BIT Numerical Mathematics (2006) 46: 195–202
DOI: 10.1007/s10543-006-0045-4
Published online: 2 March 2006 – © Springer 2006

FAST CONSTRUCTION OF THE FEJÉR AND CLENSHAW–CURTIS QUADRATURE RULES*

JÖRG WALDVOGEL[1]

[1]*Seminar for Applied Mathematics, Swiss Federal Institute of Technology ETH, CH-8092 Zurich, Switzerland. email: waldvoge@math.ethz.ch*

Abstract.

We present an elegant algorithm for stably and quickly generating the weights of Fejér's quadrature rules and of the Clenshaw–Curtis rule. The weights for an arbitrary number of nodes are obtained as the discrete Fourier transform of an explicitly defined vector of rational or algebraic numbers. Since these rules have the capability of forming nested families, some of them have gained renewed interest in connection with quadrature over multi-dimensional regions.

AMS subject classification (2000): 65D32, 65T20, 65Y20.

Key words: numerical quadrature, Fejér's quadrature rules, Clenshaw–Curtis quadrature, discrete Fourier transform.

1 Introduction.

Interpolatory quadrature rules are based on a given set of $n + 1$ evaluation points (nodes) x_k on the integration interval. Quadrature weights w_k are determined such that the definite integrals of all polynomials of degree $\leq n$ are exactly represented by the weighted sums of the values of the integrand at the nodes x_k. Popular interpolatory quadrature rules are the Newton–Cotes rules (see, e.g., [2]), where uniformly distributed nodes are being used.

In view of applications to quadrature in high-dimensional regions, families of nested quadrature rules, Q_j $(j = 0, 1, \dots)$, are of particular interest. In a nested family the set of nodes of Q_j is a subset of the nodes of Q_l for every pair l, j with $l > j$. Nested families of 1-dimensional quadrature rules are ingeniously taken advantage of in the Smolyak construction [12] of *sparse grids* in order to obtain multidimensional cubature rules with a relatively small number of evaluation points. Smolyak's ideas have recently been successfully used by several authors, e.g. Bungartz and Griebel [1], K. Petras [10], [11]. Comprehensive lists of references on sparse grids and their applications are given in [1], [11].

Clearly, nested families may be constructed by means of Newton–Cotes rules. However, as suggested by the error theory of interpolation, as well as by the

* Received September 2005. Revised January 2006. Communicated by Tom Lyche.

structure of the well known Gaussian quadratures, a mesh with decreasing step size towards the boundaries of the interval leads to significantly smaller approximation errors. In fact, Gaussian quadrature rules seem to lend themselves to building nested families by repeated extensions (Kronrod [8]; Patterson [9]); however, we propose an alternative approach which is theoretically better understood: the second Fejér and the Clenshaw–Curtis quadrature rules. Kautsky and Elhay [7], [8] developed algorithms and software for calculating the weights of general interpolatory quadratures. We will specialize on the nodes defined by Equation (2.1) below and develop explicit representations of the weights in terms of discrete Fourier transforms (DFTs).

A connection between the Fejér and Clenshaw–Curtis quadrature rules and DFTs is no surprise. In fact, already in 1972 W. M. Gentleman [6] implemented the Clenshaw–Curtis rule with $n+1$ nodes by means of a discrete cosine transformation, which has to be carried out anew at every instance of quadrature, however. Recently, a direct computation (once for all) of the Clenshaw–Curtis weights by means of DFTs of order $2n$ was submitted to *The Mathworks Central File Exchange* by G. von Winckel [13].

Our independent approach is along the same lines. We will present unified algorithms based on DFTs of order n for generating the weights of the two Fejér rules and of the Clenshaw–Curtis rule. A streamlined Matlab code is given as well. Since all three rules have the capability of forming nested families, they are suitable as basic rules for generating Smolyak sparse grids.

2 The rules by Fejér and Clenshaw-Curtis.

Let $n \geq 2$ be a given fixed integer, and define $n+1$ quadrature nodes on the standard interval $[-1,1]$ as the extremes of the Chebyshev polynomial $T_n(x)$, augmented by the boundary points,

$$x_k := \cos\vartheta_k, \quad \vartheta_k := k\frac{\pi}{n}, \quad k = 0, 1, \ldots, n\ . \tag{2.1}$$

Interpolatory quadratures approximate the definite intergal of a given function f by a weighted sum,

$$\int_{-1}^{1} f(x)\,dx = \sum_{k=0}^{n} w_k f(x_k) \,+\, R_n\,, \tag{2.2}$$

where R_n is the approximation error, and w_k are the quadrature weights. These may be obtained by integrating the n-th-degree polynomial interpolating the $n+1$ discrete points $(x_k, f(x_k))$.

Applying this procedure to the nodes (2.1) directly yields the Clenshaw–Curtis rules. Fejér's second rule [4] is obtained by omitting the nodes $x_0 = 1$ and $x_n = -1$ and using the interpolating polynomial of degree $n-2$. This may also be achieved by keeping the boundary points as nodes, but preassigning the corresponding weights as $w_0 = w_n = 0$. We will adopt this unconventional approach in order to obtain a unified treatment of the two rules. Fejér's first rule [4] is obtained by using the well-known Chebyshev points as nodes, i.e., x_k from (2.1) with $k = \frac{1}{2}, \frac{3}{2}, \dots, n - \frac{1}{2}$; the corresponding weights will also be expressed in terms of discrete Fourier transforms.

L. Fejér [3] gives explicit expressions for the weights w_k^{f1} and w_k^{f2} of his quadrature rules, together with a proof of their positiveness. These expressions were rederived in a concise way by W. Gautschi [5]. The explicit expressions for the weights considered are summarized by Davis and Rabinowitz [2]: the weights w_k^{f1}, w_k^{f2} of Fejér's rules are given in Equations (2.5.5.4) and (2.5.5.8) on p. 85; for the Clenshaw–Curtis weights w_k^{cc} see p. 86. With the notation used in the present note the explicit expressions for the Fejér weights are

$$\begin{aligned} w_k^{f1} &= \frac{2}{n}\left(1 - 2\sum_{j=1}^{[n/2]} \frac{1}{4j^2-1}\cos(j\vartheta_{2k+1})\right), \quad k = 0, 1, \dots, n-1 \ , \\ w_k^{f2} &= \frac{4}{n}\sin\vartheta_k \sum_{j=1}^{[n/2]} \frac{\sin(2j-1)\vartheta_k}{2j-1}, \quad k = 0, 1, \dots, n \ , \end{aligned} \tag{2.3}$$

and the Clenshaw–Curtis weights are given by

$$w_k^{cc} = \frac{c_k}{n}\left(1 - \sum_{j=1}^{[n/2]} \frac{b_j}{4j^2-1}\cos(2j\vartheta_k)\right), \quad k = 0, 1, \dots, n \ , \tag{2.4}$$

where the coefficients b_j, c_k are defined as

$$b_j = \begin{cases} 1, & j = n/2 \\ 2, & j < n/2 \,, \end{cases} \qquad c_k = \begin{cases} 1, & k = 0 \mod n \\ 2, & \text{otherwise} \,. \end{cases} \tag{2.5}$$

All of the above equations hold for every even or odd integer $n > 1$. Conveniently, $w_0^{f2} = w_n^{f2} = 0$ follows directly from (2.3), and Equation (2.4), together with the definition (2.5) of c_k implies

$$w_0^{cc} = w_n^{cc} = \frac{1}{n^2 - 1 + \mathrm{mod}(n,2)} \ , \tag{2.6}$$

in agreement with the particular values defined in [2] on p. 86. By the way, the equation for w_k^{cc} on p. 86 is incomplete: the factor c_k/n is missing.

3 The weights of Fejér's second rule as a discrete Fourier transform.

In the following we present equivalent expresions for w_k^{f2} in terms of the inverse discrete Fourier transform of explicitly defined vectors of rational numbers. In the cases of n being a power of 2 the numerical implementations are particularly fast.

Discrete Fourier transforms of order n are linear mappings in the space of n-periodic sequences; traditionally the interval $[0, n-1]$ is used for all indices.

Therefore we impose the periodicity condition $w_{k+n} = w_k, k = 0, 1, \dots, n-1$ which implies $w_n = w_0$ in agreement with the symmetry of the weights. Throughout we use the notation

$$\omega = \omega_n := \exp\left(i\,\frac{2\pi}{n}\right). \tag{3.1}$$

We first rewrite the Fejér weights w_k^{f2} given by Equation (2.3) in terms of complex vectors and matrices. For simplicity we illustrate the difference between cases of even or odd values of n by means of the examples $n = 4$ and $n = 5$, respectively:

$$\begin{pmatrix} w_0^{f2} \\ w_1^{f2} \\ w_2^{f2} \\ w_3^{f2} \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ 0 & 0 & s_2 & 0 \\ 0 & 0 & 0 & s_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} \begin{pmatrix} -1/3 \\ -1 \\ 1 \\ 1/3 \end{pmatrix}$$

(3.2)

$$\begin{pmatrix} w_0^{f2} \\ w_1^{f2} \\ w_2^{f2} \\ w_3^{f2} \\ w_4^{f2} \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 & 0 \\ 0 & 0 & s_2 & 0 & 0 \\ 0 & 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 0 & s_4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} \end{pmatrix} \begin{pmatrix} 1 \\ 1/3 \\ 0 \\ -1/3 \\ -1 \end{pmatrix}.$$

Here $s_k, k = 0, 1, \dots, n-1$ is the abbreviation

$$s_k := \begin{cases} (-1)^k(1-\omega^k), & n \text{ even} \\ (1-\omega^k), & n \text{ odd}. \end{cases} \tag{3.3}$$

By introducing the n-vectors $\mathbf{w}^{f2} := (w_0^{f2}, w_1^{f2}, \dots, w_{n-1}^{f2})^T$ and

$$\mathbf{u} := \begin{cases} \left(-\frac{1}{n-1}, \dots, -\frac{1}{3}, -1, 1, \frac{1}{3}, \dots, \frac{1}{n-1}\right)^T, & n \text{ even} \\ \left(1, \frac{1}{3}, \dots, \frac{1}{n-2}, 0, -\frac{1}{n-2}, \dots, -\frac{1}{3}, -1\right)^T, & n \text{ odd} \end{cases} \tag{3.4}$$

as well as the diagonal matrix $S_n := \operatorname{diag}(s_0, s_1, \dots, s_{n-1})$ and the DFT matrices

$$F_n := \left(\omega_n^{-k\,l}\right)\Big|_{k,l=0}^{n-1}, \quad F_n^{-1} = \frac{1}{n}\left(\omega_n^{k\,l}\right)\Big|_{k,l=0}^{n-1}. \tag{3.5}$$

Equation (3.2) reads as

$$\mathbf{w}^{f2} = S_n\,F_n^{-1}\,\mathbf{u}. \tag{3.6}$$

The representation of the Fejér weights according to (3.6) already results in a considerable simplification compared to the known explicit expression. Many mathematical software systems offer efficient and numerically stable implementations of the DFT for arbitrary n, particularly fast (the proper "FFT") if n

is a power of 2. E.g., in Matlab the commands for the DFT (premultiplication by F_n) and the inverse DFT (premultiplication by F_n^{-1}) of `u` are `fft(u)` and `ifft(u)`, respectively.

Further simplification can be achieved by considering the DFT of $\mathbf{w}^{f2}$,

$$F_n\,\mathbf{w}^{f2} = T_n\,\mathbf{u} \quad \text{with} \quad T_n := F_n\,S_n\,F_n^{-1}\,. \tag{3.7}$$

A short computation with $n = 4$ and $n = 5$ yields

$$T_4 = \begin{pmatrix} 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \end{pmatrix}, \quad T_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix};$$

the elements t_{kl} of the matrix T_n generally are

$$t_{kl} = \begin{cases} 1, & l-k = h_n \bmod n \\ -1, & l-k = (h_n - 1) \bmod n \\ 0, & \text{otherwise} \end{cases} \quad \text{with} \quad h_n = \begin{cases} n/2, & n \text{ even} \\ 0, & n \text{ odd}\,. \end{cases} \tag{3.8}$$

From Equation (3.7) there follows

$$\mathbf{w}^{f2} = F_n^{-1}\,\mathbf{v} \quad \text{with} \quad \mathbf{v} = T_n\,\mathbf{u}\,, \tag{3.9}$$

and Equations (3.8) and (3.4) yield for the components v_k of $\mathbf{v}$ the explicit rational expressions

$$\begin{aligned} v_k \quad &= \frac{2}{1-4k^2}\,, \quad k = 0, 1, \ldots, \left[\frac{n}{2}\right] - 1\,, \\ v_{[n/2]} &= \frac{n-3}{2\,[n/2]-1} - 1\,, \\ v_{n-k} &= \overline{v_k}\,, \quad k = 1, 2, \ldots, \left[\frac{n-1}{2}\right], \end{aligned} \tag{3.10}$$

which hold for all even and odd integers $n > 1$. The redundant complex conjugation in the third line is used for consistency with Equation (4.4) below.

4 The Clenshaw–Curtis weights and Fejér's first rule.

From the explicit formulas (2.3) and (2.4) there follows for $k = 0, 1, \ldots, n-1$:

$$d_k := w_k^{cc} - w_k^{f2} = \begin{cases} \dfrac{c_k}{n^2-1}\,(-1)^k, & n \text{ even} \\ \dfrac{c_k}{n^2}\,(-1)^k \cos\left(k\dfrac{\pi}{n}\right), & n \text{ odd}\,, \end{cases} \tag{4.1}$$

where c_k is defined in (2.5). The DFTs $\mathbf{g}$ of the vectors $\mathbf{d} = (d_0, d_1, \ldots, d_{n-1})^T$ for $n = 4$ and $n = 5$ are

$$\mathbf{g} = \frac{1}{3\cdot 5}(\ -1 \quad -1 \quad 7 \quad -1\ )^T \quad \text{and} \quad \mathbf{g} = \frac{1}{5\cdot 5}(\ -1 \quad -1 \quad 4 \quad 4 \quad -1\ )^T,$$

respectively. For any $n > 1$ the components g_k of $\mathbf{g}$ are, in analogy to (3.10),

$$\begin{aligned} g_k &= -w_0^{cc}, \quad k = 0, 1, \ldots, \left[\frac{n}{2}\right] - 1, \\ g_{[n/2]} &= w_0^{cc}\left[(2 - \operatorname{mod}(n,2))\, n - 1\right], \\ g_{n-k} &= \overline{g_k}, \quad k = 1, 2, \ldots, \left[\frac{n-1}{2}\right], \end{aligned} \tag{4.2}$$

where w_0^{cc} is defined in (2.6). Therefore, by (3.9) and (4.1) the vector $\mathbf{w}^{cc}$ of the Clenshaw–Curtis weights may be obtained as the inverse Fourier transform of $\mathbf{v} + \mathbf{g}$.

The n-vector $\mathbf{w}^{f1}$ of the weights of Fejér's first rule, given by the first line of Equation (2.3), may equivalently be written as

$$\mathbf{w}^{f1} = F_n^{-1}\,\mathbf{v}, \tag{4.3}$$

where the vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ is now complex and defined by

$$\begin{aligned} v_k &= \frac{2}{1 - 4k^2}\, e^{i\,k\pi/n}, \quad k = 0, 1, \ldots, \left[\frac{n-1}{2}\right], \\ v_k &= 0, \quad \text{if} \quad k = \frac{n}{2}, \\ v_{n-k} &= \overline{v_k}, \quad k = 1, 2, \ldots, \left[\frac{n-1}{2}\right]. \end{aligned} \tag{4.4}$$

5 Conclusions.

We have established the following

THEOREM. *For the nodes $x_k := \cos(k\pi/n)$, $k = 0, 1, \ldots, n$ with $n > 1$ the weights $\mathbf{w}^{f2} = (w_0^{f2}, w_1^{f2}, \ldots, w_{n-1}^{f2})^T$ of Fejér's second quadrature rule are given by the inverse discrete Fourier transform of the vector $\mathbf{v}$ defined in (3.10), to be augmented by $w_n^{f2} := w_0^{f2}$.*

Analogously, the weights $\mathbf{w}^{cc} = (w_0^{cc}, w_1^{cc}, \ldots, w_{n-1}^{cc})^T$ of the Clenshaw–Curtis quadrature rule are given by the inverse discrete Fourier transform of the vector $\mathbf{v} + \mathbf{g}$, where $\mathbf{g}$ is defined in (4.2), again with $w_n^{cc} := w_0^{cc}$.

The weights $\mathbf{w}^{f1} = (w_0^{f1}, w_1^{f1}, \ldots, w_{n-1}^{f1})^T$ of Fejér's first quadrature rule, using the Chebyshev nodes $x_{k+1/2}$ with $k = 0, 1, \ldots, n-1$, are given by the inverse discrete Fourier transform of the complex vector $\mathbf{v}$ defined in (4.4).

In the Matlab function below the rational vectors `v`, `g` corresponding to Equations (3.10), (4.2) and (4.4) are generated via auxiliary vectors `v0`, `g0` in a somewhat streamlined way.

```
function [wf1,wf2,wcc] = fejer(n)
% Weights of the Fejer2, Clenshaw-Curtis and Fejer1 quadratures
% by DFTs. Nodes: x_k = cos(k*pi/n), n>1
N=[1:2:n-1]'; l=length(N); m=n-l; K=[0:m-1]';

% Fejer2 nodes: k=0,1,...,n; weights: wf2, wf2_n=wf2_0=0
v0=[2./N./(N-2); 1/N(end); zeros(m,1)];
v2=-v0(1:end-1)-v0(end:-1:2); wf2=ifft(v2);

%Clenshaw-Curtis nodes: k=0,1,...,n;  weights: wcc, wcc_n=wcc_0
g0=-ones(n,1); g0(1+l)=g0(1+l)+n; g0(1+m)=g0(1+m)+n;
g=g0/(n^2-1+mod(n,2)); wcc=ifft(v2+g);

% Fejer1 nodes: k=1/2,3/2,...,n-1/2; vector of weights: wf1
v0=[2*exp(i*pi*K/n)./(1-4*K.^2); zeros(l+1,1)];
v1=v0(1:end-1)+conj(v0(end:-1:2)); wf1=ifft(v1);
```

To assess the efficiency of the above implementation by means of the DFT it was compared with a Matlab implementation of the classical explicit expressions [4], [5]. To be fair, the vectorized operations of Matlab and matrix-vector multiplications were used as much as possible, at the cost of memory, though. Some average execution times (in milliseconds on a 1.6-GHz processor) for all three weight vectors together are collected in the following table:

n	classical	DFT	n	classical	DFT
8	0.26	0.27	17	0.40	0.32
16	0.38	0.29	31	0.83	0.37
32	0.85	0.30	65	2.60	0.40
64	2.60	0.35	67	2.77	0.51
128	9.95	0.52	127	9.32	0.66
256	42.6	0.86	255	43.3	0.80
512	166.5	1.09	257	42.6	1.21
1024	660.0	1.73	1021	653.5	3.58

As expected, the classical algorithm asymptotically scales as $O(n^2)$. The asymptotic complexity $O(n \log n)$ of the DFT algorithm is barely reached within the range of the table. Owing to a rather efficient implementation of the mixed-radix FFT in Matlab the new algorithm is competitive even for prime values of n.

The combined execution times given above for all three rules together are at most 80 % (often less than 50 %) of the time used by the Matlab code [13] for the Clenshaw–Curtis weights alone.

The accuracy of weights computed by the DFT algorithm is excellent. E.g., for $n = 128$ the relative error of the weights of Fejér's second rule is at most 6 eps (eps $= 2^{-52}$ is the machine epsilon of Matlab), with a quadratic mean of 1.4 eps. 86 % of the weights are in error by less than eps in magnitude. In this range the classical algorithm loses about 2 bits of accuracy compared to the DFT algorithm.

Acknowledgement.

The author thanks Professor Walter Gautschi of Purdue University for his valuable comments and suggestions. The recent references provided by the referee are gratefully acknowledged.

REFERENCES

1. H.-J. Bungartz and M. Griebel, *Sparse grids*, Acta Numer., 13 (2004), pp. 1–123.
2. P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd edn., Academic Press, San Diego, 612 pp.
3. S. Elhay and J. Kautsky, *Algorithm 655 – IQPACK: FORTRAN subroutines for the weights of interpolatory quadratures*, ACM Trans. Math. Softw., 13 (1987), pp. 399–415.
4. L. Fejér, *Mechanische Quadraturen mit positiven Cotesschen Zahlen*, Math. Z., 37 (1933), pp. 287–309.
5. W. Gautschi, *Numerical quadrature in the presence of a singularity*, SIAM J. Numer. Anal., 4 (1967), pp. 357–362.
6. W. M. Gentleman, *Implementing Clenshaw–Curtis quadrature*, Commun. ACM, 15 (1972), pp. 337–346. *Algorithm 424 (Fortran code)*, ibid., pp. 353–355.
7. J. Kautsky and S. Elhay, *Calculation of the weights of interpolatory quadratures*, Numer. Math., 40 (1982), pp. 407–422.
8. A. S. Kronrod, *Nodes and Weights of Quadrature Formulas*, Consultants Bureau, New York, 1965.
9. T. N. L. Patterson, *The optimum addition of points to quadrature formulae*, Math. Comput., 22 (1968), pp. 847–856. *Errata*, Math. Comput., 23 (1969), p. 892.
10. K. Petras, *On the Smolyak cubature error for analytic functions*, Adv. Comput. Math., 12 (2000), pp. 71–93.
11. K. Petras, *Smolyak cubature of given polynomial degree with few nodes for increasing dimension*, Numer. Math., 93 (2003), pp. 729–753.
12. S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Sov. Math. Dokl., 4 (1963), pp. 240–243.
13. G. von Winckel, *Fast Clenshaw–Curtis Quadrature*, The Mathworks Central File Exchange, Feb. 2005.
URL http://www.mathworks.com/matlabcentral/files/6911/clencurt.m