A STABLE GAUSS–KRONROD ALGORITHM FOR CAUCHY PRINCIPAL-VALUE INTEGRALS

P. RABINOWITZ
Department of Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel

(Received 9 December 1984)

Abstract—It is shown that the ratio of the precision of the stable Kronrod extension to the precision of the stable version of the Gauss rule for Cauchy principal-value integrals is approximately the same as the ratio of the precision of the Kronrod extension to that of the Gauss rule for ordinary integrals.

We shall be concerned here with the numerical evaluation of the Cauchy principal-value (CPV) integral

\[ I(f; \lambda) = \int_{-1}^{1} w(x) \frac{f(x)}{x - \lambda} \, dx, \quad -1 < \lambda < 1, \]

where \( w(x) \) is the Gegenbauer or ultraspherical weight function

\[ w(x) = (1 - x^2)^{\beta - 1/2} \]

and \( f \) satisfies a Hölder condition of order \( \alpha \) in \([-1, 1]\), \( f \in H_\alpha, 0 < \alpha \leq 1 \). We restrict \( \mu \) to the range \( 0 \leq \mu \leq 2 \) which insures the existence of a Kronrod extension (KE) to the Gauss–Gegenbauer integration rule

\[ I = \int_{-1}^{1} w(x)f(x) \, dx = G_\mu f + C_\mu f^{(\mu)}(\xi), \quad -1 < \xi < 1, \]

where

\[ G_\mu f = \sum_{i=1}^{\mu} w_i f(x_i), \]

the abscissas \( x_i \) are the zeros of the Gegenbauer polynomial \( C_\mu(x) \), and the weights \( w_i \) are interpolatory. The KE of equation (4) is given by

\[ K_\mu f = \sum_{i=1}^{\mu} w^*_i f(x_i) + \sum_{j=1}^{\mu+1} \nu_j f(y_j), \]

where the \( y_j \) are the zeros of the Stieltjes polynomial \( E_{\mu+1,\mu}(x) \) which satisfies the orthogonality relation

\[ \int_{-1}^{1} w(x)C_\mu(x)E_{\mu+1,\mu}(x)x^k \, dx = 0, \quad k = 0, 1, \ldots, n \]

and where, as before, the weights \( w^*_i \) and \( \nu_j \) are interpolatory. As shown by Szegö [1], for \( \mu \in [0, 2] \), the zeros \( y_j \) of \( E_{\mu+1,\mu}(x) \) are all real, lie in the closed interval \([-1, 1]\), and interlace the zeros \( x_i \) of \( C_\mu(x) \). For more information on Stieltjes polynomials and KEs, see Monegato [2]. The only additional item of interest here is the result in Ref. [3] that, except for the cases \( \mu = 0, 1 \), \( K_\mu f \) is of exact precision \( M = 2[(3n + 1)/2] + 1 \), so that the ratio of the precision of \( K_\mu f \) to that of \( G_\mu f \) is \( M/(2n - 1) \).

We shall now consider an integration rule for \( I(f; \lambda) \) based on the points \( x_i \). It turns out that there are two such rules. One

\[ G_\mu(f; \lambda) = \sum_{i=1}^{\mu} \left[ w_i - \frac{Q_\mu(x_i)}{C_\mu(x_i)} \right] \frac{f(x_i)}{x_i - \lambda}, \]
is based only on the points $x_n$, assumed distinct from $\lambda$. Here $Q_{\nu}(x)$ is the function of the second kind defined by

$$Q_{\nu}(x) = \int_{-1}^{1} \frac{w(t) C_{\nu}(t)}{t-x} \, dt, \quad -1 < x < 1,$$

which, as is well-known, satisfies the same three-term recurrence relation as $C_{\nu}(x)$. The second rule, based on the points $x_n$ and the point $\lambda \neq x_n$, $i = 1, \ldots, n$ is given by

$$G_{n+1}(f; \lambda) = \sum_{i=1}^{n} \frac{w_n}{x_n - \lambda} f(x_n) + \frac{Q_{\nu}(\lambda)}{C_{\nu}(\lambda)} f(\lambda).$$

These rules are given in Ref. [4] and have been studied extensively; see, for example, Gautschi [5]. Whereas the rule $G_n(f; \lambda)$ is only of precision $n - 1$, except for isolated values of $\lambda$, while $G_{n+1}(f; \lambda)$ is of precision $2n$, nevertheless $G_n(f; \lambda)$ can be evaluated in a stable manner for all $\lambda \in (-1, 1)$ though not by using equation (7), whereas the evaluation of $G_{n+1}(f; \lambda)$ may be unstable for $\lambda$ close to one of the points $x_n$. This difference in the stability properties of these two rules may reflect a difference in the convergence behavior of these rules. As stated in Ref. [5], $G_n(f; \lambda)$ converges to $I(f; \lambda)$ for all $f \in H$, for any $\alpha \in (0, 1]$. On the other hand, it was shown [6] that $G_{n+1}(f; \lambda)$ converges to $I(f; \lambda)$ for all $f \in H_1$, but that, for any $\alpha \in (0, 1)$ and any $\mu \in [0, 2]$, there exist points $\lambda \in (-1, 1)$ and functions $f \in H_1$ such that $G_{n+1}(f; \lambda)$ does not converge to $I(f; \lambda)$.

Now, in a previous paper [4], formulas were given for the KEs of both $G_n(f; \lambda)$ and $G_{n+1}(f; \lambda)$; in particular, the KE of $G_n(f; \lambda)$ is given by

$$K_n(f; \lambda) = \sum_{i=1}^{n} \left( w_n^* - u_n \right) \frac{f(x_n)}{x_n - \lambda} + \sum_{j=1}^{n-1} \left( v_j^* - v_j \right) \frac{f(y_j)}{y_j - \lambda},$$

where

$$u_n = \left( Q_{\nu}(\lambda) E_{n+1,u}(\lambda) - \delta_\mu \right)/E_{n+1,u}(x_n) C_{\nu}(x_n),$$

$$v_j^* = \left( Q_{\nu}(\lambda) E_{n+1,u}(\lambda) - \delta_\mu \right)/E_{n+1,u}(y_j) C_{\nu}(y_j),$$

and

$$\delta_\mu = -2\Gamma(\mu + \frac{1}{2})/\Gamma(2\mu).$$

This formula, as well as the corresponding formula for the KE of $G_{n+1}(f; \lambda)$, share with formulas (7) and (9) the property of being numerically unstable for $\lambda$ close to the $x_n$ and, in addition, for $\lambda$ close to the $y_j$. However, as shown by Paget and Elliott [7], and previously by Kornečuk [8], there is a stable way to evaluate $G_n(f; \lambda)$ for all $\lambda \in (-1, 1)$ based on the functions of the second kind, $Q_{\nu}(x)$. The idea is to approximate $f(x)$ by the finite Gegenbauer expansion

$$f(x) \approx \sum_{k=0}^{n-1} a_k C_{k\nu}(x),$$

where

$$a_k = (f, C_{k\nu})/(C_{k\nu}, C_{k\nu})$$

and the discrete inner product $(g, h)$ is given by

$$(g, h) = G_n(gh) = \sum_{i=1}^{n} w_n g(x_n) h(x_n).$$

The $a_k$ are approximations to the Fourier coefficients in the Gegenbauer expansion of $f(x)$,

$$f(x) \sim \sum_{k=0}^{n} b_k C_{k\nu}(x)$$

with $b_k = I(f C_{k\nu})/I([C_{k\nu}]^2)$. The $a_k$ are equal to the $b_k$ if $f(x) \in P_{n-1}$, the set of all
polynomials of degree \( \leq n - 1 \) since \( G_n f \) is of precision \( 2n - 1 \). Thus, if \( f \in P_{n - 1} \), then

\[
f(x) = s_{n-1}(x) = \sum_{k=1}^{n-1} a_k C_{kn}(x).
\]

Since the \( a_k \) for \( f(x) \) equal the \( a_k \) for the polynomial \( p_{n-1}(x) \) interpolating to \( f(x) \) at the points \( x_{kn} \), \( i = 1, \ldots, n \), it follows that \( p_{n-1}(x) = s_{n-1}(x) \). Since \( G_n(f; \lambda) \) results from integrating \( p_{n-1}(x) \), we have that \( G_n(f; \lambda) = I(s_{n-1}; \lambda) \). However,

\[
I(s_{n-1}; \lambda) = \sum_{k=1}^{n-1} a_k \int_{-1}^{1} \frac{w(x)}{x - \lambda} C_{kn}(x) \, dx = \sum_{k=0}^{n-1} a_k Q_{kn}(\lambda)
\]

and this last term can be evaluated in a stable manner using the Clenshaw backward-recurrence algorithm [7]. Note that Elliott and Paget [9] give a similar algorithm for the evaluation of \( G_{n+1}^2(f; \lambda) \):

\[
G_{n+1}^2(f; \lambda) = \sum_{k=1}^{n} a_k Q_{kn}(\lambda) + \hat{a}_n Q_{n-1}(\lambda),
\]

where

\[
\hat{a}_n = \sum_{i=1}^{n} \left( \frac{f(\lambda) - f(x_i)}{\lambda - x_i} \right) C_{kn}(x_i).
\]

However, it is clear from equation (17) that the evaluation of \( \hat{a}_n \) is unstable for \( \lambda \) close to some \( x_{kn} \).

Now, when we apply this idea to the KE of \( G_n(f; \lambda) \), we find an interesting phenomenon which we point out below. Proceeding as above, we approximate \( f(x) \) by the finite Gegenbauer expansion

\[
f(x) \approx \sum_{k=0}^{2n} a_k^* C_{k}(x) \equiv s_n^* (x),
\]

where

\[
a_k^* = \frac{K_n(f; C_{kn})}{[C_{kn}]^2}.
\]

However, in contrast to the Gauss case, \( f \) need not be equal to \( s_n^* \) if \( f \in P_{2n} \). This is so since \( K_n \) is not of precision \( 4n + 1 \) but only of precision \( M \). Hence, the best we can say is that if

\[
N = \frac{M - 1}{2},
\]

then the approximation

\[
f(x) \approx \sum_{k=0}^{N} a_k^* C_k(x) \equiv s_n^* (x)
\]

becomes an equality if \( f \in P_N \). Hence, we can proceed as before and approximate \( I(f; \lambda) \) by \( I(s_n^*; \lambda) \), yielding

\[
I(f; \lambda) \approx \sum_{k=0}^{N} a_k^* Q_{kn}(\lambda)
\]

which can also be evaluated in a stable manner by backward recurrence. Note that in the computation of the coefficients \( a_k^* \), we have used the values of \( f(x) \) at the Gauss points \( x_k \) and at the Kronrod points \( y_k \). The ratio of the precision of formula (21) to formula (15) is

\[
\frac{N}{n-1} = \frac{M - 1}{2n - 2} \approx \frac{M}{2n - 1},
\]

the ratio of the precision of \( K_n f \) to \( G_n f \),
We close with several remarks and an example.

1. The use of Gauss–Kronrod rules for CPV integrals is called for when we wish to evaluate \( I(f; \lambda) \) for a set of values of \( \lambda \). In this case, we need to compute the \( f(x_n) \) and \( f(y_n) \) only once and use these values repeatedly for each value of \( \lambda \). For a one-shot integral, an adaptive approach based on a modified Clenshaw–Curtis rule, as given in QUADPACK [10], should be satisfactory. Alternatively, one could use the algorithm proposed by Monegato [11]. However, these approaches are inefficient in the above situation since they cannot re-use previously computed function values.

We note further that if we need to evaluate \( I(f; \lambda) \) for \( p \) values of \( \lambda \), \( p > n + 1 \), then the use of \( G_{2n+1}(f; \lambda) \) which is of the same precision as \( G_{n+1}(f; \lambda) \) requires \( 2n + 1 \) function evaluations while the use of \( G_{n+1}(f; \lambda) \) requires a total of \( n + p \) evaluations. Hence, the use of \( G_{2n+1}(f; \lambda) \) is not only more stable but also more efficient.

2. Once we have computed the \( f(x_n) \) and \( f(y_n) \) we can evaluate \( a_k^\ast \) for any \( k \) and consequently approximate \( I(f; \lambda) \) by

\[
I(s_k^\ast; \lambda) = \sum_{k=1}^{K} a_k^\ast Q_k(\lambda), \quad K > N. \tag{22}
\]

The precision of this approximation will be less than \( N \); however, for an arbitrary function \( f(x) \), it may be more accurate than formula (21). To see this, we note that the error

\[
E_k(f; \lambda) = I(f; \lambda) - I(s_k^\ast; \lambda)
\]

can be expressed in the form

\[
E_k(f; \lambda) = \sum_{k=0}^{K} (b_k - a_k^\ast) Q_k(\lambda) + \sum_{k=K+1}^{N} b_k Q_k(\lambda),
\]

from which we see that if \( K = N \) and \( f \in P_N \), then \( b_k = a_k^\ast, k = 0, \ldots, N \) and \( b_k = 0, k > N \) so that \( E_k(f; \lambda) = 0 \). However, for other functions, the size of \( E_k(f; \lambda) \) will depend on how well the \( b_k \) are approximated by the \( a_k^\ast \) and on the rate of decrease in the size of the Fourier coefficients \( b_k \) as well as on the values of \( Q_k(\lambda) \). If we take \( K > N \), we should change the definition of \( a_k^\ast \) by replacing the denominator \( K_n([C_k]^2) \) by \( I([C_k]^2) \) which is known analytically. When this is done

\[
b_k - a_k^\ast = \frac{I(fC_k) - K_n(fC_k)}{I([C_k]^2)}
\]

and its size depends on how well the Kronrod rule integrates the function \( f(x)C_k(x) \). Since the optimal value of \( K \) depends on so many unknown factors, one can only choose something by analogy and the choice \( K = 2n \) appears as a reasonable alternative to \( K = N \) and, in our example, we shall compare the two choices. We note that for \( K = 2n \), the precision of the approximation is \( n + 2 \).

3. For the case \( \mu = 0 \) (1) corresponding to the Gauss–Chebyshev integration rule of the first (second) kind, the precision of \( K_n f \) is \( 4n - 1 \) (\( 4n + 1 \)) and we can replace \( N \) in formula (21) by

\[
K = \left(\frac{4n-1}{2}\right)\left[\frac{4n+1}{2}\right],
\]

Again the ratio of the precision of \( K_n f \) to \( G_n f \) will be about the same as that of \( I(s_k^\ast; \lambda) \) to \( G_n(f; \lambda) \).

4. One could extend this idea in an obvious way to the further optimal extensions proposed by Patterson [12]. Similarly, one can apply it to the KEs of the Lobatto–Gegenbauer rules which were also discussed in Ref. [4].

Example. In this example we illustrate some of the points mentioned above. We consider

\[
I(f; \lambda) = \int_{-1}^{1} \frac{e^x}{x - y} \, dx \tag{23}
\]
Evaluation of Cauchy principal-value integrals

Table 1. The relative errors in the approximation of equation (23) by formulas (21), (22) with \( K = 2n \), (10), (7) and (15) with \( n = 11 \) using single-precision arithmetic

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Formula (21)</th>
<th>Formula (22)</th>
<th>Formula (10)</th>
<th>Formula (7)</th>
<th>Formula (15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>-0.34E-05</td>
<td>-0.10E-04</td>
<td>-0.11E-03</td>
<td>-0.15E-04</td>
<td>0.0</td>
</tr>
<tr>
<td>0.05</td>
<td>-0.74E-05</td>
<td>-0.22E-04</td>
<td>-0.25E-04</td>
<td>-0.65E-05</td>
<td>-0.46E-06</td>
</tr>
<tr>
<td>0.10</td>
<td>-0.81E-05</td>
<td>-0.21E-04</td>
<td>-0.25E-04</td>
<td>-0.38E-05</td>
<td>-0.24E-05</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.78E-05</td>
<td>0.52E-05</td>
<td>0.36E-05</td>
<td>0.47E-05</td>
<td>0.36E-05</td>
</tr>
<tr>
<td>0.30</td>
<td>-0.88E-05</td>
<td>-0.13E-04</td>
<td>-0.17E-04</td>
<td>-0.36E-05</td>
<td>-0.35E-05</td>
</tr>
<tr>
<td>0.40</td>
<td>-0.22E-05</td>
<td>-0.22E-05</td>
<td>0.45E-03</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.50</td>
<td>0.026E-05</td>
<td>-0.16E-05</td>
<td>0.12E-03</td>
<td>0.12E-03</td>
<td>0.47E-05</td>
</tr>
<tr>
<td>0.60</td>
<td>-0.21E-04</td>
<td>-0.77E-05</td>
<td>-0.07E-04</td>
<td>-0.26E-04</td>
<td>0.22E-04</td>
</tr>
<tr>
<td>0.70</td>
<td>-0.27E-04</td>
<td>-0.39E-04</td>
<td>-0.15E-03</td>
<td>-0.20E-03</td>
<td>0.12E-04</td>
</tr>
<tr>
<td>0.80</td>
<td>-0.28E-05</td>
<td>-0.11E-05</td>
<td>0.27E-04</td>
<td>0.39E-04</td>
<td>0.22E-04</td>
</tr>
<tr>
<td>0.90</td>
<td>-0.72E-05</td>
<td>0.22E-04</td>
<td>0.12E-03</td>
<td>0.12E-03</td>
<td>0.47E-05</td>
</tr>
<tr>
<td>0.95</td>
<td>0.26E-05</td>
<td>0.25E-05</td>
<td>0.33E-03</td>
<td>0.31E-04</td>
<td>-0.26E-05</td>
</tr>
<tr>
<td>0.99</td>
<td>0.44E-05</td>
<td>0.25E-04</td>
<td>0.33E-03</td>
<td>0.31E-04</td>
<td>-0.26E-05</td>
</tr>
</tbody>
</table>

for \( \lambda = 0.01, 0.05, 0.1(0.1) 0.9, 0.95, 0.99 \), which was also studied by Monegato [11]. In this case \( \mu = \frac{1}{2} \) and \( C_n(x) = P_n(x) \), the Legendre polynomial. All our computations were carried out on the IBM 3081 at the Weizmann Institute of Science Computer Center. This computer has 6- to 7-digit accuracy in single-precision calculations and about 16-digit accuracy in double-precision. We first consider the Kronrod extension to \( G_{11}(f; \lambda) \). This was computed in three different ways using formulas (21), (22) with \( K = 2n \) and (10). The results when computed in double-precision were correct to over 10 significant figures so that any errors in the single-precision calculation are due strictly to roundoff. In Table 1, we list the relative errors in the single-precision approximations. We see that formulas (21) and (22) are generally more accurate than formula (10). This is especially noticeable for the values \( \lambda = 0.01, 0.4, 0.5 \) which is due to the fact that the set of abscissas in the Kronrod rule includes the values 0.0398 and 0.519 which are close to the "bad" values of \( \lambda \). The inaccuracies in formulas (21) and (22) are due to the roundoff error incurred in evaluating the Fourier–Legendre coefficients \( a_n^2 \) since the Legendre polynomials oscillate in the interval \([-1, 1]\) and there is some subtractive cancellation.

In Table 1, we also give the results of approximating formula (23) using the 11-point Gauss rule \( G_{11}(f; \lambda) \) in its two formulations, (7) and (15). We see here the larger errors incurred using the unstable formula (7), especially for the values \( \lambda = 0.01, 0.5, 0.7, 0.9 \), corresponding to the Gauss points 0, 0.519, 0.73 and 0.887.

We also carried out some computations with the Gauss rule \( G_{11}(f; \lambda) \) since we wanted to compare its accuracy with that of \( G_{11}(f; \lambda) \) inasmuch as both of these rules are of precision 10. Since we are interested here in truncation error rather than roundoff error, we give in Table 2 the relative errors in the double-precision calculations. The results indicate that the two formulas have about the same accuracy so that when we wish to calculate \( I(f; \lambda) \) for many values of \( \lambda \), the use of \( G_{11}(f; \lambda) \) may be advantageous.

We also give in Table 3 the results of a single-precision calculation using three different
formulations of $\hat{G}_6(f; \lambda)$, namely formulas (9), (16) and the following:

$$I(f; \lambda) \approx G_\lambda[g(x; \lambda)] + Q_{0\lambda}(\lambda)f(\lambda),$$

where

$$g(x; \lambda) = \begin{cases} 
\frac{f(x) - f(\lambda)}{x - \lambda}, & \lambda \neq x \\
\lambda = x.
\end{cases}$$

This formula arises by rewriting $I(f; \lambda)$ as

$$I(f; \lambda) = Ig + Q_{0\lambda}(\lambda)f(\lambda)$$

and approximating $Ig$ by $G_\lambda g$. As shown in Ref. [6], the r.h.s. of formula (24) is mathematically identical to formula (9) for $\lambda \neq x_m$, $i = 1, \ldots, n$. The results show that formula (24) is the most stable followed by formula (16) with formula (9) in last place.

We also computed the KE of $G_6(f; \lambda)$ by applying the KE to $g$ leading to the approximation

$$I(f; \lambda) \approx K_\lambda[g(x; \lambda)] + Q_{0\lambda}(\lambda)f(\lambda),$$

the precision of which is one greater than the precision of $K_\lambda f$. The results given in Table 3 are of surprisingly good accuracy.

For a further discussion of stability questions in the numerical evaluation of CPV integrals, see van der Sluis and Zweerus [13].

REFERENCES