International Journal of Pure and Applied Mathematics
Volume 28 No. 4 2006, 565-583

TRANSFORMS BETWEEN THE FOUR DIFFERENT GAUSS-CHEBYSHEV QUADRATURE FORMULAE

Lijin Wang[1], Rudolf Scherer[2 §]
[1]State Key Laboratory of Scientific and Engineering Computing
Institute of Computational Mathematics and
Scientific/Engineering Computing
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
P.O. Box 2719, Beijing, 100080, P.R. CHINA
e-mail: ljwang@lsec.cc.ac.cn
[1,2]Institute for Practical Mathematics
University of Karlsruhe
Englerstrasse 2, Karlsruhe, 76128, GERMANY
[2]e-mail: scherer@math.uni-karlsruhe.de

Abstract: For the numerical integration of a given function, the four different Gauss-Chebyshev quadrature formulae can be transformed into each other. In this article, the weights, nodes, cost, error bound representations, and numerical results of the quadrature formulae when transformed to each other are discussed. Different functions are integrated by using the four kinds of quadrature formulae. Numerical experiments compare the effect of these formulae.

AMS Subject Classification: 65D30, 65D32
Key Words: Chebyshev polynomials, derivative-free error estimate, Gauss-Chebyshev quadrature, orthogonality

1. Introduction

There are four kinds of Chebyshev polynomials which are orthogonal polynomials on the interval $[-1, 1]$ with respect to four different weight functions

Received: May 23, 2006
§Correspondence author

(Boyadjiev and Scherer [1], Mason [4], Mason and Handscomb [5]). Accordingly the four different kinds of Gauss-Chebyshev quadrature formulae arise, where their nodes are the zeros of the four different kinds of Chebyshev polynomials in $[-1, 1]$. Given an integration problem $\int_{-1}^{1} f(x)dx$, it is possible to choose an appropriate weight function $w(x)$, and then compute $\int_{-1}^{1} w(x)g(x)dx$ numerically by applying the quadrature formula corresponding to $w(x)$. The four different kinds of Gauss-Chebyshev quadrature formulae can be transformed into each other (see (2.7)-(2.11)) by varying the weight functions from $w_1(x)$ to $w_4(x)$ (see (2.5)), and the integrands from $g_1(x)$ to $g_4(x)$ (see (2.7)-(2.10)). Thus the questions arise, which one of the four formulae would be the best as far as numerical aspects such as nodes, weights, cost, accuracy, and so on are concerned, and which one would be the most appropriate.

In the literature, the first kind of Gauss-Chebyshev quadrature formulae are mostly mentioned. Only few attention is paid to the comparison between the four kinds yet. It is mentioned in Boyadjiev and Scherer [1] that all the other three kinds of Gauss-Chebyshev quadrature formulae may be transformed to the first kind, and in some cases these transforms can supply better numerical results.

Here the discussion deals with the numerical comparison between the four quadrature formulae when transformed into each other. The different nodes, weights and corresponding derivative-free error representations and error bounds are observed. However, it is still difficult to compare the four kinds of errors generally, due to different integrands. Numerical tests are performed for empirical analysis.

It is found that for odd integrands the first and second kind of formulae give exact numerical integration results for all number of nodes $n \geq 1$. The numerical results produced by the third and fourth kind of formulae are of opposite sign for odd integrands and all $n \geq 1$, while for even integrands, they are equal. Numerical cost in terms of the amount of elementary operations involved in the four formulae are found to be of no difference. It is obtained through numerical experiments that for a lot of even integrands the third and fourth kind of formulae are more efficient, while if the integrands are neither odd nor even, the third kind of formula functions usually the best. However, as n gets larger (e.g., $n \geq 15$), the difference in accuracy between the four formulae becomes small. The cost of applying the four kinds of quadrature formulae in terms of CPU times elapsed in computations are found to be equal.

Section 2 introduces the four different kinds of Gauss-Chebyshev quadrature formulae and the transforms into each other of a given integrand on the interval $[-1, 1]$. Section 3 compares the different nodes, weights, costs, and the error representations of the four kinds of formulae. Numerical experiments with respect to different integrands such as exponential, logarithmic, trigonometric, anti-trigonometric, rational functions are performed in Section 4. Certain empirical conclusions are obtained.

2. Gauss-Chebyshev Quadrature Formulae and their Transforms

The four kinds of Chebyshev polynomials are defined as

$$\text{kind } 1: \ T_n(x) = \cos n\theta, \qquad \text{kind } 2: \ U_n(x) = \frac{\sin(n+1)\theta}{\sin\theta}, \tag{2.1}$$

$$\text{kind } 3: \ V_n(x) = \frac{\cos(n+\frac{1}{2})\theta}{\cos\frac{\theta}{2}}, \ \text{kind } 4: \ W_n(x) = \frac{\sin(n+\frac{1}{2})\theta}{\sin\frac{\theta}{2}}, \tag{2.2}$$

where $x = \cos\theta$ with $\theta \in [0, \pi]$ and $x \in [-1, 1]$. Divided by the leading coefficients, the monic forms of the four kinds of Chebyshev polynomials are

$$\begin{aligned} \Phi_n^1(x) &= 2^{1-n}T_n(x), \quad \Phi_n^2(x) = 2^{-n}U_n(x), \\ \Phi_n^3(x) &= 2^{-n}V_n(x), \quad \Phi_n^4(x) = 2^{-n}W_n(x). \end{aligned} \tag{2.3}$$

One of the important properties of the Chebyshev polynomials is their orthogonality on the interval $[-1, 1]$ with respect to corresponding weight functions $w_i(x)$ (Boyadjiev and Scherer [1], Mason and Handscomb [5], Szegö [11])

$$\begin{aligned} \int_{-1}^{1} w_1(x)T_n(x)T_m(x)dx &= \begin{cases} \frac{\pi}{2}, & n = m, \\ 0, & n \neq m, \end{cases} \\ \int_{-1}^{1} w_2(x)U_n(x)U_m(x)dx &= \begin{cases} \frac{\pi}{2}, & n = m, \\ 0, & n \neq m, \end{cases} \\ \int_{-1}^{1} w_3(x)V_n(x)V_m(x)dx &= \begin{cases} \pi, & n = m, \\ 0, & n \neq m, \end{cases} \\ \int_{-1}^{1} w_4(x)W_n(x)W_m(x)dx &= \begin{cases} \pi, & n = m, \\ 0, & n \neq m, \end{cases} \end{aligned} \tag{2.4}$$

where

$$w_1(x) = \frac{1}{\sqrt{1-x^2}}, \qquad w_2(x) = \sqrt{1-x^2},$$

$$w_3(x) = \sqrt{\frac{1+x}{1-x}}, \qquad w_4(x) = \sqrt{\frac{1-x}{1+x}}. \tag{2.5}$$

Based on the orthogonality of the Chebyshev polynomials on $[-1,1]$, the four different Gauss-Chebyshev quadrature formulae arise as

$$\int_{-1}^{1} w_i(x) f(x) dx = \sum_{k=1}^{n} A_k^i f(x_k^i) + R_n^{(i)}(f), \quad i = 1,2,3,4, \tag{2.6}$$

where the nodes $x_k^{(i)}$ are the n zeros of the first, second, third, and fourth kind of Chebyshev polynomial of degree n in the interval $[-1,1]$, and $A_k^{(i)}$ are the corresponding weights, respectively. Both the nodes and weights are given in Section 3. The remainder terms are abbreviated by $R_n^{(i)}(f)$.

The integration problem $\int_{-1}^{1} f(x)dx$ can be integrated numerically by applying one of the following four kinds of Gauss-Chebyshev quadrature formulae

$$\begin{aligned} \int_{-1}^{1} f(x)dx &= \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \underbrace{f(x)\sqrt{1-x^2}}_{g_1(x)} dx \\ &= \sum_{k=1}^{n} A_k^{(1)} g_1(x_k^{(1)}) + R_n^{(1)}(g_1), \end{aligned} \tag{2.7}$$

$$\begin{aligned} \int_{-1}^{1} f(x)dx &= \int_{-1}^{1} \sqrt{1-x^2} \underbrace{\frac{f(x)}{\sqrt{1-x^2}}}_{g_2(x)} dx \\ &= \sum_{k=1}^{n} A_k^{(2)} g_2(x_k^{(2)}) + R_n^{(2)}(g_2), \end{aligned} \tag{2.8}$$

$$\begin{aligned} \int_{-1}^{1} f(x)dx &= \int_{-1}^{1} \sqrt{\frac{1+x}{1-x}} \underbrace{f(x)\sqrt{\frac{1-x}{1+x}}}_{g_3(x)} dx \\ &= \sum_{k=1}^{n} A_k^{(3)} g_3(x_k^{(3)}) + R_n^{(3)}(g_3), \end{aligned} \tag{2.9}$$

$$\begin{aligned} \int_{-1}^{1} f(x)dx &= \int_{-1}^{1} \sqrt{\frac{1-x}{1+x}} \underbrace{f(x)\sqrt{\frac{1+x}{1-x}}}_{g_4(x)} dx \\ &= \sum_{k=1}^{n} A_k^{(4)} g_4(x_k^{(4)}) + R_n^{(4)}(g_4). \end{aligned} \tag{2.10}$$

Since we have

$$\begin{aligned} g_2(x) &= \frac{g_1(x)}{1-x^2}, \qquad g_3(x)=\frac{g_1(x)}{1+x}, \qquad g_4(x)=\frac{g_1(x)}{1-x},\\ g_3(x) &= g_2(x)(1-x), \quad g_4(x)=g_2(x)(1+x),\\ g_4(x) &= g_3(x)(\frac{1+x}{1-x}), \end{aligned} \tag{2.11}$$

(2.7)-(2.10) can also be regarded as transforms between the four kinds of formulae. In the following section, comparisons are made on different numerical aspects of these transforms.

3. Comparisons of Numerical Aspects of the Transforms

The four kinds of nodes $x_k^{(i)}$ $(i=1,2,3,4;\ k=1,\cdots,n)$ used in the four kinds of Gauss-Chebyshev quadrature formulae (2.7)-(2.10) are given by (Mason and Handscomb [5], Rivlin [6], Szegö [11])

$$\begin{aligned} x_k^{(1)} &= \cos\frac{(k-\frac{1}{2})\pi}{n}, \quad x_k^{(2)}=\cos\frac{k\pi}{n+1},\\ x_k^{(3)} &= \cos\frac{(k-\frac{1}{2})\pi}{n+\frac{1}{2}}, \quad x_k^{(4)}=\cos\frac{k\pi}{n+\frac{1}{2}}, \end{aligned} \tag{3.1}$$

and the weights $A_k^{(i)}$ $(i=1,2,3,4;\ \ k=1,\cdots,n)$ are

$$\begin{aligned} A_k^{(1)} &= \frac{\pi}{n}, \qquad A_k^{(2)}=\frac{\pi}{n+1}(1-(x_k^{(2)})^2),\\ A_k^{(3)} &= \frac{\pi}{n+\frac{1}{2}}(1+x_k^{(3)}), \quad A_k^{(4)}=\frac{\pi}{n+\frac{1}{2}}(1-x_k^{(4)}). \end{aligned} \tag{3.2}$$

As an example, we refer to the transform from (2.7) to (2.9). Other transforms included in (2.7)-(2.10) can be discussed likewise. In (2.7), the nodes and weights applied for computation are $x_k^{(1)}$ and $A_k^{(1)}=\frac{\pi}{n}$, while the formula (2.9) underlies a computation with nodes $x_k^{(3)}$ and weights $A_k^{(3)}=\frac{\pi}{n+\frac{1}{2}}(1+x_k^{(3)})$. At first sight, computation with (2.7) seems easier

$$\int_{-1}^{1} f(x)dx=\frac{\pi}{n}\sum_{k=1}^{n} f(\cos\frac{(k-\frac{1}{2})\pi}{n})\sqrt{1-(\cos\frac{(k-\frac{1}{2})\pi}{n})^2}+R_n^{(1)}(g_1). \tag{3.3}$$

However, the integrand after the transform from (2.7) to (2.9) becomes $g_3(x) = \frac{g_1(x)}{1+x}$, which together with the form of the weights $A_k^{(3)}$ compensate, at least formally, the complexity of computation

$$\int_{-1}^{1} f(x)dx = \frac{\pi}{n+\frac{1}{2}} \sum_{k=1}^{n} f(\cos\frac{(k-\frac{1}{2})\pi}{n+\frac{1}{2}})\sqrt{1-(\cos\frac{(k-\frac{1}{2})\pi}{n+\frac{1}{2}})^2} + R_n^{(3)}(g_3). \quad (3.4)$$

From (3.3) and (3.4), we may judge that the cost of computation in terms of the number of elementary operations involved will not change after the transform from (2.7) to (2.9). Similar analysis can be done for the other transforms included in (2.7)-(2.10). Thus the following result is obtained.

Theorem 1. *Transforms in (2.7)-(2.10) between the four different kinds of Gauss-Chebyshev quadrature formulae keep the cost of the algorithms.*

Note that,

$$\cos\frac{(k-\frac{1}{2})\pi}{n} = -\cos\frac{(n-k+\frac{1}{2})\pi}{n}, \quad (3.5)$$

$$\cos\frac{k\pi}{n+1} = -\cos\frac{(n-k+1)\pi}{n+1}, \quad k = 1,\cdots,n, \quad (3.6)$$

then both the first kind of nodes $x_k^{(1)} = \cos\frac{(k-\frac{1}{2})\pi}{n}$ and the second kind of nodes $x_k^{(2)} = \cos\frac{k\pi}{n+1}$ distribute symmetrically about zero in $[-1,1]$. If the number of nodes n is even, they are pairwise of opposite sign. In case of odd n, zero itself is a node, and the other nodes are pairwise of opposite sign. Thus, if the function $f(x)$ is an odd function, which in turn makes the integrands $g_1(x)$ and $g_2(x)$ also odd, the numerical sum in (2.7) and (2.8) will be

$$\sum_{k=1}^{n} A_k^{(1)} g_1(x_k^{(1)}) = \sum_{k=1}^{n} A_k^{(2)} g_2(x_k^{(2)}) = 0 \quad (3.7)$$

due to the symmetric property (3.5)-(3.6) of the nodes. Note that the accurate value of $\int_{-1}^{1} f(x)dx$ is also zero when $f(x)$ is odd. We obtain the following result.

Theorem 2. *For odd integrands $f(x)$, the first and second kind of Gauss-Chebyshev quadrature formulae (2.7) and (2.8) give exact integration results for an arbitrary number of nodes $n \geq 1$.*

The third and fourth kind of nodes $x_k^{(3)}$ and $x_k^{(4)}$ are nonsymmetric about zero. However, it holds that

$$\cos\frac{(k-\frac{1}{2})\pi}{n+\frac{1}{2}} = -\cos\frac{(n+1-k)\pi}{n+\frac{1}{2}}, \quad k=1,\cdots,n, \tag{3.8}$$

which implies that the third and fourth kind of nodes are of opposite sign to each other. With the relation between $g_1(x)$ and $g_4(x)$ in (2.11) as well as the form of $A_k^{(4)}$ in (3.2), we obtain

$$\int_{-1}^{1} f(x)dx = \frac{\pi}{n+\frac{1}{2}}\sum_{k=1}^{n} f(\cos\frac{k\pi}{n+\frac{1}{2}})\sqrt{1-(\cos\frac{k\pi}{n+\frac{1}{2}})^2} + R_n^{(4)}(g_4). \tag{3.9}$$

Comparing (3.4) with (3.9), it is found that, if $f(x)$ is an odd function, for the numerical sum in (2.9) and (2.10) it holds

$$\sum_{k=1}^{n} A_k^{(3)} g_3(x_k^{(3)}) = -\sum_{k=1}^{n} A_k^{(4)} g_4(x_k^{(4)}). \tag{3.10}$$

If $f(x)$ is even, the relation between (2.9) and (2.10) is

$$\sum_{k=1}^{n} A_k^{(3)} g_3(x_k^{(3)}) = \sum_{k=1}^{n} A_k^{(4)} g_4(x_k^{(4)}), \tag{3.11}$$

as a consequence of relation (3.8) between the third and fourth kind of nodes. From (3.10)-(3.11) the following result is derived.

Theorem 3. *For odd integrands $f(x)$, the numerical integration results produced by the third and fourth kind of Gauss-Chebyshev quadrature formulae (2.9) and (2.10) are of opposite sign, while for even integrands $f(x)$, (2.9) and (2.10) give the same numerical integration values.*

For the error analysis of the quadrature formulae in (2.6), we refer to the contour error representation for analytic integrands $f(x)$ (Boyadjiev and Scherer [1], Gautschi [2], Gautschi and Varga [3], Scherer and Schira [7], Schira [8], Schira [9], Schira [10]),

$$R_n^{(i)}(f) = \frac{1}{2\pi i}\int_\Gamma K_n^{(i)}(z)f(z)dz, \tag{3.12}$$

where $f(x)$ is analytic in a region D, which contains $[-1,1]$ in its interior, and Γ is a contour in D surrounding $[-1,1]$. The kernel functions are given by

$$K_n^{(i)}(z) = \frac{Q_n^{(i)}(z)}{P_n^{(i)}(z)} \tag{3.13}$$

with

$$Q_n^{(i)}(z) = \int_{-1}^{1} \frac{w_i(x)P_n^{(i)}(x)}{z-x}dx, \tag{3.14}$$

and $P_n^{(i)}(x)$ are the four different kinds of Chebyshev polynomials. $Q_n^{(i)}(z)$ are called functions of second kind of $P_n^{(i)}(z)$. Based on the contour error representation (3.12), the following error bound estimates are achieved

$$|R_n^{(i)}(f)| \le \frac{l(\Gamma)}{2\pi} \max_{z\in\Gamma} |K_n^{(i)}(z)| \cdot \max_{z\in\Gamma} |f(z)|, \quad i = 1,2,3,4, \tag{3.15}$$

where $l(\Gamma)$ denotes the length of Γ. When Γ is a circle $|z| = r$, denoted by C_r, $r > 1$, the maxima of $K_n^{(i)}(z)$ are given by (Boyadjiev and Scherer [1], Gautschi and Varga [3], Schira [8])

$$\begin{aligned} \max_{z\in C_r} |K_n^{(i)}(z)| &= K_n^{(i)}(r), \quad i = 1,2,3, \\ \max_{z\in C_r} |K_n^{(4)}(z)| &= K_n^{(4)}(-r). \end{aligned} \tag{3.16}$$

For the ellipse contour

$$\Gamma_\rho = \{z : z = \frac{1}{2}(\rho e^{i\theta} + \rho^{-1}e^{-i\theta}),\ 0 \le \theta \le 2\pi,\ \rho > 1\}$$

with length $l(\Gamma_\rho)$, the results are (Boyadjiev and Scherer [1], Gautschi [2], Gautschi and Varga [3], Scherer and Schira [7], Schira [8], Schira [9], Schira [10])

$$\begin{aligned} \max_{z\in\Gamma_\rho} |K_n^{(i)}(z)| &= K_n^{(i)}(\frac{1}{2}(\rho+\rho^{-1})), \quad i = 1,3, \\ \max_{z\in\Gamma_\rho} |K_n^{(4)}(z)| &= K_n^{(4)}(-\frac{1}{2}(\rho+\rho^{-1})), \\ \max_{z\in\Gamma_\rho} |K_n^{(2)}(z)| &= \begin{cases} |K_n^{(2)}(\frac{i}{2}(\rho-\rho^{-1}))|, & n \text{ odd, or } n \text{ even and } \rho \ge \rho_{n+1}, \\ |K_n^{(2)}(\frac{1}{2}(\rho e^{i\theta^*} + \rho^{-1}e^{-i\theta^*}))|, & n \text{ even and } 1 < \rho < \rho_{n+1}, \end{cases} \end{aligned}$$

where ρ_{n+1} is the unique zero of the equation

$$\frac{a_1(\rho)}{a_{n+1}(\rho)} = \frac{1}{n+1} \quad (a_k(\rho) = \frac{1}{2}(\rho^k + \rho^{-k}),\ k \ge 1),$$

and

$$\frac{\frac{n}{n+1}\pi}{2} < \theta^* < \frac{\pi}{2}.$$

Thus we obtain the following results.

Theorem 4. *The error estimates of the four kinds of Gauss-Chebyshev quadrature formulae (2.7)-(2.10) are*

$$|R_n^{(i)}(g_i(x))| \leq \begin{cases} r \cdot \max_{x\in C_r} |K_n^{(i)}(x)| \cdot \max_{x\in C_r} |g_i(x)|, & r > 1, \\ \frac{l(\Gamma_\rho)}{2\pi} \cdot \max_{x\in \Gamma_\rho} |K_n^{(i)}(x)| \cdot \max_{x\in \Gamma_\rho} |g_i(x)|, & \rho > 1, \end{cases} \tag{3.17}$$

where $\max_{x\in\Gamma} |K_n^{(i)}(x)|$ *are given in (3.16) and (3.17).*

There are algorithms for the computation of the function values $|K_n^{(i)}(z)|$ if z is located on the axes (Gautschi and Varga [3]), which is the case in (3.16) and in the first two equations in (3.17). In the third equation in (3.17), the maximum point of $K_n^{(2)}(z)$ is either on the imaginary axis or close to the imaginary axis in which case $|K_n^{(2)}(z)|$ can be computed approximately. This makes it possible to numerically compute the error estimates. However, the maximum value of $g_i(x)$ on Γ may only be found for given $f(x)$ (see (2.7)-(2.10)), and is difficult to be discussed generally. Certain rules of the behavior of the errors are found in the numerical tests in Section 4.

4. Numerical Tests

In each of the numerical tests below, the given integration problems $\int_{-1}^{1} f(x)dx$ are solved numerically applying the four different kinds of Gauss-Chebyshev quadrature formulae (2.7)-(2.10). Their effects and differences can be seen from the figures and data in the tables. For the generality of discussion, the integrands $f(x)$ are chosen to be different functions.

Example 1. $f(x) = \sin(x)$. The numerical results are illustrated for the integrand $f(x) = \sin(x)$ with Gauss-Chebyshev quadrature formulae of the first and second kind (Figure 1.1), and third and fourth kind (Figure 1.2), with respect to different numbers of nodes $1 \leq n \leq 100$. It is obvious that the true value of this integration is 0.

The integration with the first and second kind of Gauss-Chebyshev quadrature formulae reaches a very high accuracy (10^{-16}) even with small n (e.g., n=5), while with increasing n there is no growth in accuracy; it keeps stable with slight oscillation in the scale of 10^{-16} as the machine error. This result coincides with the theoretical analysis in Theorem 2.

The third and fourth kind of Gauss-Chebyshev quadrature formulae are less accurate than the first and second kind as n is relatively small (e.g., $n \leq 10$). However, as n gets larger, the accuracy grows very fast. For a clearer

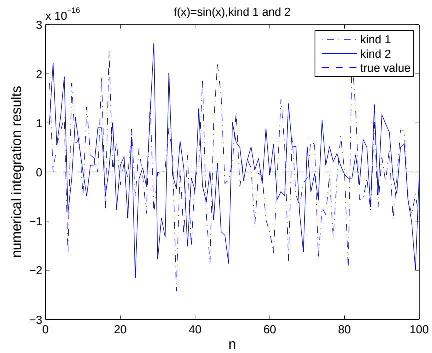

Figure 1.1: Numerical results of kind 1 and 2 as $1 \leq n \leq 100$

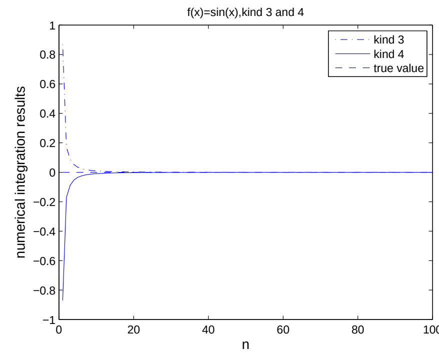

Figure 1.2: Numerical results of kind 3 and 4 as $1 \leq n \leq 100$

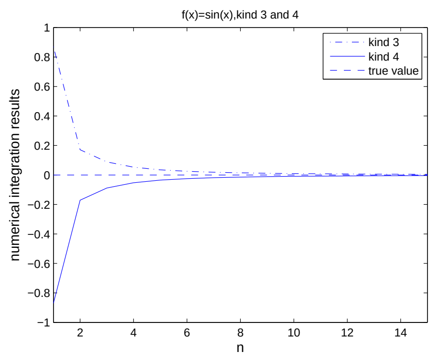

Figure 1.3: Numerical results of kind 3 and 4 as $1 \leq n \leq 15$

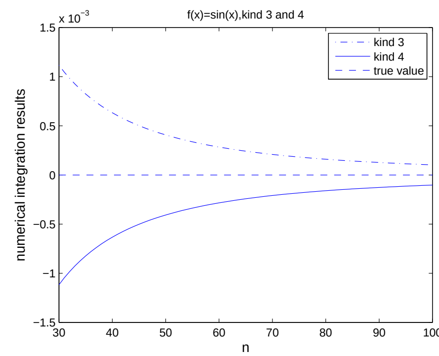

Figure 1.4: Numerical results of kind 3 and 4 as $30 \leq n \leq 100$

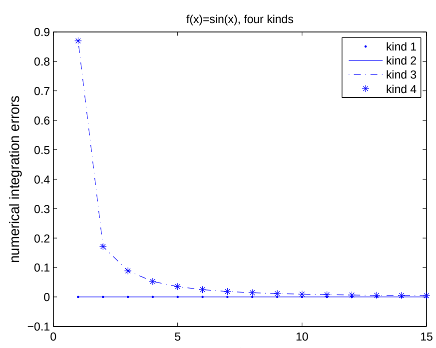

Figure 1.5: Errors of the four kinds as $1 \leq n \leq 15$

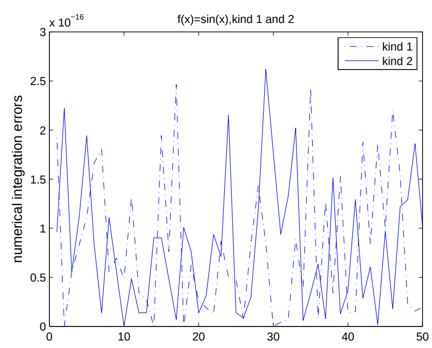

Figure 1.6: Errors of kind 1 and 2 as $1 \leq n \leq 50$

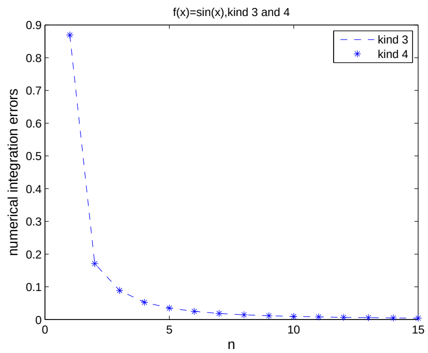

Figure 1.7: Errors of kind 3 and 4 as $1 \leq n \leq 15$

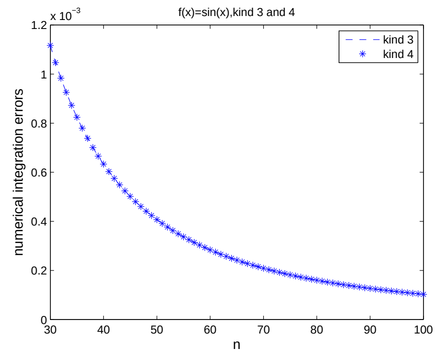

Figure 1.8: Errors of kind 3 and 4 as $30 \leq n \leq 100$

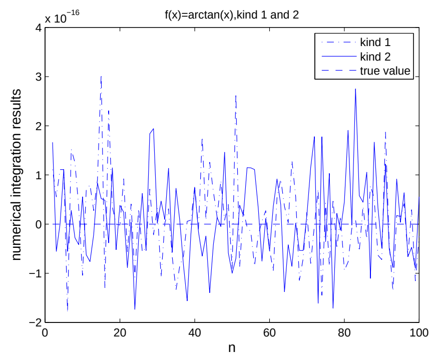

Figure 2.1: Numerical results of kind 1 and 2 as $2 \leq n \leq 100$

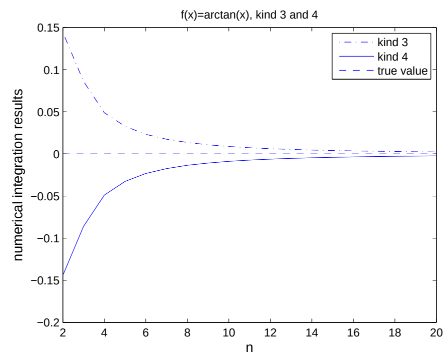

Figure 2.2: Numerical results of kind 3 and 4 as $2 \leq n \leq 20$

observation of the behavior of the third and fourth kind of formulae as $n \leq 15$ and $30 \leq n \leq 100$, see Figures 1.3, 1.4, 1.7, and 1.8.

The third and fourth kind of formulae reach the accuracy of 10^{-4} at about $n = 33$, which indicates explicitly the superiority of the first and second kind of formulae for the integrand $f(x) = \sin(x)$. The results produced by the third and fourth kind of formulae have opposite sign to each other for all n, which reflects exactly the result in Theorem 3.

Figure 1.5 shows the evolution of errors of the four kinds of formulae with $1 \leq n \leq 15$. The first and second kind create almost the exact value, while the third and fourth kind have relative large errors as n is small. At the same time, there is no difference in accuracy between the first and second kind, as can be seen from Figure 1.6. The errors of the third and fourth kind coincide for both small and relative large n, which is indicated by Figure 1.7 and 1.8. The

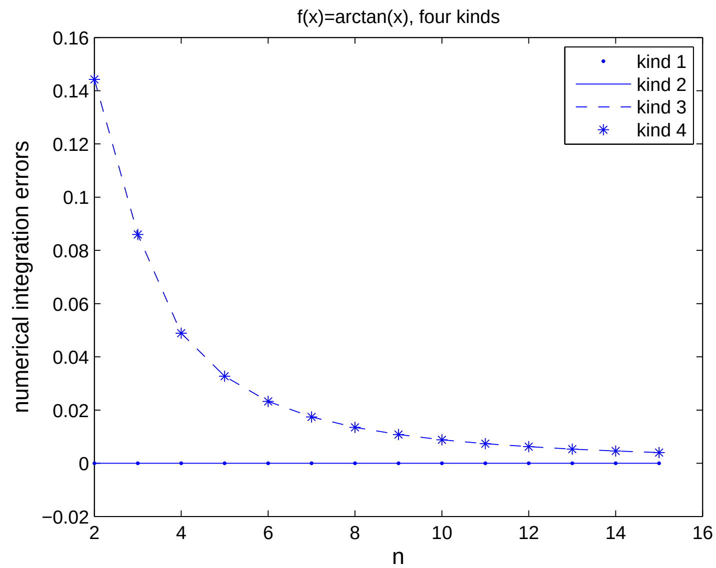

Figure 2.3: Numerical errors of kind 1,2,3 and 4 as $2 \leq n \leq 15$

n		kind 1	kind 2	kind 3	kind 4
100	error	9.4109e-017	5.6379e-018	1.0279e-004	1.0279e-004
	CPU time (seconds)	0.000432	0.001235	0.001139	0.001140
1000	error	2.6206e-017	6.8195e-017	1.0371e-006	1.0371e-006
	CPU time (seconds)	0.018542	0.022468	0.018278	0.018277
5000	error	7.7193e-017	3.7038e-017	4.1517e-008	4.1517e-008
	CPU time (seconds)	0.437004	0.428570	0.430376	0.436856
100,000	error	1.6587e-017	6.5385e-017	1.0381e-010	1.0381e-010
	CPU time (seconds)	234.593059	235.826457	235.251201	234.654195

Table 1: $f(x) = \sin(x)$

concrete errors arising from numerical integrations by applying the four kinds of formulae, as well as the corresponding CPU times elapsed for computing the numerical results, as an indication of the cost of the algorithm, are listed in the following Table 1, for $n = 100$, 1000, 5000, and 100000.

The data reflect the differences mentioned above. From the CPU times one can see that the cost of the four kinds of formulae are almost the same even

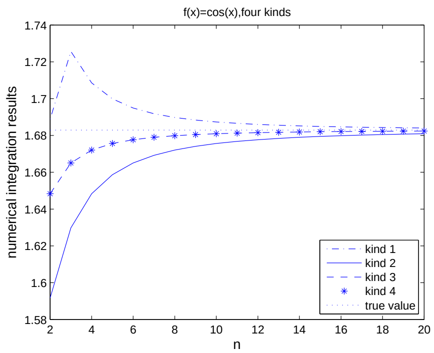

Figure 3.1: Numerical results of the four kinds as $2 \le n \le 20$

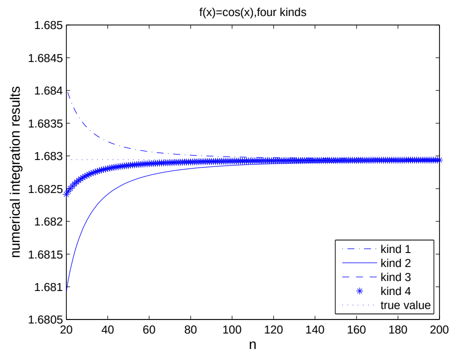

Figure 3.2: Numerical results of the four kinds as $20 \le n \le 200$

with respect to very large n, which is also stated theoretically in Theorem 1.

Example 2. $f(x) = \arctan(x)$. With the odd integrand $f(x) = \arctan(x)$ in this example, the four kinds of Gauss-Chebyshev quadrature formulae behave the same way as they do in the previous example, as expected.

Example 3. $f(x) = \cos(x)$. In this example the integrand $f(x) = \cos(x)$ is an even function. The exact integration result is $2\sin(1)$. Figure 3.1 and 3.2 show the evolution of the numerical integration results with the increasing of the number of nodes n by applying the four kinds of Gauss-Chebyshev quadrature formulae. Since as n is small, the errors are relatively large so that the values for larger n become almost invisible, we divide the graphs into two parts, i.e. Figure 3.1 ($2 \le n \le 20$) and Figure 3.2 ($20 \le n \le 200$). From them one can see that the numerical results produced by the third and fourth kind of formulae coincide for all n which confirms the corresponding statement in Theorem 3, and are much more accurate than the results of the first or second kind as $n \le 15$. However, as n gets larger, the difference becomes small very fast (e.g., as n=20, the difference in accuracy is in the magnitude of 10^{-3}). The error graph Figure 3.3 and the data in Table 2 give the same information. Again, the CPU times used for numerical integrations by the four kinds of formulae (2.7)-(2.10) are found to be almost of no difference.

Example 4. $f(x) = \ln(|x|)$. The two figures above give the evolution curves of the numerical integration results with respect to n by applying the four kinds of formulae (2.7)-(2.10) to integrating the even function $\ln(|x|)$. The zigzag in Figure 4.1 is caused by the existence of the node 0 in the first and second kind of nodes as n is odd, which makes the function value of $\ln(|x|)\sqrt{1-x^2}$ tend to minus infinity and thus the numerical results for odd n have great error, while

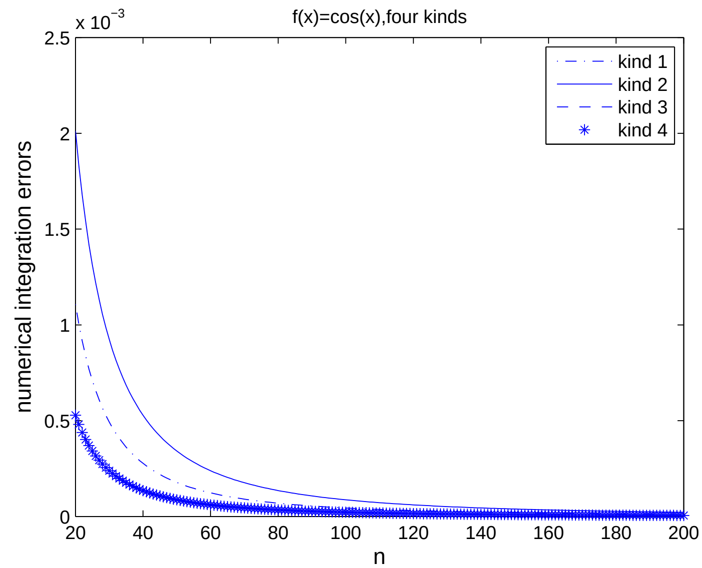

Figure 3.3: Numerical errors of the four kinds as $20 \leq n \leq 200$

n		kind 1	kind 2	kind 3	kind 4
100	error	4.4433e-005	8.7120e-005	2.1998e-005	2.1998e-005
	CPU time (seconds)	0.002118	0.002323	0.002230	0.002193
1000	error	4.4438e-007	8.8699e-007	2.2197e-007	2.2197e-007
	CPU time (seconds)	0.018050	0.018025	0.018021	0.018046
5000	error	1.7775e-008	3.5536e-008	8.8858e-009	8.8858e-009
	CPU time (seconds)	0.447193	0.444399	0.443713	0.444645
100,000	error	4.4449e-011	8.8869e-011	2.2205e-011	2.2187e-011
	CPU time (seconds)	230.853978	231.188065	231.582981	231.603952

Table 2: $f(x) = \cos(x)$

the results of the first and second kind of formulae for even n, as well as that of the third and fourth kind seem to coincide with the exact value, as shown in Figure 4.1. The third and fourth kind of nodes do not contain zero for all n, and have smooth curves. To compare the effectiveness of the four formulae, we thereafter choose only even number of nodes n to let the four formulae have comparable situations, to which Figure 4.2 is contributed. Again the third and

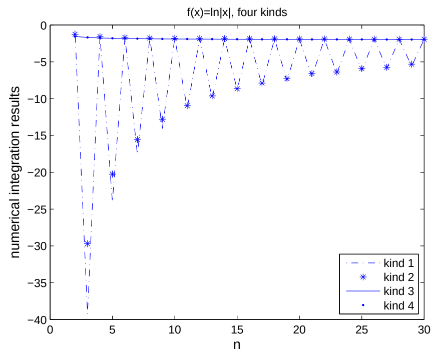

Figure 4.1: Numerical results of the four kinds as $2 \leq n \leq 30$

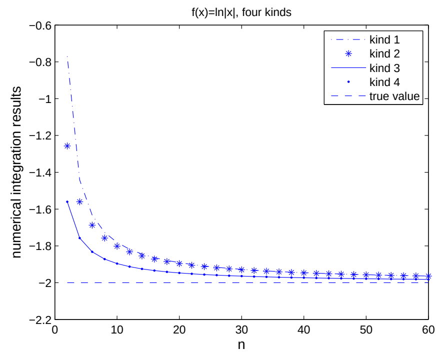

Figure 4.2: Numerical results of $2 \leq n\,(\text{even}) \leq 60$

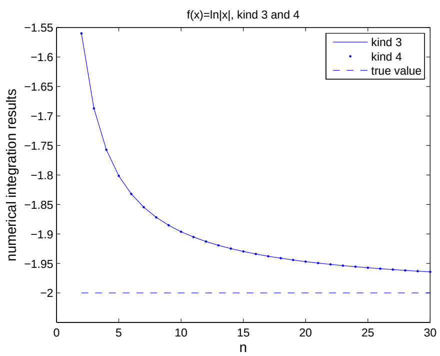

Figure 4.3: Numerical results of kind 3 and 4 as $2 \leq n \leq 30$

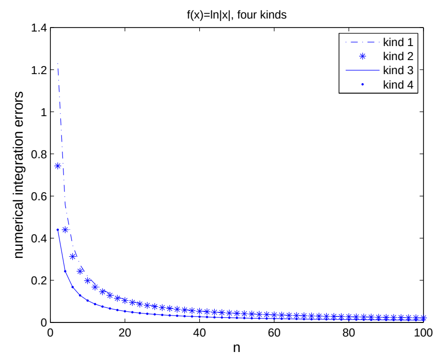

Figure 4.4: Numerical errors of the four kinds as $2 \leq n\,(\text{even}) \leq 100$

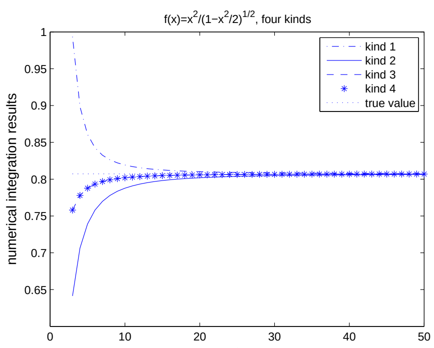

Figure 5.1: Numerical results of the four kinds as $3 \leq n \leq 50$

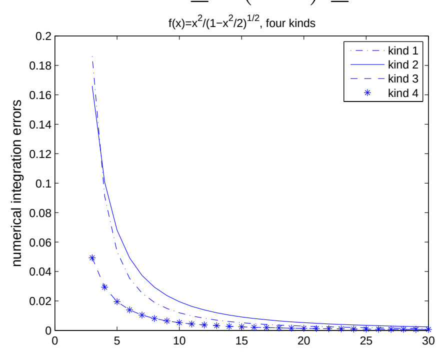

Figure 5.2: Numerical results of the four kinds as $3 \leq n \leq 30$

n		kind 1	kind 2	kind 3	kind 4
100	error	0.0218	0.0216	0.0108	0.0108
	CPU time (seconds)	0.002290	0.002397	0.002247	0.002208
1000	error	0.0022	0.0022	0.0011	0.0011
	CPU time (seconds)	0.036005	0.025837	0.035146	0.029839
5000	error	4.3552e-004	4.3543e-004	2.1774e-004	2.1774e-004
	CPU time (seconds)	0.434460	0.443664	0.442873	0.436336
100,000	error	2.1776e-005	2.1776e-005	1.0888e-005	1.0888e-005
	CPU time (seconds)	232.141441	232.267216	233.002551	232.045699

Table 3: $f(x) = \ln(|x|)$

fourth kind of formulae give the same results and are more accurate than the first or second kind, as concluded in Example 3.

Figure 4.3, 4.4 and Table 3 give information comporting with the conclusion above for even integrands (note that the values of n in Table 3 are all even).

Example 5. $f(x) = \frac{x^2}{\sqrt{1-\frac{1}{2}x^2}}$. For the even integrand $f(x) = \frac{x^2}{\sqrt{1-\frac{1}{2}x^2}}$, the four formulae behave in the same way as in Example 3 and 4. Based on these and more experiments we may come to the following empirical result.

Conclusion 5. *For a large number of even integrands $f(x)$, the third and fourth kind of Gauss-Chebyshev quadrature formulae (2.9)-(2.10), which result in the same numerical integration values, are more accurate than the first and second kind of formulae, especially when n is small (e.g., as $n \leq 15$).*

Example 6. $f(x) = \mathrm{e}^x$. The integrand $f(x) = e^x$ in this example is neither odd nor even. The exact integration value is $e - e^{-1}$. We may see from the numerical integration result in Figure 6.1 and the error in Figure 6.2 that the third kind of formula (2.9) gives the best numerical results, as is also indicated by the data in Table 4. And the CPU times for computation keep almost invariant under the transforms in (2.7)-(2.10).

Example 7. $f(x) = \frac{x^2}{1-\frac{1}{2}x^3}$. Numerical integrations of the rational function $f(x) = \frac{x^2}{1-\frac{1}{2}x^3}$ also give support to the rule discovered in the previous example, from which the following empirical conclusion may be obtained.

Conclusion 6. *For many integrands $f(x)$ which are neither odd nor even, the third kind of Gauss-Chebyshev quadrature formula (2.9) produces the most*

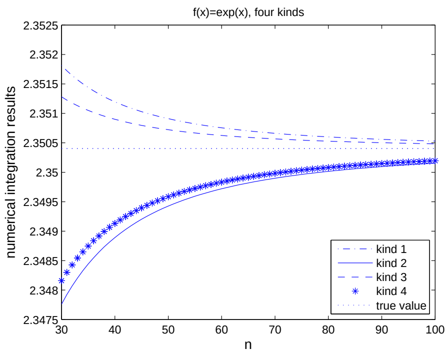

Figure 6.1: Numerical results of the four kinds as $30 \leq n \leq 100$

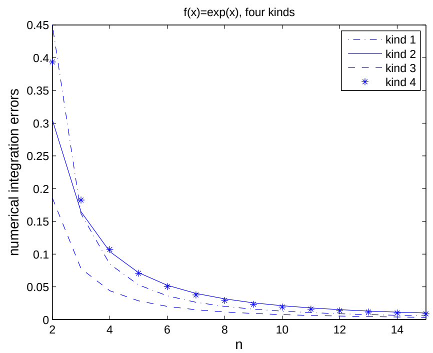

Figure 6.2: Numerical results of the four kinds as $2 \leq n \leq 15$

n		kind 1	kind 2	kind 3	kind 4
100	error	1.2693e-004	2.4884e-004	8.0732e-005	2.0639e-004
	CPU time (seconds)	0.002181	0.002392	0.002285	0.002250
1000	error	1.2691e-006	2.5332e-006	8.1447e-007	2.0823e-006
	CPU time (seconds)	0.018235	0.033795	0.018239	0.033872
5000	error	5.0765e-008	1.0149e-007	3.2605e-008	8.3360e-008
	CPU time (seconds)	0.441123	0.438668	0.440146	0.440597
100,000	error	1.2691e-010	2.5381e-010	8.1496e-011	2.0844e-010
	CPU time (seconds)	231.017699	231.076390	231.159574	231.209157

Table 4: $f(x) = \mathrm{e}^x$

accurate numerical integration results, especially as n is small (e.g., as $n \leq 15$).

References

[1] L. Boyadjiev, R. Scherer, On the Chebyshev polynomials, *Kuwait Journal of Science and Engineering*, **28** (2001), 227-240.

[2] W. Gautschi, *Remainder Estimates for Analytic Functions*, Numerical Integration, NATO ASI Series, Kluwer Academic Publishers, Netherlands (1992).

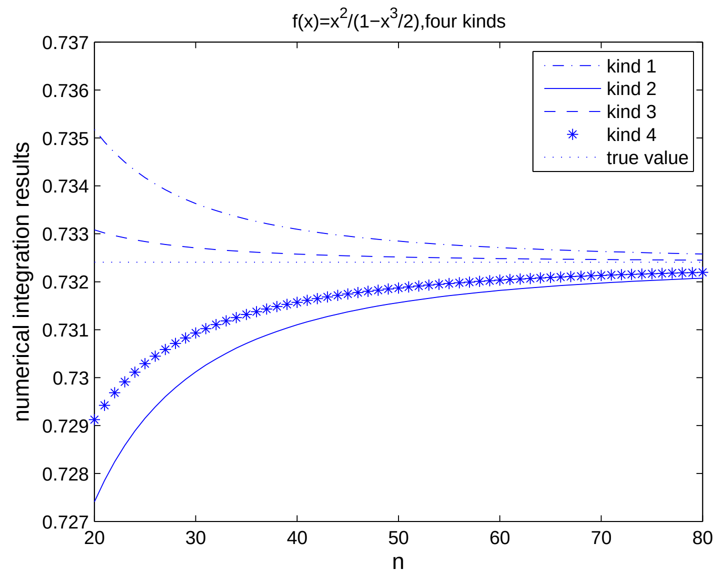

Figure 7.1: Numerical results of the four kinds as $20 \leq n \leq 80$

[3] W. Gautschi, R.S. Varga, Error bounds for Gaussian quadrature of analytic functions, *SIAM J. Numer. Anal.*, **20** (1983), 1170-1186.

[4] J.C. Mason, Chebyshev polynomials of the second, third and fourth kinds in approximation, indefinite integration, and integral transforms, *J. Comput. Appl. Math.*, **49** (1993), 169-178.

[5] J.C. Mason, D.C. Handscomb, *Chebyshev Polynomials*, CRC Press, New York (2002).

[6] T.J. Rivlin, *Chebyshev Polynomials*, Second Edition, John Wiley and Sons, Inc., New York (1990).

[7] R. Scherer, T. Schira, Estimating quadrature errors for analytic functions using kernel representations and biorthogonal systems, *Numer. Math.*, **84** (2000), 497-518.

[8] T. Schira, *Ableitungsfreie Fehierabschätzungen bei numerischer Integration holomorpher Funktionen*, Dissertation , Universität Karlsruhe (1994).

[9] T. Schira, The remainder term for analytic functions of Gauss-Lobatto quadratures, *J. Comput. Appl. Math.*, **76** (1996), 171-193.

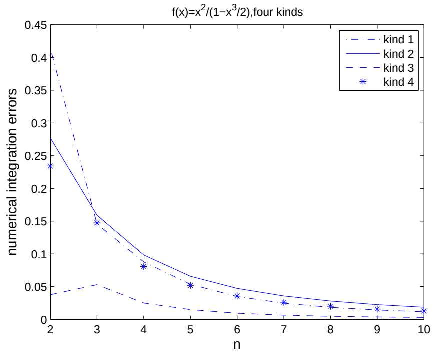

Figure 7.2: Numerical results of the four kinds as $2 \leq n \leq 10$

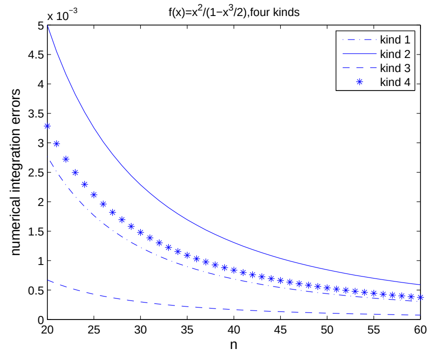

Figure 7.3: Numerical results of the four kinds as $20 \leq n \leq 60$

[10] T. Schira, The remainder term for analytic functions of symmetric Gaussian quadratures, *Math. Comput.*, **66** (1997), 297-310.

[11] G. Szegö, *Orthogonal Polynomials*, Fourth Edition, Amer. Math. Soc. Colleg. Publ., **23**, Amer. Math. Soc., Providence, RI (1975).