Fast Construction of Fejér and Clenshaw-Curtis rules for general weight functions

Alvise Sommariva^a

^a*Department of Mathematics, University of Padova (Italy)*

Abstract

The main purpose of this paper is to compute the weights of Clenshaw-Curtis and Fejér type quadrature rules via DCT and DST, for general weight functions w . The approach is different from that used by Waldvogel in [25], where the author considered the computation of these sets only for the Legendre weight $w \equiv 1$ using DFT arguments.

Key words: Quadrature formula; Fejér rules; Clenshaw-Curtis rules; Discrete Cosine Transform; Discrete Sine Transform; Gegenbauer weight function; Trigonometric quadrature.

1 Introduction

In this paper we propose algorithms for the fast computation of Clenshaw-Curtis rules as well as of the Fejér formulas of type I and II, so that their nodes $\{x_k\}_{k=1,\dots,n}$ and the relative weights $\{w_k\}$ provide a quadrature rule of interpolatory type for a general weight function w

$$\int_{-1}^1 f(x)w(x)dx \approx \sum_{k=1}^n w_k f(x_k), \quad (1)$$

* Work supported by the “ex-60%” funds and by the project “Interpolation and Extrapolation: new algorithms and applications” of the University of Padova, and by the INdAM GNCS.

* Corresponding author. Address: via Trieste 63, 35121 Padova (Italy).
Email address: `alvise@math.unipd.it` (Alvise Sommariva).

i.e. its algebraic degree of precision is equal to $n - 1$, that means

$$\int_{-1}^1 p_s(x)w(x)dx = \sum_{k=1}^n w_k p_s(x_k),$$

with p_s belonging to the set $\mathbb{P}_{n-1}$ of algebraic polynomials of degree $n - 1$.

In particular the nodes $\{x_k\}$ are of Chebyshev-type while the weights $\{w_k\}$ are computed by sums of trigonometric functions.

The history of these formulas is rather long. Fejér introduced two rules [9], [10] in 1933. The nodes of the first one are the roots of the Chebyshev polynomial of the first kind T_n , while for the second one they are the zeros of the Chebyshev polynomial of second kind U_n . Furthermore, for the case $w \equiv 1$, Fejér established a closed formula for the weights as well as their positiveness.

Later, in 1960, Clenshaw and Curtis [3] introduced a new approach based on the nodes of Fejér rule of second type plus the interval extrema $-1, +1$. This interpolatory rule has many features, including that it coincides with the integration of a certain finite Chebyshev expansion of the integrand f in (1). From this point of view, the evaluation of the integral by Clenshaw-Curtis quadrature can be performed by DCT as shown by Gentleman [12], [13]. Many results are known for quite general weights, for instance, if $\int_{-1}^1 |w(x)|^p dx < +\infty$, for some $p > 1$, convergence properties [21] and error estimates [22] of this formula have been established. Lately there has been a renewed interest in Clenshaw-Curtis quadrature rule after the publication of [24], in which Trefethen has shown not only the fast computation of the integrals but also its surprising performance, often comparable to Gaussian rules.

In [25], Waldvogel described how to compute these formulas, using a clever approach via FFT and iFFT. Here we show how for general weight functions w they are connected to DCT and DST, computing their nodes and weights once the weighted modified moments of Chebyshev polynomials of the first or second kind (w.r.t. w) are available. This allows to generalise the techniques of [25] where only the Legendre weight $w(x) \equiv 1$ is considered. After having implemented these algorithms in Matlab, we show their performance for $n = 2^k$, $k = 1, \dots, 20$ for Gegenbauer weight functions and for a certain w having interest in trigonometric quadrature.

2 Fejér rules of type I

The n -points Fejér rules of type I in the reference interval $(-1, 1)$ has nodes

$$x_k = \cos(\theta_k), \quad \theta_k = \frac{(2k-1)\pi}{2n}, \quad k = 1, \dots, n. \quad (2)$$

i.e. the Chebyshev nodes.

By a simple modification of classical arguments (see e.g. [8, p.85]), if $T_m(x) = \cos(m \arccos(x))$ is the degree m Chebyshev polynomial, we have for a general weight function $w : (-1, 1) \rightarrow \mathbb{R}$

$$w_k = \frac{1}{n} \left(\gamma_0 + 2 \sum_{m=1}^{n-1} \gamma_m T_m(x_k) \right), \quad \gamma_m = \int_{-1}^1 T_m(x) w(x) dx. \quad (3)$$

Thus

$$w_k = \frac{1}{n} \left(\gamma_0 + 2 \sum_{m=1}^{n-1} \gamma_m \cos \left(m \frac{(2k-1)\pi}{2n} \right) \right), \quad k = 1, \dots, n. \quad (4)$$

Now we carefully show that the quantity on the r.h.s. of (4) can be computed by a Discrete Cosine Transform of type III (shortened as DCT III), that maps the values $y_0, \dots, y_{n-1}$ into $\hat{y}_0^{(III)}, \dots, \hat{y}_{n-1}^{(III)}$ as

$$\hat{y}_j^{(III)} = \frac{1}{2} y_0 + \sum_{m=1}^{n-1} y_m \cos \left(\frac{\pi m}{n} \left(j + \frac{1}{2} \right) \right), \quad j = 0, \dots, n-1. \quad (5)$$

Setting $j = k-1$, $w_j^* \equiv w_{j+1}$ and being

$$\hat{\gamma}_j^{(III)} = \frac{\gamma_0}{2} + \sum_{m=1}^{n-1} \gamma_m \cos \left(\frac{\pi m}{n} \left(j + \frac{1}{2} \right) \right) \quad (6)$$

from (6) we have

$$w_j^* = w_{j+1} = \frac{2}{n} \left(\frac{\gamma_0}{2} + \sum_{m=1}^{n-1} \gamma_m \cos \left(m \frac{(2j+1)\pi}{2n} \right) \right) = \frac{2}{n} \hat{\gamma}_j^{(III)}, \quad j = 0, \dots, n-1. \quad (7)$$

hence all the weights $w_1, \dots, w_n$ are readily available by a DCT III.

If the weight function w is symmetric then, as one can easily check, the odd weighted moments γ_{2k+1} are null and $w_k = w_{n-k+1}$, hence putting $N = \lfloor (n-1)/2 \rfloor + 1$, we only need to determine

$$\begin{aligned}
w_j^* = w_{j+1} &= \frac{2}{n} \left(\frac{\gamma_0}{2} + \sum_{s=1}^{\lfloor (n-1)/2 \rfloor} \gamma_{2s} \cos \left(\frac{s(2j+1)\pi}{n} \right) \right) \\
&= \frac{2}{n} \left(\frac{\gamma_0}{2} + \sum_{s=1}^{N-1} \gamma_{2s} \cos \left(\frac{s(2j+1)\pi}{n} \right) \right)
\end{aligned} \tag{8}$$

for $j = 0, \dots, N-1$.

We consider first the case in which n is even that implies $N = 1 + \lfloor (n-1)/2 \rfloor = n/2$. Then from (8) we have

$$w_j^* = w_{j+1} = \frac{2}{n} \left(\frac{\gamma_0}{2} + \sum_{s=1}^{N-1} \gamma_{2s} \cos \left(\frac{s(2j+1)\pi}{2N} \right) \right) \tag{9}$$

for $j = 0, \dots, N-1$. Again, the last term can be computed by mapping the even moments $\gamma_0, \dots, \gamma_{2s}, \dots, \gamma_{2N-2}$ into $\hat{\gamma}_0^{(III)}, \dots, \hat{\gamma}_{2s}^{(III)}, \dots, \hat{\gamma}_{2N-2}^{(III)}$ via DCT III, i.e

$$\hat{\gamma}_{2j}^{(III)} = \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_{2m} \cos \left(\frac{\pi m}{N} \left(j + \frac{1}{2} \right) \right), \quad j = 0, \dots, N-1 \tag{10}$$

providing

$$w_j^* = w_{j+1} = \frac{2}{n} \hat{\gamma}_{2j}^{(III)} = \frac{1}{N} \hat{\gamma}_{2j}^{(III)}, \quad j = 0, \dots, N-1. \tag{11}$$

If n is odd, then $N = 1 + \lfloor (n-1)/2 \rfloor = (n+1)/2$ and (8) gives

$$\begin{aligned}
w_j^* = w_{j+1} &= \frac{2}{2N-1} \left(\frac{\gamma_0}{2} + \sum_{s=1}^{N-1} \gamma_{2s} \cos \left(\frac{s(2j+1)\pi}{2N-1} \right) \right) \\
&= \frac{2}{2N-1} \left(\frac{\gamma_0}{2} + \sum_{s=1}^{N-1} \gamma_{2s} \cos \left(s \frac{(j + \frac{1}{2})\pi}{N - \frac{1}{2}} \right) \right)
\end{aligned} \tag{12}$$

for $j = 0, \dots, N-1$. Remembering that the Discrete Cosine Transform of type VII maps the values $y_0, \dots, y_{n-1}$ into $\hat{y}_0^{(VII)}, \dots, \hat{y}_{n-1}^{(VII)}$ as (check)

$$\hat{y}_j^{(VII)} = \frac{y_0}{2} + \sum_{s=1}^{n-1} y_s \cos \left(\frac{s(j + \frac{1}{2})\pi}{n - \frac{1}{2}} \right) \tag{13}$$

if the even moments $\gamma_0, \dots, \gamma_{2k}, \dots, \gamma_{2n-2}$ are mapped by DCT VII into $\hat{\gamma}_0^{(VII)}, \dots, \hat{\gamma}_{2k}^{(VII)}, \dots, \hat{\gamma}_{2n-2}^{(VII)}$, i.e.

$$\hat{\gamma}_{2k}^{(VII)} = \frac{\gamma_0}{2} + \sum_{s=1}^{N-1} \gamma_{2s} \cos \left(\frac{s(k + \frac{1}{2})\pi}{N - \frac{1}{2}} \right)$$

we finally have from (12)

$$w_j^* = w_{j+1} = \frac{2}{n} \hat{\gamma}_{2j}^{(VII)} = \frac{2}{2N-1} \hat{\gamma}_{2j}^{(VII)} \quad (14)$$

for $j = 0, \dots, N-1$.

3 Fejér rules of type II

The n -point Fejér rule of type II has nodes

$$x_k = \cos(\theta_k), \quad \theta_k = \frac{k\pi}{n+1}, \quad k = 1, \dots, n,$$

i.e. the zeros of the Chebyshev polynomial of second kind of degree n

$$U_n(x) = \frac{\sin((n+1)\arccos(x))}{\sin(\arccos(x))}.$$

Let $w : (-1, 1) \rightarrow \mathbb{R}$ be a weight function. Our purpose is to determine the weights $\{w_k\}$ of the n -point Fejér rule of type II for w , following the outline of [8] where the proof is just sketched for the Legendre weight function $w \equiv 1$.

From the Darboux-Christoffel formula (see [8, p.31, p.35-36]) we have

$$U_n(x) = \frac{2(x_k - x)}{U_{n+1}(x_k)} \sum_{j=0}^n U_s(x_k) U_s(x).$$

Let $\lambda_s = \int_{-1}^1 U_s(x) w(x) dx$ be the weighted moments of the Chebyshev polynomials of second kind (w.r.t. w). Then

$$\begin{aligned} w_k &= \frac{1}{U_n'(x_k)} \int_{-1}^1 \frac{U_n(x)}{x - x_k} w(x) dx \\ &= \frac{-2}{U_n'(x_k) U_{n+1}(x_k)} \sum_{s=0}^n U_s(x_k) \int_{-1}^1 U_s(x) w(x) dx \\ &= \frac{-2}{U_n'(x_k) U_{n+1}(x_k)} \sum_{s=0}^n U_s(x_k) \lambda_s. \end{aligned} \quad (15)$$

Since $U_{n+1}(x_k) = T_{n+1}(x_k)$, $U_n(x_k) = 0$, $T_{n+1}^2(x_k) = 1$, and

$$U_n'(x) = \frac{(n+1)T_{n+1}(x) - xU_n(x)}{x^2 - 1},$$

we immediately have

$$\frac{1}{U'_n(x_k)U_{n+1}(x_k)} = \frac{x_k^2 - 1}{(n+1)T_{n+1}^2(x_k)} = \frac{x_k^2 - 1}{n+1}. \quad (16)$$

Consequently, for $k = 1, \dots, n$,

$$\begin{aligned} w_k &= \frac{2(1 - x_k^2)}{(n+1)} \sum_{s=0}^n U_s(x_k) \lambda_s = \frac{2(1 - \cos^2(\theta_k))}{n+1} \sum_{s=0}^n \lambda_s \frac{\sin((s+1)\theta_k)}{\sin(\theta_k)} \\ &= \frac{2 \sin(\theta_k)}{n+1} \sum_{s=0}^n \lambda_s \sin((s+1)\theta_k) = \frac{2 \sin(\theta_k)}{n+1} \sum_{s=0}^{n-1} \lambda_s \sin((s+1)\theta_k), \end{aligned} \quad (17)$$

where in the last equality we have used the fact that $\sin((n+1)\theta_k) = 0$.

Now we observe that putting $j = k - 1$, $w_j^* = w_{j+1}$, we have from (17)

$$\begin{aligned} w_j^* = w_{j+1} &= \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{n-1} \sin((s+1)\theta_{j+1}) \lambda_s \\ &= \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{n-1} \lambda_s \sin\left(\frac{(s+1)(j+1)\pi}{n+1}\right), \quad j = 0, \dots, n-1. \end{aligned} \quad (18)$$

The Discrete Sine Transform I (shortened as DST I), maps $y_0, \dots, y_{n-1}$ into $\tilde{y}_0^{(I)}, \dots, \tilde{y}_{n-1}^{(I)}$ defined as

$$\tilde{y}_j^{(I)} = \sum_{s=0}^{n-1} y_s \sin\left(\frac{(s+1)(j+1)\pi}{n+1}\right), \quad j = 0, \dots, n-1. \quad (19)$$

Consequently, the DST I maps $\lambda_0, \dots, \lambda_{n-1}$ into $\tilde{\lambda}_0^{(I)}, \dots, \tilde{\lambda}_{n-1}^{(I)}$ with

$$\tilde{\lambda}_j^{(I)} = \sum_{s=0}^{n-1} \lambda_s \sin\left(\frac{(s+1)(j+1)\pi}{n+1}\right), \quad j = 0, \dots, n-1. \quad (20)$$

and comparing (18) with (20) we get

$$w_j^* = w_{j+1} = \frac{2 \sin(\theta_{j+1})}{n+1} \tilde{\lambda}_j^{(I)}, \quad j = 0, \dots, n-1. \quad (21)$$

so determining the weights w_k of the Fejér rule of type II by scaling a DST I.

Let us suppose that the weight function w is symmetric. Since the Chebyshev polynomials of the second kind U_k of odd degree are odd functions, we have that $\lambda_{2s+1} = 0$. Using the notations introduced for the Fejér rules of type I, after simple calculations, also observing that since the weights are symmetric,

i.e. $w_k = w_{n-k+1}$ for $k = 1, \dots, n$, only the first $N = \lfloor (n-1)/2 \rfloor + 1$ weights must be computed, we get from (17)

$$\begin{aligned} w_j^* = w_{j+1} &= \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{\lfloor (n-1)/2 \rfloor} \lambda_{2s} \sin\left(\frac{(2s+1)(j+1)\pi}{n+1}\right), \\ &= \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{N-1} \lambda_{2s} \sin\left(\frac{(2s+1)(j+1)\pi}{n+1}\right), \quad j = 0, \dots, N-1 \end{aligned} \quad (22)$$

If n is even, then $N = n/2$ and (22) becomes

$$w_j^* = w_{j+1} = \frac{2 \sin(\theta_{j+1})}{2N+1} \sum_{s=0}^{N-1} \lambda_{2s} \sin\left(\frac{(2s+1)(j+1)\pi}{2N+1}\right), \quad j = 0, \dots, N-1 \quad (23)$$

Now we observe that a DST VI (see, e.g. [4]) maps $y_0, \dots, y_{n-1}$ into $\tilde{y}_0^{(VI)}, \dots, \tilde{y}_{n-1}^{(VI)}$ as

$$\tilde{y}_j^{(VI)} = \sum_{s=0}^{n-1} y_s \sin\left(\frac{(s+\frac{1}{2})(j+1)\pi}{n+\frac{1}{2}}\right), \quad j = 0, \dots, n-1. \quad (24)$$

and consequently the even moments $\lambda_0, \dots, \lambda_{2k}, \dots, \lambda_{2N-2}$ into $\tilde{\lambda}_0^{(VI)}, \dots, \tilde{\lambda}_{2k}^{(VI)}, \dots, \tilde{\lambda}_{2N-2}^{(VI)}$ as

$$\tilde{\lambda}_{2j}^{(VI)} = \sum_{s=0}^{N-1} \lambda_{2s} \sin\left(\frac{(s+\frac{1}{2})(j+1)\pi}{N+\frac{1}{2}}\right), \quad j = 0, \dots, N-1. \quad (25)$$

giving

$$w_j^* = w_{j+1} = \frac{2 \sin(\theta_{j+1})}{2N+1} \tilde{\lambda}_{2j}^{(VI)}, \quad j = 0, \dots, n-1. \quad (26)$$

If n is odd then $N = (n+1)/2$ and (22) gives

$$\begin{aligned} w_j^* = w_{j+1} &= \frac{2 \sin(\theta_{j+1})}{2N-1} \sum_{s=0}^{N-1} \lambda_{2s} \sin\left(\frac{(2s+1)(j+1)\pi}{2N}\right), \\ &= \frac{2 \sin(\theta_{j+1})}{2N-1} \sum_{s=0}^{N-1} \lambda_{2s} \sin\left(\frac{(s+\frac{1}{2})(j+1)\pi}{N}\right), \quad j = 0, \dots, N-1 \end{aligned} \quad (27)$$

We recognize this time that the DST II maps $y_0, \dots, y_{N-1}$ into $\tilde{y}_0^{(II)}, \dots, \tilde{y}_{N-1}^{(II)}$ defined by

$$\tilde{y}_j^{(II)} = \sum_{s=0}^{N-1} y_s \sin\left(\frac{\pi(s+\frac{1}{2})(j+1)}{N}\right), \quad j = 0, \dots, N-1 \quad (28)$$

from (27), the even moments $\lambda_0, \dots, \lambda_{2k}, \dots, \lambda_{2N-2}$ are mapped by DST II into $\tilde{\lambda}_0^{(II)}, \dots, \tilde{\lambda}_{2k}^{(II)}, \dots, \tilde{\lambda}_{2N-2}^{(II)}$ as

$$\tilde{\lambda}_{2j}^{(II)} = \sum_{s=0}^{N-1} \lambda_{2s} \sin \left(\frac{\pi (s + \frac{1}{2}) (j + 1)}{N} \right), \quad j = 0, \dots, N-1 \quad (29)$$

and we easily get

$$w_j^* = w_{j+1} = \frac{\sin(\theta_{j+1})}{N} \tilde{\lambda}_{2j}^{(II)}, \quad j = 0, \dots, N-1. \quad (30)$$

Remark 1 *Since*

$$U_k(x) = \begin{cases} 2 \sum_{j \text{ odd}}^k T_j(x), & k \text{ odd} \\ 2 \sum_{j \text{ even}}^k T_j(x) - 1, & k \text{ even} \end{cases}$$

we have for odd k

$$\lambda_k = \int_{-1}^1 U_k(x) w(x) dx = 2 \sum_{j \text{ odd}}^k \int_{-1}^1 T_j(x) w(x) dx = 2 \sum_{j \text{ odd}}^k \gamma_j$$

while, being $\gamma_0 = \int_{-1}^1 w(x) dx$, for even k

$$\lambda_k = \int_{-1}^1 U_k(x) w(x) dx = 2 \sum_{j \text{ odd}}^k \int_{-1}^1 T_j(x) w(x) dx - \int_{-1}^1 w(x) dx = 2 \sum_{j \text{ even}}^k \gamma_j - \gamma_0.$$

Consequently the moments of the Chebyshev polynomials of second kind $\{\lambda_s\}_{s=0, \dots, n-1}$ can be obtained from those of the Chebyshev polynomials of first kind $\{\gamma_s\}_{s=0, \dots, n-1}$.

4 Clenshaw-Curtis rules

The n -points Clenshaw-Curtis rule in the reference interval $[-1, 1]$ has nodes

$$x_k = \cos \left(\frac{(k-1)\pi}{n-1} \right), \quad k = 1, \dots, n.$$

It is straightforward to observe that they are the nodes of the $(n-2)$ -points Fejér rule of type II with the union of the interval extrema $-1, 1$. Setting as $P_n(x) \equiv (1-x^2) U_{n-2}(x)$ and $\Pi_n(x) \equiv \prod_{k=1, \dots, n} (x - x_k)$ we readily have for $j = 2, \dots, n-1$, since $U_{n-2}(x_2) = \dots = U_{n-2}(x_{n-1}) = 0$,

$$\begin{aligned}
w_j &= \frac{1}{\Pi'_n(x_j)} \int_{-1}^1 \frac{\Pi_n(x)w(x)}{x - x_j} dx \\
&= \frac{1}{P'_n(x_j)} \int_{-1}^1 \frac{P_n(x)w(x)}{x - x_j} dx \\
&= \frac{1}{(1 - x_j^2) U'_{n-2}(x_j)} \int_{-1}^1 \frac{(1 - x^2)U_{n-2}(x)w(x)}{x - x_j} dx
\end{aligned} \tag{31}$$

If we define $w^*(x) = (1 - x^2) w(x)$ we get

$$w_j = \frac{1}{(1 - x_j^2) U'_{n-2}(x_j)} \int_{-1}^1 \frac{U_{n-2}(x)w^*(x)}{x - x_j} dx \tag{32}$$

Comparing (32) with the first equation in (15), we observe that if

$$w_k^{(f2)} = \frac{1}{U'_{n-2}(x_k)} \int_{-1}^1 \frac{U_{n-2}(x)}{x - x_k} w^*(x) dx, \quad k = 1, \dots, n-2$$

are the weights of the $(n-2)$ -points Fejér rule of type II w.r.t. the weight function w^* then

$$w_j = \frac{1}{1 - x_j^2} w_{j-1}^{(f2)}, \quad j = 2, \dots, n-1.$$

In other words, the Clenshaw-Curtis weights $w_2, \dots, w_{n-2}$ w.r.t. w can be derived by the weights of the $(n-2)$ -point Fejér rule of type II w.r.t. w^* .

The computation of the weights w_1 and w_n , relative to the nodes $x_1 = 1$ and $x_n = -1$, is of different nature. Being $P_n(x) = (1 - x^2) U_{n-2}(x)$, $P'(1) = -2 U_{n-2}(1)$, $U_{n-2}(1) = n-1$, we get

$$\frac{P_n(x)}{P'_n(1)} = \frac{(1 - x^2)U_{n-2}(x)}{-2 U_{n-2}(1)} = \frac{(1 - x^2) U_{n-2}(x)}{-2(n-1)},$$

and from

$$w_1 = \frac{1}{P'_n(1)} \int_{-1}^1 \frac{P_n(x)w(x)}{x - 1} dx$$

we easily have

$$w_1 = \frac{1}{2(n-1)} \int_{-1}^1 (1+x) U_{n-2}(x)w(x) dx. \tag{33}$$

We can determine the integral on the r.h.s. of (33) from the Chebyshev moments $\gamma_k = \int_{-1}^1 T_k(x)w(s)dx$. If k is even, since $T_0(x) = 1$,

$$U_k(x) = 2 \sum_{j \text{ even}}^k T_j(x) - 1 = T_0(x) + 2 \sum_{j=1}^{k/2} T_{2j}(x).$$

Now, being $T_1(x) = xT_0(x)$, $xT_j(x) = \frac{1}{2}(T_{j+1}(x) + T_{j-1}(x))$ for $j = 1, 2, \dots$, we have

$$\begin{aligned}
\int_{-1}^1 x U_k(x) w(x) dx &= \gamma_1 + 2 \sum_{j=1}^{k/2} \int_{-1}^1 x T_{2j}(x) w(x) dx \\
&= \gamma_1 + \sum_{j=1}^{k/2} (\gamma_{2j+1} + \gamma_{2j-1}) \\
&= \gamma_1 + 2 \sum_{j=1}^{(k/2)-1} \gamma_{2j+1} + \gamma_{k+1}.
\end{aligned} \tag{34}$$

Consequently, from

$$\int_{-1}^1 U_k(x) w(x) dx = \int_{-1}^1 \left(T_0(x) + 2 \sum_{j=1}^{k/2} T_{2j}(x) \right) w(x) dx = \gamma_0 + 2 \sum_{j=1}^{k/2} \gamma_{2j}, \tag{35}$$

we finally have for even k

$$\begin{aligned}
\int_{-1}^1 (1+x) U_k(x) w(x) dx &= \left(\gamma_0 + 2 \sum_{j=1}^{k/2} \gamma_{2j} \right) + \left(\gamma_1 + 2 \sum_{j=1}^{(k/2)-1} \gamma_{2j+1} + \gamma_{k+1} \right) \\
&= \gamma_0 + 2 \sum_{j=1}^k \gamma_j + \gamma_{k+1}
\end{aligned} \tag{36}$$

The case in which k is odd is very similar. Easy computations provide, from

$$\int_{-1}^1 x U_k(x) w(x) dx = \sum_{j \text{ odd}}^k (\gamma_{j+1} + \gamma_{j-1}) = \gamma_0 + 2 \sum_{j \text{ even}}^{k-1} \gamma_j + \gamma_{k+1},$$

as well as $U_k(x) = 2 \sum_{j \text{ odd}}^k T_j(x)$ and $xT_j(x) = \frac{1}{2}(T_{j+1}(x) + T_{j-1}(x))$,

$$\begin{aligned}
\int_{-1}^1 (1+x) U_k(x) w(x) dx &= 2 \sum_{j \text{ odd}}^k \int_{-1}^1 (1+x) T_j(x) w(x) dx \\
&= 2 \sum_{j \text{ odd}}^k \gamma_j + \sum_{j \text{ odd}}^k (\gamma_{j+1} + \gamma_{j-1})
\end{aligned} \tag{37}$$

$$= \gamma_0 + 2 \sum_{j=1}^k \gamma_j + \gamma_{k+1}, \tag{38}$$

that is the same quantity of the even case (see (36)).

Hence from (33), since (37) holds for any k , we have

$$\begin{aligned} w_1 &= \frac{1}{2(n-1)} \int_{-1}^1 (1+x) U_{n-2}(x) w(x) dx \\ &= \frac{1}{2(n-1)} \left(\gamma_0 + 2 \sum_{j=1}^{n-2} \gamma_j + \gamma_{n-1} \right). \end{aligned} \quad (39)$$

The case of the node $x_n = -1$ is similar. Since $U_n(-1) = (-1)^n(n+1)$, $P_n(x) = (1-x^2)U_{n-2}(x)$, $P'_n(-1) = 2U_{n-2}(-1)$, then

$$\frac{P_n(x)}{P'_n(-1)} = \frac{(1-x^2)U_{n-2}(x)}{-2U_{n-2}(-1)} = \frac{(1-x^2)U_{n-2}(x)}{2(-1)^n(n-1)},$$

and from

$$w_n = \frac{1}{P'_n(-1)} \int_{-1}^1 \frac{P_n(x)w(x)}{x+1} dx$$

we finally have

$$w_n = \frac{1}{2(n-1)(-1)^n} \int_{-1}^1 (1-x)U_{n-2}(x)w(x)dx, \quad (40)$$

The integral on the r.h.s. of (40) can be computed with the same technique used for the node $x_1 = 1$. After some trivial computations,

$$\int_{-1}^1 (1-x)U_s(x)w(x)dx = \begin{cases} 2 \sum_{k=0}^{s+1} (-1)^k \gamma_k - \gamma_0 + \gamma_{s+1}, & s \text{ even} \\ 2 \sum_{k=0}^{s+1} (-1)^{k+1} \gamma_k + \gamma_0 + \gamma_{s+1}, & s \text{ odd.} \end{cases}$$

that provides the explicit formula of the weight w_n

$$w_n = \begin{cases} \frac{1}{2(n-1)} \left(2 \sum_{k=0}^{n-1} (-1)^k \gamma_k - \gamma_0 + \gamma_{s+1} \right), & s \text{ even} \\ \frac{(-1)}{2(n-1)} \left(2 \sum_{k=0}^{n-1} (-1)^{k+1} \gamma_k + \gamma_0 + \gamma_{s+1} \right), & s \text{ odd,} \end{cases}$$

i.e.

$$w_n = \frac{1}{2(n-1)} \left(2 \sum_{k=0}^{n-1} (-1)^k \gamma_k - \gamma_0 + (-1)^s \gamma_{s+1} \right).$$

Remark 2 We point out that there is an alternative strategy for determining w_n . Since the rule is interpolatory, the sum of the weights is equal to the first moment $\gamma_0 = \int_{-1}^1 w(x) dx$, hence

$$w_n = \gamma_0 - \sum_{j=1}^{n-1} w_j.$$

Remark 3 *If the weight function w is symmetric, i.e. $w(x) = w(-x)$, then one can prove that $w_1 = w_n$. Thus we easily have*

$$w_1 = w_n = \frac{1}{2} \left(\int_{-1}^1 w(x) dx - \sum_{k=2}^{n-1} w_k \right) = \frac{1}{2} \left(\gamma_0 - \sum_{k=2}^{n-1} w_k \right),$$

implying that in this case, the computation of w_1, w_n is straightforward once that $\gamma_0 = \int_{-1}^1 w(x) dx$ and $w_2, \dots, w_{n-1}$ are determined.

Remark 4 *Since*

$$(1 - x^2) U_k(x) = x T_{k+1}(x) - T_{k+2}(x), \quad k = 0, 1, \dots$$

and

$$x T_{k+1}(x) = \frac{1}{2} (T_{k+2}(x) + T_k(x)), \quad k = 0, 1, \dots$$

we have

$$\begin{aligned} (1 - x^2) U_k(x) &= x T_{k+1}(x) - T_{k+2}(x) = \frac{1}{2} (T_{k+2}(x) + T_k(x)) - T_{k+2}(x) \\ &= \frac{1}{2} (-T_{k+2}(x) + T_k(x)), \quad k = 0, 1, \dots \end{aligned} \quad (41)$$

Hence, setting $\gamma_k = \int_{-1}^1 T_k(x) w(x) dx$, the moments μ_k needed to compute the Clenshaw-Curtis weights $w_2, \dots, w_{n-1}$ for w via the Fejer rule of type II w.r.t $w^(x) = (1 - x^2) w(x)$ verify the equality*

$$\begin{aligned} \mu_k &= \int_{-1}^1 (1 - x^2) U_k(x) w(x) dx \\ &= \frac{1}{2} \int_{-1}^1 (-T_{k+2}(x) + T_k(x)) w(x) dx \\ &= \frac{1}{2} (-\gamma_{k+2} + \gamma_k). \end{aligned} \quad (42)$$

This implies that the modified weighted moments $\{\mu_k\}_{k=0, \dots, n-1}$ of the Chebyshev second kind polynomial w.r.t “ w^ ”, as well as $\int_{-1}^1 w(x) dx$, are easily available from the modified moments $\{\gamma_k\}_{k=0, \dots, n+1}$ of the Chebyshev first kind polynomials w.r.t the weight function “ w ”.*

Thus the weights of Fejer rule of degree $n - 2$, w.r.t. “ w^ ” can be easily computed and from them also the weights $w_2, \dots, w_n$ of the Clenshaw-Curtis rule w.r.t. “ w ”.*

5 Matlab codes

We have implemented all these rules in Matlab, that provides the routines `dct`, `idct`, `dst`, `idst` but not all the existing DCTs and DSTs [17]. In particular, apart from some normalisations, `dct` is a DCT I, `idct` is a DCT III, `dst` and `idst` are DST I (see [18] for further details). Thus, they are sufficient to define codes that determine Fejér type I, Fejér type II and Clenshaw-Curtis rules for a general weight function w .

The Fejér type I rule can be easily obtained by the weighted moments of the Chebyshev polynomials of first type $\gamma_0, \dots, \gamma_{n-1}$, but in view of the normalisation of `idct` one must transform the vector $\left[\frac{1}{\sqrt{2}}\gamma_0; \gamma_1; \dots; \gamma_{n-1}\right]$. With this observation, the column vectors of the nodes $\mathbf{x} = [x_1; \dots; x_n]$ and of the weights $\mathbf{w} = [w_1; \dots; w_n]$ can be obtained in Matlab by the function

```
function [x,w]=fejerI(n,moms)
moms(1)=sqrt(1/2)*moms(1);
x=cos((2*(1:n)-1)*pi/(2*n));
w=sqrt(2/n)*idct(moms);
```

where `moms` is the column vector of weighted modified moments $[\gamma_0; \gamma_1; \dots; \gamma_{n-1}]$ that must be provided by the user.

The Fejér rule of type II for a general weight function w needs the computation of a DST I of the weighted moments of the Chebyshev polynomials of first type $\lambda_0, \dots, \lambda_{n-1}$ that, as we have shown, can be obtained from $\gamma_0, \dots, \gamma_{n-1}$, that we suppose stored in the vector `moms`. Hence, observed that `dst` is the standard DST I (see (20)), the nodes $\mathbf{x}$ and the weights $\mathbf{w}$ are given by the short code

```
function [x,w]=fejerII(n,moms)
momsII=compute_moments_IIw(n-1,moms);
theta=(1:n)*pi/(n+1); theta=theta'; x=cos(theta);
w=(2*sin(theta)/(n+1)).*dst(momsII);
```

where `momsII=compute_moments_IIw(n-1,moms)` is a call that calculates the vector $[\lambda_0; \dots; \lambda_{n-1}]$ from $[\gamma_0; \dots; \gamma_{n-1}]$.

Concerning the Clenshaw-Curtis rule, supposing again to have computed the weighted moments `momsIIcc` $=[\mu_0; \dots; \mu_{n-3}]$ of the Chebyshev polynomial of the second kind (w.r.t. the weight function $(1-x^2)w(x)$) from the weighted moments `moms` $=[\gamma_0; \dots; \gamma_{n-1}]$ of the Chebyshev polynomial of the first kind (w.r.t. the weight function $w(x)$), nodes $\mathbf{x}$ and weights $\mathbf{w}$ are at hand with the routine

```

function [x,w]=clenshaw_curtis(n,moms)
momsIIcc=compute_moments_IIwcc(n-3,moms);
theta=(1:n-2)'*pi/(n-1); xx=cos(theta);
w=((2*sin(theta)/(n-1)).*dst(momsIIcc))./(1-xx.^2);
w1=(2*sum(moms)-moms(1)-moms(end))/(2*(n-1));
wn=moms(1)-w1-sum(w);
x=[1;xx;-1]; w=[w1;w;wn];

```

where `momsIIcc=compute_moments_IIwcc(n-3,moms)` is a call that calculates the vector $[\mu_0; \dots; \mu_{n-3}]$ from the moments $[\gamma_0; \dots; \gamma_{n-1}]$, stored in the column vector `moms`. As described in the Remark 3 of the previous section, this code can be simplified in the case of symmetric weight functions w , computing the weights w_1, w_n from γ_0 and the already known quantities $w_2, \dots, w_{n-1}$.

6 Numerical experiments with Gegenbauer weight functions

In [16], Hunter and Smith investigated a method for the numerical evaluation of integrals over $(-1, 1)$ of functions of the form

$$(1 - x^2)^{\lambda - \frac{1}{2}} f(x), \quad \lambda > -\frac{1}{2},$$

based on approximating f by a finite Chebyshev expansion. The case $\lambda = 1/2$ correspond to the classical Legendre weight function $w \equiv 1$, already studied by Waldvogel [25]. In [16], they have also shown that

$$\gamma_0 = \int_{-1}^1 (1 - x^2)^{\lambda - \frac{1}{2}} dx = \frac{\Gamma(\lambda + \frac{1}{2}) \cdot \sqrt{\pi}}{\Gamma(\lambda + 1)}, \quad (43)$$

and

$$\gamma_{2k} = \int_{-1}^1 T_{2k}(x) (1 - x^2)^{\lambda - \frac{1}{2}} dx = \frac{\Gamma(\lambda + \frac{1}{2}) \cdot \sqrt{\pi}}{\Gamma(\lambda + 1)} \cdot G_k(\lambda) = \gamma_0 G_k(\lambda), \quad k = 1, 2, \dots \quad (44)$$

where

$$G_k(\lambda) = \prod_{j=1}^k \frac{j - 1 - \lambda}{j + \lambda}.$$

The odd moments γ_{2k+1} , $k = 0, 1, \dots$ are null, since w is symmetric and the Chebyshev polynomials of first type and of odd degree T_{2k+1} are odd functions.

Thus, the weighted modified moments $\{\gamma_k\}$ are easily available and we can use our implementation of the Fejér and Clenshaw-Curtis rules to determine

Nodes	Fejér I	Fejér II	Clen.-Cur.
2	$2.07e-04$	$5.04e-04$	$9.09e-04$
4	$2.22e-04$	$5.88e-04$	$9.21e-04$
8	$2.39e-04$	$5.90e-04$	$9.33e-04$
16	$2.63e-04$	$6.06e-04$	$9.51e-04$
32	$2.79e-04$	$6.74e-04$	$9.58e-04$
64	$2.86e-04$	$6.76e-04$	$9.62e-04$
128	$2.99e-04$	$6.93e-04$	$9.75e-04$
256	$3.42e-04$	$7.26e-04$	$1.03e-03$
512	$3.51e-04$	$9.11e-04$	$1.16e-03$
1024	$6.13e-04$	$1.40e-03$	$2.18e-03$
2048	$1.43e-03$	$2.81e-03$	$3.95e-03$
4096	$1.45e-03$	$2.75e-03$	$4.14e-03$
8192	$4.36e-03$	$8.04e-03$	$1.18e-02$
16384	$5.47e-03$	$1.11e-02$	$1.68e-02$
32768	$1.14e-02$	$2.05e-02$	$3.44e-02$
65536	$2.44e-02$	$4.77e-02$	$7.23e-02$
131072	$5.21e-02$	$1.33e-01$	$2.48e-01$
262144	$9.96e-02$	$2.30e-01$	$3.52e-01$
524288	$2.12e-01$	$5.67e-01$	$1.05e+00$
1048576	$4.57e-01$	$1.13e+00$	$1.70e+00$

Table 1

Cputime (in seconds) required for the computation of Fejér rules of type I, II and Clenshaw-Curtis rules for Gegenbauer quadrature ($\lambda = 0.75$).

these quadrature formulas for the Gegenbauer weight function

$$w(x) = (1 - x^2)^{\lambda - \frac{1}{2}}, \quad \lambda > -\frac{1}{2}.$$

As numerical experiment we compute the nodes and the weights of these rules for the case $\lambda = 0.75$ and $n = 2^k$ with $k = 1, \dots, 20$. The results are listed in Table 1.

In our tests, we have used Matlab 7.6 (v. R 2008a), on a 2.5 Ghz Intel Core i5 with 8GB of RAM. The relative codes are available at the author's homepage [23].

Numerical experiments show that the cputime needed to compute the modified weighted moments $\{\gamma_k\}$, is very small w.r.t. the overall cost. It is interesting to compare these results with those needed to calculate the relative Gaussian rules. Their computation can be achieved with the codes that have been proposed in [2], [15] using the approach suggested in [14], or alternatively with those cited in [19] by a more classic variant of the Golub-Welsch algorithm. For

a Gaussian rule with $n = 64$, $n = 512$, i.e. with algebraic degree of precision 127 and 1023, the faster method needs respectively $1.80 \cdot 10^{-3}$ and $1.22 \cdot 10^{-2}$ seconds. The comparison of these results with those in Table 1, makes the formulas studied in this paper particularly appealing.

We also observe that due to the lack of special DCTs and DSTs in Matlab, we have used a general purpose implementation. The weight function w is in this case symmetric and fast DCTs/DSTs, as e.g. those suggested in [4], could improve the performance.

7 Numerical experiments with trigonometric quadrature

In [1], [5], [6], [7], [11], the authors studied the problem of constructing a quadrature rule with $n + 1$ angles and positive weights, exact in the $(2n + 1)$ -dimensional space of trigonometric polynomials of degree $\leq n$ on intervals with length smaller than 2π . In [5] it has been shown that

Theorem 1 *Let $\{(\xi_j, \lambda_j)\}_{1 \leq j \leq n+1}$, be the nodes and positive weights of the algebraic Gaussian quadrature formula for the weight function*

$$w(x) = \frac{2 \sin(\omega/2)}{\sqrt{1 - \sin^2(\omega/2)x^2}}, \quad x \in (-1, 1). \quad (45)$$

Then, denoting by $\mathbb{T}_n([- \omega, \omega]) = \text{span}\{1, \cos(k\theta), \sin(k\theta), 1 \leq k \leq n, \theta \in [- \omega, \omega]\}$, the $(2n+1)$ -dimensional space of trigonometric polynomials on $[- \omega, -\omega]$, we have

$$\int_{-\omega}^{\omega} f(\theta) d\theta = \sum_{j=1}^{n+1} \lambda_j f(\phi_j), \quad \forall f \in \mathbb{T}_n([- \omega, -\omega]), \quad 0 < \omega \leq \pi$$

where

$$\phi_j = 2 \arcsin(\sin(\omega/2)\xi_j) \in (-\omega, \omega), \quad j = 1, 2, \dots, n + 1.$$

An effective computation of these Gaussian rules has been studied in [19], while a formula of Fejér type I has been considered in [1]. In view of these results, we are interested in studying Fejér and Clenshaw-Curtis rules relatively to w defined in (45).

As it is apparent from the previous sections, one of the key points is the fast computation of the weighted modified moments $\{\gamma_k\}$. As described in [7], this can be efficiently done by the Olver's algorithm [20]. Numerical experiments

show that the moments computation is just a fraction of the overall process, that is however extremely fast.

In Table 2 we report the performance of each rule, for the cardinalities $n = 2^k$, with $k = 1, \dots, 20$. Though one hardly uses $n > 256$, it is important to observe that the whole computation of the rules (including the determination of the moments $\{\gamma_k\}$) is very competitive if compared to the calculation of the relative Gaussian rules. For example, the average cost of a Gaussian rule, via a fast method, of $n = 64$ and $n = 512$ points is respectively of $1.8855 \cdot 10^{-2}$ and $2.8007 \cdot 10^{-2}$ seconds, while the cost of a Clenshaw-Curtis with almost the same degree of precision is $4.4192 \cdot 10^{-4}$ and $8.4563 \cdot 10^{-4}$. As in the case of Gegenbauer weights, since function w is symmetric then fast DCTs/DSTs, as e.g. those suggested in [4], could allow inferior cputimes.

Nodes	Fejér I	Fejér II	Clen.-Cur.
2	$1.27e-04$	$2.04e-04$	$2.03e-04$
4	$1.57e-04$	$2.11e-04$	$2.23e-04$
8	$2.38e-04$	$2.18e-04$	$2.27e-04$
16	$2.37e-04$	$2.24e-04$	$2.28e-04$
32	$2.32e-04$	$2.61e-04$	$2.79e-04$
64	$2.73e-04$	$2.96e-04$	$3.17e-04$
128	$2.97e-04$	$3.62e-04$	$4.42e-04$
256	$4.22e-04$	$4.41e-04$	$4.52e-04$
512	$5.19e-04$	$5.19e-04$	$5.90e-04$
1024	$8.48e-04$	$8.72e-04$	$8.46e-04$
2048	$1.50e-03$	$1.99e-03$	$1.50e-03$
4096	$2.54e-03$	$2.47e-03$	$2.39e-03$
8192	$4.69e-03$	$4.90e-03$	$4.78e-03$
16384	$8.23e-03$	$8.56e-03$	$9.74e-03$
32768	$1.67e-02$	$1.71e-02$	$1.85e-02$
65536	$3.03e-02$	$4.46e-02$	$3.62e-02$
131072	$7.00e-02$	$9.61e-02$	$1.90e-01$
262144	$1.25e-01$	$1.51e-01$	$1.39e-01$
524288	$2.64e-01$	$4.20e-01$	$5.61e-01$
1048576	$5.49e-01$	$7.90e-01$	$6.94e-01$

Table 2

Cputime (in seconds) required for the computation of Fejér rules of type I, II and Clenshaw-Curtis rules for trigonometric quadrature.

We finally point out that all the computed rules enjoy positive weights, implying their numerical stability in view of Polya-Steklov theorem.

References

- [1] L. Bos and M. Vianello, Subperiodic trigonometric interpolation and quadrature, *Appl. Math. Comput.* 218 (2012), pp. 10630–10638.
- [2] Chebfun Team, Gauss-Jacobi rules, (available online at <http://www.mathworks.com/matlabcentral/fileexchange/23972-chebfun/content/chebfun/jacpts.m>).
- [3] C.W. Clenshaw and A.R. Curtis, *A method for numerical integration on an automatic computer*, *Numerische Mathematik* 2, 197 (1960), pp. 197–205.
- [4] R.K. Chivukula and Y.A. Reznik, *Fast Computing of Discrete Cosine and Sine Transforms of Types VI and VII*, *Proc. SPIE Vol. 8135*, 2011.
- [5] G. Da Fies and M. Vianello, *Trigonometric Gaussian quadrature on subintervals of the period*, *Electron. Trans. Numer. Anal.* 39 (2012), pp. 102–112.
- [6] G. Da Fies and M. Vianello, *Algebraic cubature on planar lenses and bubbles*, *Dolomites Res. Notes Approx.* 5 (2012), pp. 7–12.
- [7] G. Da Fies, A. Sommariva and M. Vianello, *Algebraic cubature on generalized elliptical sectors*, submitted.
- [8] P.J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Second edition. Dover Publications, Inc. 1984.
- [9] L. Fejér, *On the infinite sequences arising in the theories of harmonic analysis, of interpolation, and of mechanical quadrature*, *Bullettin of the American Mathematical Society*, 39 (1933), pp. 521–534.
- [10] L. Fejér, *Mechanische Quadraturen mit positiven Cotesschen Zahlen*, *Mathematische Zeitschrift*, 37, 287 (1933), pp. 287–309.
- [11] W. Gautschi, *Sub-range Jacobi polynomials*, *Numer. Algorithms*, published online March 13, 2012.
- [12] W.M. Gentleman, *Implementing Clenshaw-Curtis quadrature, I methodology and experience*, *Communications of the ACM*, Volume 15 Issue 5, May 1972, pp. 337–342.
- [13] W.M. Gentleman, *Implementing Clenshaw-Curtis quadrature, II computing the cosine transformation*, *Communications of the ACM*, Volume 15 Issue 5, May 1972, pp. 343–346.
- [14] A. Glaser, X. Liu and V. Rokhlin, *A fast algorithm for the calculation of the roots of special functions*, *SIAM J. Sci. Comput.* 29 (2007), pp. 1420–1438.
- [15] N. Hale and L.N. Trefethen, *Chebfun and Numerical quadrature*, *Science in China*, submitted.

- [16] D.B. Hunter and H.V. Smith, A quadrature formula of ClenshawCurtis type for the Gegenbauer weight-function, *Journal of Computational and Applied Mathematics* 177 (2005), pp. 389–400.
- [17] S. Martucci, *Symmetric Convolution and the Discrete Sine and Cosine Transform*, IEEE Transactions of Signal Processing, Vol. 42, No. 5, May 1994, pp. 1038–1051.
- [18] Mathworks, *Matlab R2012a Documentation, Signal Processing Toolbox*, (available online at <http://www.mathworks.it/help/toolbox/signal/>).
- [19] G. Meurant and A. Sommariva, *Fast variants of the Golub and Welsch algorithm for symmetric weight functions*, submitted.
- [20] F.W.J. Olver, *Numerical solution of second order linear difference equations*, J. Res. Nat. Bur. Standards 71B (1967), pp. 111–129.
- [21] I.H. Sloan, W.E. Smith, *Product integration with the Clenshaw-Curtis and related points. Convergence properties.*, Numer. Math., 30 (1978), pp. 415–428.
- [22] I.H. Sloan, W.E. Smith, *Product integration with the Clenshaw-Curtis points: implementation and error estimates*, Numer. Math., 34 (1980), pp. 387–401.
- [23] A. Sommariva, *Software homepage*, (software available online at <http://www.math.unipd.it/~alvise/software.html>).
- [24] L.N. Trefethen, *Is Gauss quadrature better than Clenshaw-Curtis?*, SIAM Review 50 (2008), pp. 67–87.
- [25] J. Waldvogel, *Fast Construction of the Fejér and Clenshaw-Curtis quadrature rules*, BIT Numerical Mathematics, Vol. 43, No. 1, (2003), pp. 1–18.