials and their roots or extrema. The main question motivating this research is, which interpolation is the best choice for Filon-type methods? Beside the accuracy of the interpolation polynomial p on $[-1,1]$, there are mainly three other aspects that affect the efficiency of the Filon-type method $I_k(p)$ in practice: (i) the cost of construction of p, (ii) the cost of computing the moments, (iii) and the order of interpolation at the endpoints ±1.

In general, Lagrange interpolation at arbitrary points is rather costly. However, in [6], a rapid and stable construction method is proposed based on the fast multipole method (FMM) of [8]. Instead of an arbitrary set of interpolation points, however, one prefers roots or extrema of orthogonal polynomials and utilize FFT for better performance. Among several kinds of orthogonal polynomials, Chebyshev polynomials are often used mainly because their roots and extrema are in the most simple and explicit forms.

For fast computing the moments, one prefers polynomials which satisfy a recurrence relation. Then the moments can be computed recursively with few computational cost. Orthogonal polynomials not only can be constructed efficiently by FFT, but also they satisfy three-term recurrence relations. Assume that (q_j) in (2) satisfies the three-term recurrence relation

$$q_{j+2} = a_j q_{j+1} + b_j q_j, \qquad j = 0, 1, \ldots, \tag{5}$$

where a_j and b_j are known scalars with $|a_j| + |b_j| \neq 0$ for all j. The first two moments can easily be computed analytically. Then, the other ones are computed by (5), recursively:

$$m_{j+2} = a_j m_{j+1} + b_j m_j, \qquad j = 0, 1, \ldots. \tag{6}$$

On the other hand, the asymptotic expansion of the oscillatory integral $I_k(f)$ turns out that the value of $I_k(f)$ is mostly determined by its amplitude, f, and their derivatives up to some order at the endpoints ±1. More precisely, if f is smooth enough in $[-1,1]$, and

$$\left(f^{(m)} - p^{(m)}\right)(\pm 1) = 0, \qquad m = 0, \ldots, s, \tag{7}$$

for some integer $s \geq 0$, then

$$|I_k(f) - I_k(p)| = \mathcal{O}(k^{-s-2}), \quad \text{as } k \to \infty \tag{8}$$

(see, *e.g.*, [11]). Therefore, not only the accuracy of the interpolation, *i.e.* $\|f - p\|_\infty$, is important for the total accuracy of a Filon-type rule $I_k(p)$, but also its order s at the endpoints.

The conditions (7) implies an example of Hermite-Birkhoff interpolation, which construction rarely utilizes fast algorithms like FFT or FMM. Interpolation at the Clenshaw-Curtis points $t_{n,N} := \cos(n\pi/N)$, $j = 0, \dots, n$, have all the above advantages. Because of the distribution of the Clenshaw-Curtis points in $[-1, 1]$, interpolation at them is most accurate and numerically stable. Moreover, thanks to the three-term recurrence relation for the Chebyshev polynomials, it can stably be constructed by FFT at the cost of $\mathcal{O}(n \log n)$ (see [5]). In addition, the computational cost for computing t_n, $n = 0, \dots, N$, is negligible. Finally, having the Lagrange interpolant at the Clenshaw-Curtis, one can efficiently construct the corresponding Hermite-Birkhoff interpolant p that satisfies (7), too (see [14]).

Therefore, without doubt, Filon-Clenshaw-Curtis (FCC) rules and their extensions are the most efficient Filon-type rules for computing the oscillatory integral (1). Extended FCC (FCC+) rules and their adaptive versions can be constructed efficiently by the algorithms of [14] and [13], respectively. To the best of our knowledge, they are the most efficient algorithms for constructing the FCC+ and adaptive FCC+ rules so far.

In comparison with the algorithm of [5] for constructing the FCC rules, the algorithms of [14] and [13] are more complex. This difference in the computational costs may be negligible when the maximum order of derivatives s in the FCC+ rules is rather small. On the other hand, the asymptotic order of an (adaptive) FCC+ rule is better than that of the corresponding FCC rule. Actually, one cannot readily answer which algorithm is more efficient, and this question naturally triggers a comparative study. Indeed, this paper fills a gap in the literature and help in the comparison between Filon-type methods.

In this paper, we carry out a set of numerical experiments to compare efficiency of FCC, FFC+, and adaptive FCC+. Actually, we are interested in the CPU run-time to achieve a specific accuracy. We apply these three classes of rules to a set of oscillatory integrals with different regularity characteristics. Since FCC is less complex than the other rules, one can use more interpolation points for the same computational effort as (adaptive) FCC+.

The rest of the paper is organized as follows. In Section 2, we carry out a set of numerical experiments in order to compare the efficiency of the FCC and the FCC+ rules. In Section 3, the same is done for the FCC and the adaptive FCC+ rules. We also modify Theorem 2.2 of [13] and give a more practical error bound for the adaptive FCC+ rules that is explicit in k and the rule parameters. In Section 4, we account several other advantages of the FCC rules over the (adaptive) FCC+ ones. Finally, we bring some conclusions in Section 5.

2. FCC VS FCC+

In this section, we compare the efficiency of the FCC and FCC+ rules. In contrast to the common ideas in the literature, we see that the FCC+ rules have no advantages over the plain FCC rules in practice. At first, we give a brief review of the FCC+ rules (see [7, 14]).

2.1. A review of the FCC+ rules. The $(N+1)$-point FCC+ rule of order $s \geq 1$ is defined by

$$I_{k,N,s}^{\mathrm{FCC+}}(f) := I_k(p_{N+2s}), \tag{9}$$

where p_{N+2s} is the $(N+2s)^{\mathrm{th}}$ degree Hermite interpolation polynomial satisfying the following interpolation conditions (see [14]):

$$p_{N+2s}(t_{n,N}) = f(t_{n,N}), \quad n = 0, \ldots, N, \tag{10a}$$

$$p_{N+2s}^{(m)}(\pm 1) = f^{(m)}(\pm 1), \quad m = 1, \ldots, s. \tag{10b}$$

Then, the asymptotic error of the rule will be

$$\left| I_k(f) - I_{k,N,s}^{\mathrm{FCC+}}(f) \right| = \mathcal{O}\left(k^{-s-2}\right), \tag{11}$$

see, *e.g.*, [11].

An important advantage of the FCC rules over the FCC+ ones is that there are error estimates for the FCC rules, which are applicable even if f is of confined regularity. Furthermore, they are explicit in both k and N (see [4, 5, 20, 21]). Such error bounds do not exist for the FCC+ rules so far (see [14] for details).

In [7], an efficient algorithm is proposed for constructing the FCC+ rule (9) that can be implemented by $\mathcal{O}\left(N \log N + s^3\right)$ flops. A more efficient algorithm was later proposed in [14]. One advantage of the latter is that its computational cost can be determined precisely. By the algorithm of [14], one needs $\mathcal{O}(s^3 + Ns + s^2)$ flops to extend the $(N+1)$-point FCC rule to the corresponding FCC+ rule of order s in a natural way. Thus, beside its simplicity and naturality, the algorithm of [14] enables one to compare computational costs of the two rules easily. The algorithm for constructing the $(N+1)$-point FCC+ rule of order $s > 1$ leads to two linear systems of order s. However, the FCC+ rules of order 1 can be stated explicitly, and one only needs about $3N + 14$ extra flops to build it from the corresponding $(N+1)$-point FCC rule.

2.2. The race for convergence speed. Error estimate (11) turns out that for a fixed N, the asymptotic decaying rate of the error of $I_{k,N,s}^{\mathrm{FCC+}}(f)$ is higher than that of $I_{k,N}^{\mathrm{FCC}}(f)$, and it grows by s. On the other hand for rather large N, the complexity of the FCC rule, $\mathcal{O}\left(N \log N\right)$, is comparable to that of the FCC+ rule, $\mathcal{O}\left(N \log N + s^3\right)$ because s is usually taken rather small in practice. Thus, some authors conclude that the FCC+ rules are always the winner in the competition with the FCC rules. According to [3],“the generalization of FCC ... enjoys all the advantages of ‘plain’ FCC ... in cost and simplicity, while arbitrarily increasing the asymptotic rate of decay of the error” (p. 66).

In this section, we show that such claims are in doubt in practice. Although, it may be required in some applications to compute $I_k(f)$ for several k and

f, but each time, we face the problem of computing one integral with a fixed parameter k and a fixed amplitude function f. The FCC+ rule $I_{k,N_1,s}^{\text{FCC+}}(f)$ approximates the integral to a specific accuracy for some rather large integers N_1 and s. As mentioned above, the FCC rule $I_{k,N_1}^{\text{FCC}}(f)$ with the same N_1 generally achieves less accuracy. However, we may choose larger integer $N_2 > N_1$ and achieve the same accuracy or even better while the computational cost does not exceed that of $I_{k,N_1,s}^{\text{FCC+}}(f)$. Recall from [14] that the FCC+ rule $I_{k,N,s}^{\text{FCC+}}(f)$ is more complex than its corresponding FCC rule $I_{k,N}^{\text{FCC}}(f)$ by at least $\mathcal{O}(s^3) + N(8s-5) + s(17s+9) + 1$ flops. Now assume that $C_k(N)$ is the exact number of flops for constructing $I_{k,N}^{\text{FCC}}(f)$ by the algorithm proposed in [5]. Thus, in order to compare the simplicity of the two rules $I_{k,N_1,s}^{\text{FCC+}}(f)$ and $I_{k,N_2}^{\text{FCC}}(f)$, we should compare the two numbers $C_k(N_2)$ and

$$C_k(N_1) + \mathcal{O}(s^3) + N_1(8s-5) + s(17s+9) + 1.$$

Even if we compute the number of flops for construction of the two rules exactly, error estimates (specially for the FCC+ rules) are not so sharp that the accuracy of the rules can be estimated a priori for a given N and s. For practical examining the efficiency of an approximation method (or rule), one can consider the run-time that the method consumes in a specific machine to reach a given accuracy.

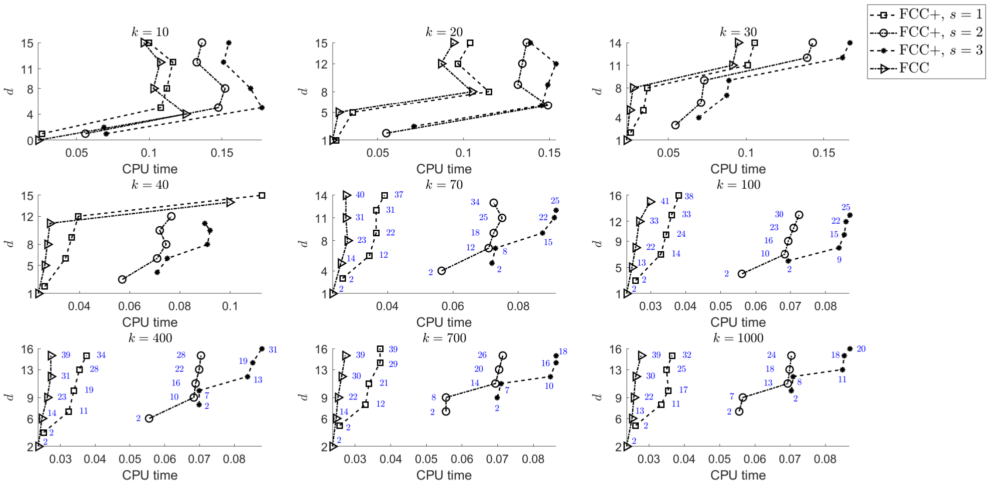

Fig. 2.1. Accuracy vs execution time (in miliseconds) when the FCC rules and the FCC+ rules with $s = 1, 2, 3$ are applied to $I_k(f)$ for $f(x) = (1+x)/(1+x^2)$ and some wave numbers k that range from small to quite large values.

In the following, we apply FCC and FCC+ to a set of oscillatory integrals (1), covering wave numbers from moderate to quite large values. We try to have at least one sample of the amplitude function in each class of $C^\infty([-1,1])$, $C^m([-1,1])$ for some integer $m > 0$, elementary functions, and non-elementary functions. In each method, we choose N as the minimum number of interpolation points to achieve a given accuracy. As discussed above, this number for FCC is generally larger than that of FCC+.

Experiment 1. Let $d := -\log_{10} E$, where E is the relative error of an approximation. Indeed, d is roughly the number of correct significant digits in the approximation. In Fig. 2.1, we plotted d versus the machine run-time (in milliseconds) for the FCC and FCC+ rules (9) of orders $s = 1, 2, 3$ when applied to $I_k(f)$ with $f(x) = (1 + x)/(1 + x^2)$ and some wave numbers $k > 0$ ranging from small to quite large values. Note that each time an algorithm is run by a specific machine, the run-time may vary a little mainly because of invisible services running in the background on the operating system (OS). For that reason, in all the numerical experiments in the sequel concerning run-times, we run the corresponding MATLAB algorithm ten times and consider the minimum of the run-times. For $k \geq 70$, for which the plots have space enough, we label each marker by the corresponding N. All the MATLAB codes have been optimized in order to use less memory and take less time to execute.

Although, the derivatives of f have been precalculated manually to the advantage of FCC+, our numerical experiments turn out that FCC are still more efficient than FCC+, especially for moderate and larger k. Also, among the FCC+ rules, those with smaller s are more efficient. This observation is against some authors' claims that the FCC+ rules are 'the winner' just because the asymptotic decaying rates of their error are higher than those of the FCC rules. As it is seen, for a specific N, accuracy of the FCC+ rules is higher than that of the FCC rule, and this accuracy enhances as s grows. For example, in the last plot corresponding to $k = 1000$, the 31-point FCC rule, the 26-point FCC+ rule of order 1, the 19-point FCC+ rule of order 2, and the 12-point FCC+ rule of order 3 reach approximately 13 correct significant digits. However, the FCC rules ultimately reach a given accuracy faster than all the FCC+ rules.

Experiment 2. The amplitude function f in the previous experiment is infinitely differentiable. Here, we consider $f(x) = x^{3/2}$, $x \in [0, 1]$, for which only the first derivative exists on $[0, 1]$. Thus, the parameter s in the FCC+ rules cannot exceed 1. In Fig. 2.2, we have plotted the accuracy d versus the machine run-time (in milliseconds) for the FCC rules and the FCC+ rules of order 1. Because in this example the degree of regularity is low, the rules reach a given accuracy only for larger N.

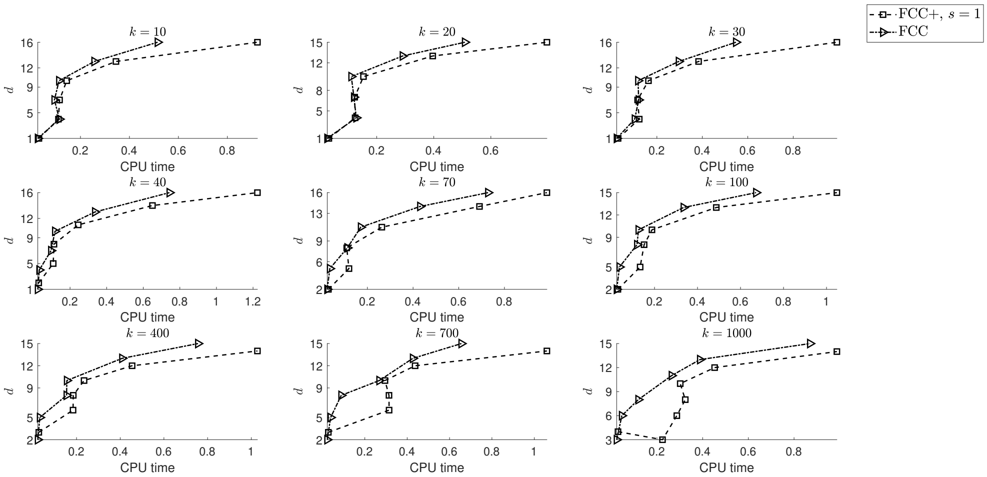

Fig. 2.2. Accuracy vs execution time (in miliseconds) when the FCC rules and the FCC+ rules with $s = 1$ are applied to $I_k(f)$ for $f(x) = x^{3/2}$, $x \in [0, 1]$, and some wave numbers k that range from small to quite large values.

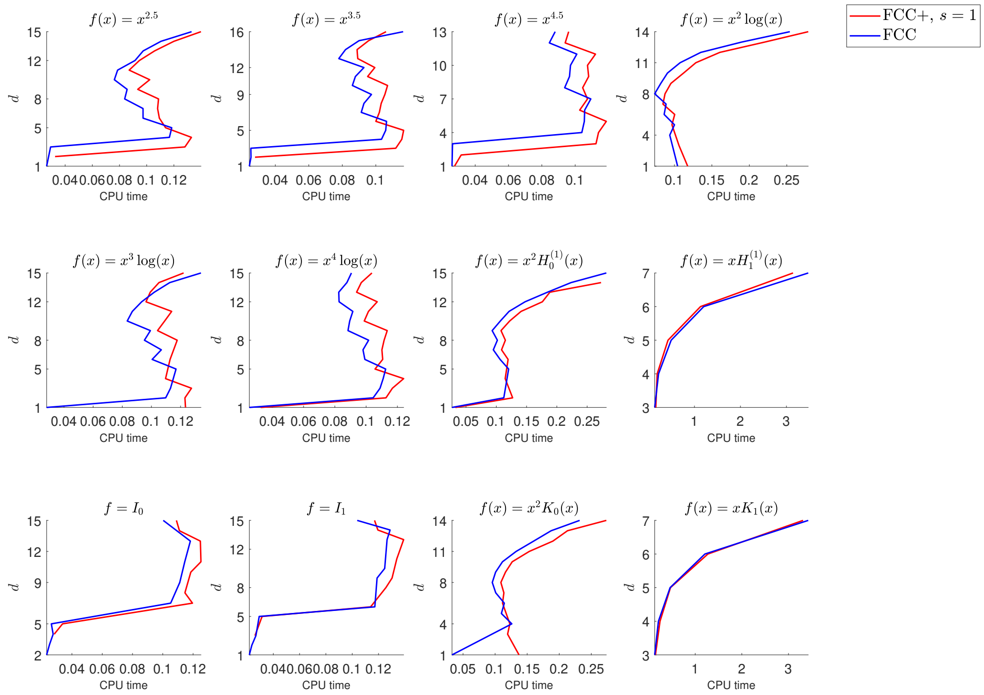

Fig. 2.3. Accuracy vs execution time (in miliseconds) when the FCC rules and the FCC+ rules with $s = 1$ are applied to $I_{10}(f)$ for functions f of Experiment 3.

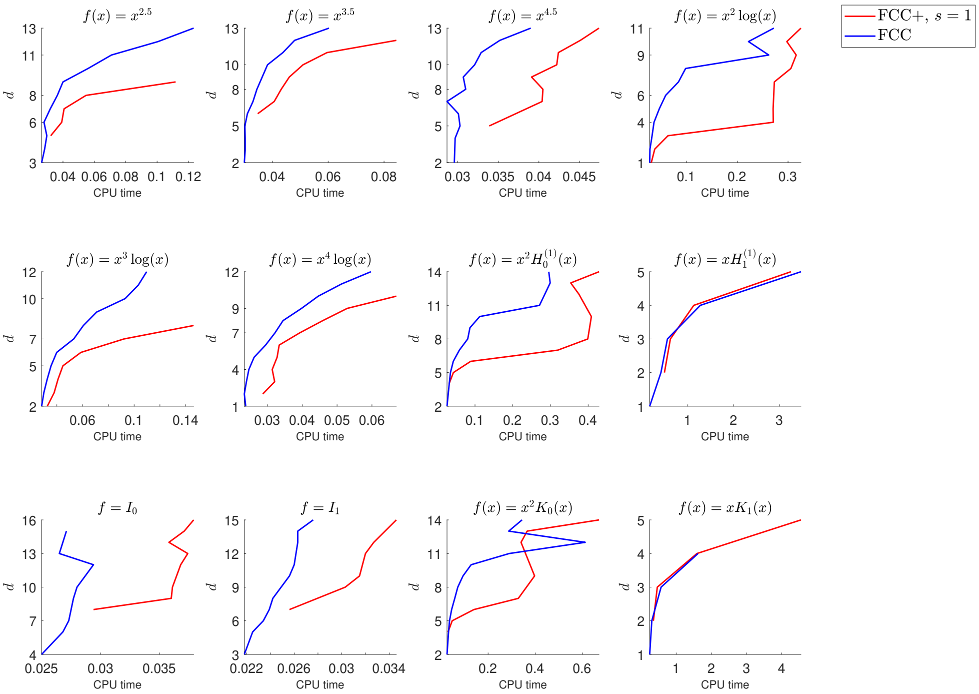

Fig. 2.4. Accuracy vs execution time (in miliseconds) when the FCC rules and the FCC+ rules with $s = 1$ are applied to $I_{1000}(f)$ for functions f of Experiment 3.

Experiment 3. In order to see if the above experimental results hold for other amplitude functions f, here we choose a set of elementary and non-elementary functions of various types on $[0, 1]$:

$$\begin{array}{cl} x^\alpha, & \alpha = 2.5, 3.5, 4.5, \\ x^n \log(x), & n = 2, 3, 4, \\ x^2 H_0^{(1)}(x), & xH_1^{(1)}(x), \\ I_\nu(x), & \nu = 0, 1, \\ x^2 K_0(x), & xK_1(x). \end{array}$$

Here, $H_\nu^{(1)}$, I_ν, and K_ν are the Hankel function of the first kind, the modified Bessel function of the first kind, and the modified Bessel function of the second kind of order ν, respectively. $I_\nu(x)$ is smooth on $[0, 1]$ while each of the other functions has one and only one singularity at zero. The singularity is bounded and of algebraic or logarithmic type.

In Fig. 2.3 and Fig. 2.4, we have compared the efficiency of the FCC rules and FCC+ rules of order one for computing $I_{10}(f)$ and $I_{1000}(f)$, respectively. As it is seen, the FCC rules are most often the winner, especially for the larger wave number.

3. FCC VS ADAPTIVE FCC+

In this section, we compare the efficiency in practice of the FCC and the adaptive FCC+ rules of [13]. An adaptive FCC+ rule and its corresponding FCC+ rule always have the same asymptotic order while the former is derivative-free and less complex. We will see, however, that in the race to achieve a given accuracy, the adaptive FCC+ rules are still overtaken by the corresponding FCC rules. At first, we give a brief review of [13], where the concept of adaptive FCC+ rules has been introduced for the first time.

3.1. A review of the adaptive FCC+ rules. The $(N+2s+1)$-point adaptive FCC+ rule of order s is defined by

$$I^{\text{AFCC+}}_{k,N,s}(f) := I_k(p_{N+2s}), \tag{12}$$

where p_{N+2s} is the Lagrange interpolant of f at $t_{n,N}$, for $n = 0, \dots, N$, and $2s$ extra points $t'_m(k)$, $t''_m(k)$, $m = 1, \dots, s$, that should be near the endpoints ±1 and are defined as follows:

$$t'_m(k) := \cos(\eta_m(k)) \quad \text{with} \quad \eta_m(k) := \sqrt{\frac{m}{k+1}}, \quad t''_m(k) := -t'_m(k). \tag{13}$$

As mentioned above, an adaptive FCC+ rule preserves the asymptotic order of the corresponding FCC+ rule. More precisely, we have the following result proved in [13]:

Theorem 1. *Assume that $f \in C^{N+2s+1}([-1,1])$. Then,*

$$\left| I_k(f) - I^{\text{AFCC+}}_{k,N,s}(f) \right| = \mathcal{O}\left(k^{-s-2}\right), \tag{14}$$

as $k \to \infty$.

In contrast to the FCC+ rules, there are error bounds for the adaptive FCC+ rules that are explicit in all the three parameters k, N, and s.

Corollary 2. *Assume that $f \in C^{N+2s+1}([-1,1])$, and $k+1 \geq 2s$. Then,*

$$\left| I^{\text{AFCC+}}_{k,N,s}(f) - I_k(f) \right| \leq k^{-2}2^{-N} \tfrac{17}{(N+2s-4)!} \left\| f^{(N+2s+1)} \right\|_\infty, \tag{15}$$

for all $N > 1$.

Proof. For $N > 1$,

$$\begin{aligned} \tfrac{2Ns!\,k^{-s}}{(N+2s+1)!} + \tfrac{\gamma_{N,s}}{(N+2s-1)!} &< \tfrac{2}{(N+2s-4)!}\left(N^{-4}k^{-s}s! + 2\right) \\ &< \tfrac{2(1/8+2)}{(N+2s-4)!}, \end{aligned}$$

where $\gamma_{N,s} := s(4N+2s-1) + N(2N^2+1)/3$. Now, the result is followed by Theorem 2.2 of [13]. ■

In [13], it was shown numerically that, adaptive FCC+ is always more efficient than FCC+. However, our numerical experiments in the next section show that it is still overtaken by FCC is the race to achieve a given accuracy.

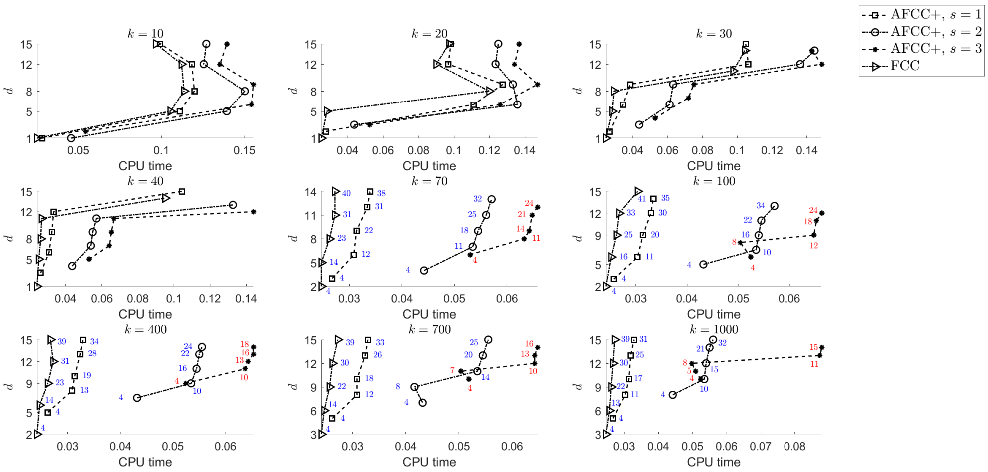

Fig. 3.5. Accuracy vs execution time (in milliseconds) when the FCC rules and the adaptive FCC+ rules with $s = 1, 2, 3$ are applied to $I_k(f)$ for $f(x) = (1 + x)/(1 + x^2)$ and some wave numbers k that range from small to quite large values.

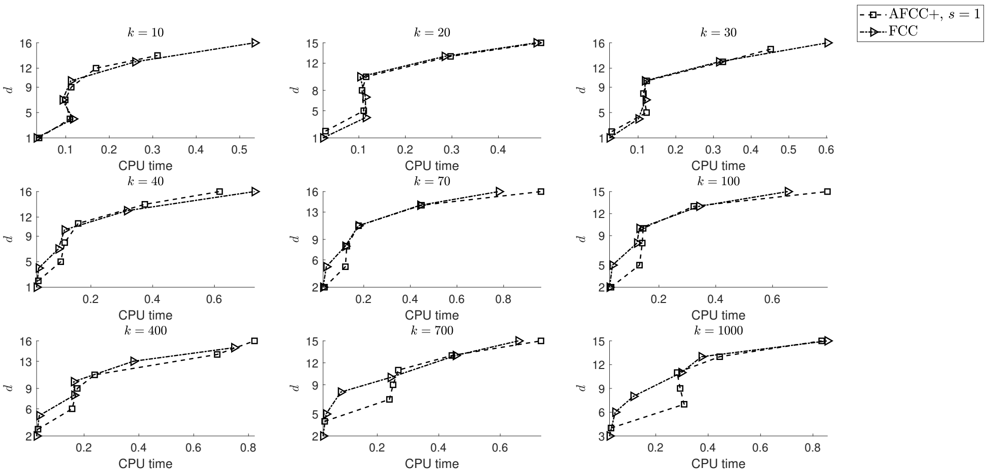

Fig. 3.6. Accuracy vs execution time (in milliseconds) when the FCC rules and the adaptive FCC+ rules with $s = 1$ are applied to $I_k(f)$ for $f(x) = x^{3/2}$, $x \in [0, 1]$, and some wave numbers k that range from small to quite large values.

3.2. The race for convergence speed. In this section, we repeat the numerical experiments of Section 2.2, now by the adaptive FCC+ instead of the FCC+ rules. The same discussion is applicable here, too. In Fig. 3.5–Fig. 3.8, we compare convergence rates of the adaptive FCC+ rules and the corresponding FCC rules. The results are the same as in Fig. 2.1–Fig. 2.4, and FCC is always the winner.

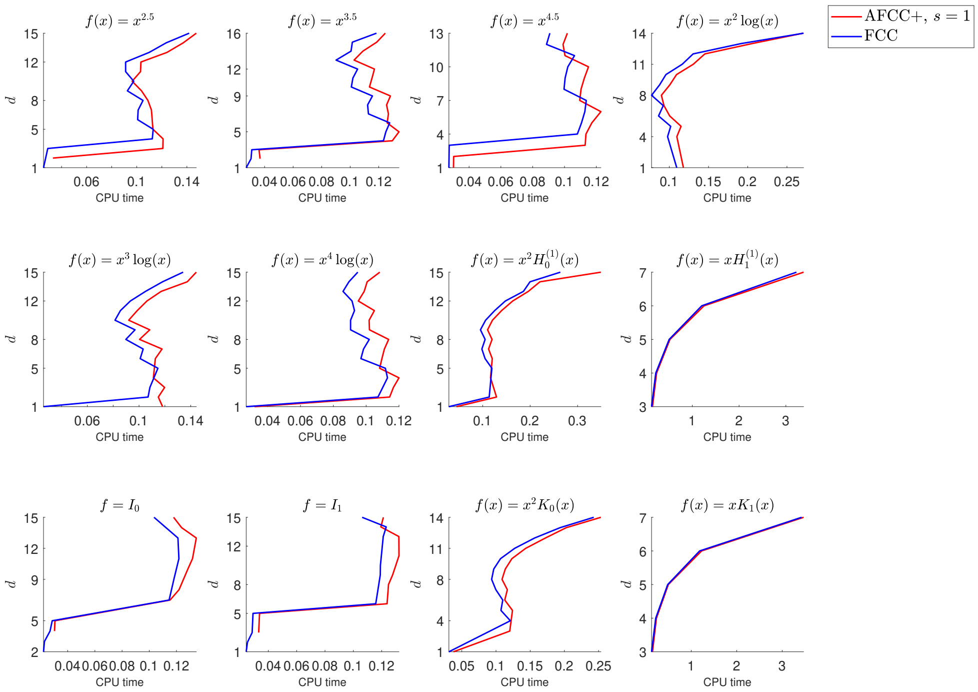

Fig. 3.7. Accuracy vs execution time (in miliseconds) when the FCC rules and the adaptive FCC+ rules with $s = 1$ are applied to $I_{10}(f)$ for functions f of Experiment 3.

4. OTHER ADVANTAGES OF FCC RULES

The FCC rules have further advantages over the FCC+ and adaptive FCC+ rules. In this section, we account some of them.

(1) Composite FCC rules can readily be developed [4] while the composite versions of the (adaptive) FCC+ rules require many data within the integration interval. In the case of FCC+ rules, one should compute derivatives of the amplitude function at some (or many) internal points in addition to the endpoints. This can increase the computational cost most largely.

(2) One can apply composite FCC rules with suitably graded meshes and treat oscillatory integrals with algebraic or logarithmic endpoint singularities [4]. Trying to apply this idea to the FCC+ rules, one should compute derivatives of the amplitude function at some (or many) points clustered near the singular endpoint. In the case of adaptive FCC+ rules, we will have a bunch of points clustered near the endpoints of each subinterval. Thus, in the vicinity of the singular endpoint, the density of the points may become so high that rounding error pollutes the total accuracy.

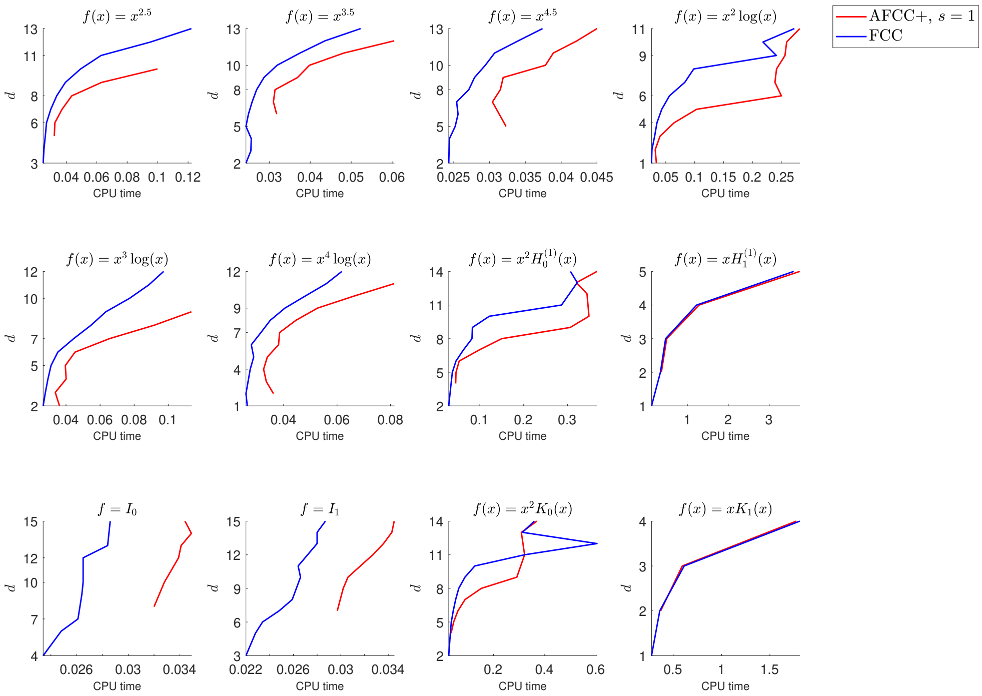

Fig. 3.8. Accuracy vs execution time (in miliseconds) when the FCC rules and the adaptive FCC+ rules with $s = 1$ are applied to $I_{1000}(f)$ for functions f of Experiment 3.

(3) When estimating the computational cost of the FCC+ rules in [14], it is assumed that the derivatives of the amplitude function have been precalculated. This is also the case in the numerical experiments of Section 2.2. However, in many situations it is impossible or awkward to differentiate manually [10]. It is well-known that computing derivatives (especially of high-order) by finite differences is subject to cancellation. For that reason, many special numerical differentiation methods has been developed, *e.g.* automatic differentiation [9, 18, 19], complex and multicomplex step differentiation [12, 16, 17], differentiation rules base on Cauchy integrals [1, 2], etc. Thus, computing the derivatives is not necessarily a trivial task and imposes an extra computational cost.

(4) One can study the stability of the FCC rules [15], but such study can be sophisticated for (adaptive) FCC+ rules.

5. CONCLUSIONS

In this work, we carried out a benchmark of numerical experiments and observed that the (adaptive) FCC+ rules are most often overtaken by the plain FCC rules in the speed race to achieve a given accuracy. Availability of error estimates when the amplitude function is of confined regularity, capability to develop efficient composite rules, availability of a stability analysis, and being derivative-free are other features that make the FCC rules the most-efficient decisive-choice ones among other Filon-type methods for oscillatory integrals (1).

REFERENCES

[1] F. Bornemann, *Accuracy and stability of computing high-order derivatives of analytic functions by Cauchy integrals*, Found. Comput. Math., 11 (2011), pp. 1–63, https://doi.org/10.1007/s10208-010-9075-z. ↗.

[2] F. Bornemann and G. Wechslberger, *Optimal contours for high-order derivatives*, IMA J. Numer. Anal., 33 (2013), pp. 403–412, https://doi.org/10.1093/imanum/drs030. ↗.

[3] A. Deaño, D. Huybrechs, and A. Iserles, *Computing Highly Oscillatory Integrals*, SIAM, 2017, https://doi.org/10.1137/1.9781611975123. ↗.

[4] V. Domínguez, I. Graham, and T. Kim, *Filon-Clenshaw-Curtis rules for highly oscillatory integrals with algebraic singularities and stationary points*, SIAM J. Numer. Anal., 51 (2013), pp. 1542–1566, https://doi.org/10.1137/120884146. ↗.

[5] V. Domínguez, I. Graham, and V. Smyshlyaev, *Stability and error estimates for Filon-Clenshaw-Curtis rules for highly oscillatory integrals*, IMA J. Numer. Anal., 31 (2011), pp. 1253–1280, https://doi.org/10.1093/imanum/drq036. ↗.

[6] A. Dutt, M. Gu, and V. Rokhlin, *Fast algorithms for polynomial interpolation, integration, and differentiation*, SIAM J. Numer. Anal., 33 (1996), pp. 1689–1711, https://doi.org/10.1137/0733082. ↗.

[7] J. Gao and A. Iserles, *A generalization of Filon-Clenshaw-Curtis quadrature for highly oscillatory integrals*, BIT Numer. Math., 57 (2017), pp. 943–961, https://doi.org/10.1007/s10543-017-0682-9. ↗.

[8] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348, https://doi.org/10.1016/0021-9991(87)90140-9. ↗.

[9] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, 2008, https://doi.org/10.5555/1455489. ↗.

[10] A. Iserles and S. Nørsett, *On quadrature methods for highly oscillatory integrals and their implementation*, BIT Numer. Math., 44 (2004), pp. 755–772, https://doi.org/10.1007/s10543-004-5243-3. ↗.

[11] A. Iserles and S. Nørsett, *Efficient quadrature of highly oscillatory integrals using derivatives*, Proc. R. Soc. Lond. A, 461 (2005), pp. 1383–1399, https://doi.org/10.1098/rspa.2004.1401. ↗.

[12] G. Lantoine, R. Russell, and T. Dargent, *Using multicomplex variables for automatic computation of high-order derivatives*, ACM Trans. Math. Softw., 38 (2012), pp. 1–21, https://doi.org/10.1145/2168773.2168774. ↗.

[13] H. Majidian, *Adaptive FCC+ rules for oscillatory integrals*, J. Comput. Appl. Math., 424 (2023), p. 115012, https://doi.org/10.1016/j.cam.2022.115012. ↗.

[14] H. Majidian, *Efficient construction of FCC+ rules*, J. Comput. Appl. Math., 417 (2023), p. 114592, https://doi.org/10.1016/j.cam.2022.114592. ↗.

[15] H. Majidian, M. Firouzi, and A. Alipanah, *On the stability of Filon-Clenshaw-Curtis rules*, Bull. Iran. Math. Soc., 48 (2022), pp. 2943–2964, https://doi.org/10.1007/s41980-022-00681-4. ↗.

[16] J. Martins, P. Sturdza, and J. Alonso, *The complex-step derivative approximation*, ACM Trans. Math. Softw., 29 (2003), pp. 245–262, https://doi.org/10.1145/838250.838251. ↗.

[17] H. Millwater and S. Shirinkam, *Multicomplex Taylor series expansion for computing high order derivatives*, Int. J. Appl. Math., 27 (2014), pp. 311–334, https://doi.org/10.12732/ijam.v27i4.2. .

[18] R. Neidinger, *Introduction to automatic differentiation and MATLAB object-oriented programming*, SIAM Rev., 52 (2010), pp. 545–563, https://doi.org/10.1137/080743627. .

[19] M. Patterson, M. Weinstein, and A. Rao, *An efficient overloaded method for computing derivatives of mathematical functions in MATLAB*, ACM Trans. Math. Softw., 39 (2013), pp. 1–36, https://doi.org/10.1145/2450153.2450155. .

[20] S. Xiang, X. Chen, and H. Wang, *Error bounds for approximation in Chebyshev points*, Numer. Math., 116 (2010), pp. 463–491, https://doi.org/10.1007/s00211-010-0309-4. .

[21] S. Xiang, G. He, and Y. Cho, *On error bounds of Filon-Clenshaw-Curtis quadrature for highly oscillatory integrals*, Adv. Comput. Math., 41 (2015), pp. 573–597, https://doi.org/10.1007/s10444-014-9377-9. .

Received by the editors: December 18, 2023; accepted: February 22, 2024; published online: March 6, 2024.