Numerical Integration and Optimal Quadrature Formulae

by

Gordon W. Thorgeirson

A Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

University of Manitoba

March, 1974

ACKNOWLEDGEMENTS

The author expresses his sincere gratitude to Professor W. D. Hoskins for the encouragement, guidance, and constructive criticism given by him throughout the preparation of this thesis. A special thanks is directed to Mrs. M. Hoeppner for her typing of the thesis.

Numerical Integration and Optimal Quadrature Rules

ABSTRACT

This thesis studies the subject of numerical integration with attention being focused on the concept of an optimal quadrature rule. Some characteristics considered to be desirable in a quadrature rule are presented and the classical quadrature rules which adhere to these properties are identified. The undesirable traits which often accompany these quadrature rules are also noted. A definition of an optimal quadrature rule is then formulated and an attempt is made to identify known quadrature rules which best conform to the proposed definition of optimality.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 Approximate integration of functions of a single variable

Restricting attention to Riemann-integrable functions of a single variable, quadrature becomes the process of evaluating numerically integrals of the form

$$I(w,f,a,b) = \int_a^b w(x)\ f(x)\ dx \tag{1.1}$$

where x is the independent variable, $w(x)$ is the weight function, a and b are the integration limits, and $w(x)\cdot f(x)$ is the integrand. The integral in (1.1) may be improper and/or indefinite and may or may not have an explicit solution obtainable by application of analytic methods. If an analytic solution cannot be found or easily determined, or in fact even when one is not desired, approximate methods for evaluating (1.1) must be used.

The most common methods for approximation of (1.1) use formulae which express the integral as a weighted sum of values of the integrand and its derivatives at a finite number of abscissae. In their most general form the approximations are

$$I(w,f,a,b) \approx \sum_{j=0}^{n} \sum_{k=0}^{m} a_{jk} f^{(k)}(x_{jk}) \tag{1.2}$$

with the set $\{a_{jk}\}$ being the assigned weights and the set $\{x_{jk}\}$ being the specified nodes. For fixed n and m, the weights and abscissae are considered to be parameters of (1.2) with the only restriction being that the abscissae are not to be located outside the interval of integration. In most cases, the quadrature weights are determined uniquely by the quadrature abscissae. When quadrature formulae of the form (1.2) are used to approximate (1.1) the process is sometimes called mechanical quadrature. Specific forms of (1.2) and the class of functions for which these quadrature rules are applicable are discussed in subsequent chapters.

Ideally a quadrature formula as given by (1.2) should be applicable to all integrable functions, be easily evaluated, and have a high degree of accuracy. The quadrature formulae developed to date fail to satisfy the above criteria with most being applicable to a restricted class of functions [37]. Often several methods are available for integrating functions of a particular type. In such instances someone choosing a quadrature formula is faced with the problem of determining which approximation should be used for the particular function or class of functions being considered. Hopefully the selected quadrature rule would be optimal in some sense for the given problem.

1.2 Historical development of quadrature techniques

Historically quadrature originates with the Greeks and their study of the problem of squaring the circle. In attempting to solve the problem, the method of exhaustion was devised whereby an assemblage of non-overlapping elementary figures is used to exhaust the area of the main figure. This method represents the first example of a limiting process for integration and is limited to approximating areas of sufficiently regular closed curves. It was first used by Antiphon (ca. 430 B.C.) and later applied to other plane curves by Archimedes (ca. 287 B.C.) [7]. In the ensuing centuries other geometrical techniques for quadrature were developed (cf. Montucla [36]).

The origins of analytical methods for integration are associated with Cavalieri (1598-1647) and the method of indivisibles [6]. Cavalieri's method constituted a rudimentary form of integral calculus and because of its non-rigorous nature, the method fell into disfavour. A variation of the method of indivisibles was preposed by Newton (1647-1727) in his method of fluxions. With Newton's method of differential calculus and Leibniz's notation, the present form of integral calculus has proved to be the most successful method for integration even though Riemann's definition of integration as a definite limiting process provides an intuitive approach to the problem.

The first numerical integration formulae originated with Newton when he used interpolation to effect quadrature of plane curves [25]. Using Newton's results, Cotes devised interpolatory quadrature formulae using up to eleven abscissae. About a century later Gauss used the properties of orthogonal polynomials to establish a new type of quadrature formula. Gauss's quadrature formulae, which are also interpolatory, increase the precision of the rule without increasing the number of quadrature nodes. Modifications to Gauss's method were later made by Lobatto and Radau. More recently, quadrature formulae for specific applications have been developed with Filon's method [17] being an example. Since the development of digital computers and high-level programming languages, interest in quadrature has been directed to developing "automatic integrators". These quadrature schemes are either adaptive iterative or non-iterative algorithms based on the classical quadrature rules and are meant to be used for integrating a broad class of functions. In their ideal form these automatic quadrature methods would have the integrand, integration limits, and error tolerance level serve as input and an approximation and associated error constitute the output.

1.3 Criteria for optimal quadrature formulae

It was mentioned previously that the quadrature weights and abscissae in (1.2) are parameters and that their specification determines a unique approximation to (1.1). The fact that there exist infinitely many choices for the weights and abscissae with a corresponding number of calculable approximations then raises the question as to how these parameters are to be determined. For whatever method is used to establish these parameters, a non-rigorous approach is unfeasible because no definite conclusions can be drawn as to the quality of the resulting approximation [14, p.160]. The only alternative then is to set forth criteria which characterize an optimal approximation for (1.1).

The need for the definition, classification, and development of optimal quadrature formulae is realized when the following facts are considered. First, it has been noted by Oliver [37] that for any quadrature formula, a pathological function can be found such that an application of the quadrature rule in question to such a function yields an unfavorable error in the resulting approximation. On that basis it can be said that certain types of quadrature rules are applicable to a restricted class of functions. Secondly it is to be noted that there exists an overlapping of areas for which different families of quadrature rules are applicable and that each method of quadrature has distinguishable advantages and disadvantages. For example, both Gaussian and Newton-Cotes quadrature rules can be used to integrate functions which have continuous high-order derivatives but each has differing intrinsic characteristics. Finally there is the observation that approximate integration is used in a wide range of disciplines with the people using the known quadrature rules having varying degrees of familiarity with the subject. As a consequence, it is quite common for someone who has access to a computer to use a standard library program without giving sufficient consideration as to the usefulness of the method for his particular problem. If a characterization of each type of quadrature rule is made available, then it would be easier for the users of quadrature rules to evaluate their selection of approximation. Additional benefits of such knowledge would be the increased confidence in the approximation and the ascertainment of whether the approximation is overcomputed or undercomputed.

The study of optimal quadrature formulae began only recently with the results of Sard's investigation of "Best approximate integration formulas" being the first publication on the subject. Since then people such as Davis, Meir and Sharma, Barnhill and others (see [15], [41], [30]) have considered the problem with some having different objectives in mind. Throughout all of this work, however, a consistent theme of imposing restrictions on the quadrature rule is evident with either minimal or maximal properties for the quadrature rule being sought. In some cases the specified restriction(s) define the quadrature rule uniquely.

For the majority of suggested qualities of an optimal quadrature formulae, attention is focused on the truncation error $T_{n,m}$ which is related to the integral approximation to the extent that

$$\int_a^b w(x)\, f(x)dx = \sum_{j=0}^{n} \sum_{k=0}^{m} a_{jk} f^{(k)}(x_{jk}) + T_{n,m}(f). \quad (1.3)$$

Its dependence on the integral and quadrature formula's parameters is realized in its integral representation as determined by Peano's theorem [12, p.70]. In a more abstract sense $T_{n,m}$ is considered as a linear functional with the integral sign being an operator defined over a function space. In other instances, consideration is given to the roundoff error which is associated with the calculattion of the sum in (1.2). Specifically, for the family of functions F for which the approximation (1.2) is to be applied, an optimal quadrature formulae has been characterized as one in which a choice of abscissae and/or weights results in either: (i) the truncation error being proportional to some order of the derivative of f on the interval $[a,b]$ (i.e. $T_{n,m}(f)=K\cdot f^{(n)}(\xi)$ for the largest possible integer n and for some ξ, $a<\xi<b$), (ii) the quadrature rule having the least square roundoff error, (iii) the quadrature rule providing the least upper bound truncation error, (iv) the truncation error having

a minimum norm,(v)the quadrature rule being best in the sense of Sard (viz. minimization of the square integral of the Peano kernel function) [12,p.76], or (vi)the sum of the absolute values of the quadrature weights being as small as possible. The characterization (i) is usually referred to as the precision of the quadrature rule with the quadrature rule (1.2) having a precision of degree n if for some constant K, $T_{n,m}(f)=K\cdot f^{(n+1)}(\xi)$.

1.4 <u>An approach to the subject of quadrature</u>

In view of the foregoing discussion on the significance of a comparative examination of quadrature rules, one of the objectives of this study is to present an overview of the subject of quadrature based on the notion that the criteria for an optimal approximation define a specific quadrature rule. Secondly, in realizing the importance of the digital computer as an aid in numerical integration, consideration is given to some of the problems associated with the design of automatic integrators.

The extent to which the above objectives are fulfilled is established on the basis of the following methodology. First a criterion for optimality is introduced and the underlying mathematical theory is discussed. Then the derived quadrature rules are identified with known quadrature formulae. Possible variations and extensions of the basic type of quadrature are investigated and their peculiarities discussed. Finally, some "successful" computer algorithms are presented and their computational limitations noted.

CHAPTER II

Constrained Quadrature Rules with Maximum Precision

2.1 Introduction

It was mentioned in the last chapter that a defining characteristic of an optimal quadrature approximation is the extent to which the quadrature rule integrates functions exactly. This quality of exactness is referred to as the precision of the quadrature rule with one quadrature rule being said to have a higher degree of precision if it can integrate exactly a larger class of functions. In order to generate quadrature formulae which have some measure of precision it is common practice to choose a set of basis functions which generate a function space and force the quadrature rule to integrate these functions exactly. Usually, the function space considered is the space of polynomials for which a quadrature rule is said to have a precision of degree n if it integrates all polynomials of degree $\leq n$.

In considering the actual use of approximations of the form (1.2) there exist two distinct possibilities, namely; (i) the function being integrated is known analytically and (ii) only values of the function $f(x)$ and its derivatives are given for a specified number of abscissae. This chapter concentrates on the development of quadrature rules which have maximum precision subject to the restriction imposed by situtations such as in case (ii). Initially, emphasis is placed on those quadrature rules which do not involve derivatives of $f(x)$. The above considerations then result in an approximation of the form

$$\int_a^b w(x)f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j) \qquad (2.1)$$

with the abscissae $a \le x_0 < x_1 < \ldots < x_n \le b$ being fixed and the $n+1$ quadrature weights a_j $(j = 0, 1, \ldots, n)$ being variable. To determine quadrature approximations of the form (2.1) having maximum precision it is necessary to choose the weights so that

$$\int_a^b w(x)f(x)dx = \sum_{j=0}^{n} a_j f(x_j) + T_n(f) \qquad (2.2)$$

with $T_n(f) = 0$ for as large a class of functions as possible.

2.2 Derivation of quadrature rules with maximum precision

Because there are $n+1$ degrees of freedom associated with the quadrature sum (2.1), it is possible to impose a corresponding number of conditions on the quadrature rule so that both the quadrature weights are determined uniquely and the quadrature rule is of maximum precision. Indeed, if a set of $n+1$ linearly independent functions $\{g_0(x), g_1(x), \ldots, g_n(x)\}$ are chosen as a basis for a function subspace and the quantities

$$\mu_k = \int_a^b w(x)g_k(x)dx \quad k = 0, 1, \ldots, n \qquad (2.3)$$

are calculated then the conditions $T_n(g_k) = 0$ $(k = 0, 1, \ldots, n)$ for (2.2) establish a system of $n+1$ equations

$$\sum_{j=0}^{n} a_j g_k(x_j) = \mu_k \quad k = 0, 1, \ldots, n \qquad (2.4)$$

for the unknown quadrature weights. By writing the system (2.4) as a matrix equation $G\underline{a} = \underline{\mu}$ with $G = (g_{ij})=(g_i(x_j)), \underline{a}^T=(a_0,a_1,\ldots,a_n)$ and $\underline{\mu}^T=(\mu_0,\mu_1,\ldots,\mu_n)$ the independence of the basis functions guarantees a unique non-trivial solution. Moreover, for any function f which is an element of the subspace spanned by the basis functions, we have $T_n(f) = 0$. The fact that an enforcement of a larger number of conditions on (2.2) will generate an inconsistent system of equations determines the maximum degree of precision for (2.2) to be n.

Some possible bases for derivation of quadrature rules of maximum precision are the set of monomials $\{1, x, x^2, \ldots, x^n\}$, the set of exponentials $\{1, e^x, e^{2x}, \ldots, e^{nx}\}$, and the sets of trigonometric functions $\{1, \sin x, \ldots, \sin nx\}$, $\{1, \cos x, \ldots, \cos nx\}$, etc. For each case, a particular quadrature rule will be established with its precision being defined in terms of the chosen basis. When the monomials up to degree n are chosen, there is a correspondence with the theory of interpolation and some common quadrature rules.

If a quadrature rule of the form (2.2) is considered within the limitations mentioned in sec. 2.1 then it can be shown that the quadrature rule with maximum precision over the set P_n of all polynomials of degree $\leq n$ is interpolatory. In fact the following theorem [29, p.80] is true.

<u>Theorem 2.1.</u> The quadrature rule given in (2.2) is interpolatory if and only if it is exact for all ploynomials of degree $\leq n$.

Although the above theorem shows that quadrature formulae of precision n are interpolatory it does not assert that such formulae have the highest precision. However, the quadrature formula (2.2) has a degree of precision of at most n, for if it were exact for polynomials of degree $n+1$, then it would contradict the interpolation theorem given by Cheney [9] which states that the interpolating function is unique. Therefore, for quadrature formulae based on $n+1$ nodes, the highest degree of precision that can be obtained is n.

Since it is assumed that no function derivatives are to appear in the quadrature approximation, the interpolatory quadrature rules must be based on a Lagrange interpolation of the integrand. If the quadrature nodes $x_0, x_1, \ldots, x_n$ are prescribed then the integrand $f(x)$ is approximated by a collocating polynomial such that

$$f(x) = \sum_{k=0}^{n} \ell_k(x)f(x_k) + r_n(x) \quad , \quad \ell_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{(x-x_j)}{(x_k-x_j)} \qquad (2.5)$$

with $r_n(x)$ being the truncation error associated with the interpolation. Now multiplying both sides of (2.5) by the weight function $w(x)$ and integrating over the interval $[a,b]$, a quadrature formulae such as (2.2) is obtained with

$$a_j = \int_a^b w(x)\ell_j(x)dx \qquad j = 0(1)n$$

and

$$T_n = \int_a^b w(x)r_n(x)dx \quad . \qquad (2.6)$$

From Weierstrass's theorem on polynomial approximation we know that the accuracy of the integrand approximation and hence quadrature approximation usually improves as the number of nodes increases and in any case is exact for all functions $f(x)$ which are polynomials of degree n or less.

The calculation of the quadrature weights $a_k (k = 0, 1, \ldots, n)$ by means of equation (2.6) is unsuitable because the addition of new points to the quadrature formula necessitates the re-evaluation of all the weights. A more appropriate method is to use divided differences. By using Newton's divided-difference interpolation formula (which is equivalent to Lagrange interpolation) it is possible to compute the weights only once even though points may be appended later to the quadrature formula [23]. Such a scheme is advantageous in developing an algorithm for adaptive quadrature using interpolation formulae.

2.3 Newton-Cotes quadrature formuale

Interpolatory quadrature formulae are applicable to both uniformly and non-uniformly distributed nodes in the interval $[a,b]$. However, convenient quadrature rules are obtained when the nodes are equidistant $\left(\text{i.e. } x_i - x_{i-1} = h,\ i=1(1)n,\ h = \frac{x_n - x_0}{n}\right)$ and in such cases the quadrature formulae are classifed as being of the Newton-Cotes type. The Newton-Cotes formulae may use both endpoints, in which case they are called closed formulae, or they may have all nodes in the interior of the interval $[a,b]$ in which case they are referred to as being open formulae. Other quadrature formulae using only one of the integration limits are also possible. In the derivation of these quadrature formulae, the weight function $w(x)$ is assumed to be a constant function.

For the case of a constant weight function, the weights for the Newton-Cotes formulae are called Cotesian numbers and the methods used in obtaining their values are varied [25]. One approach, however, leads to an asymptotic representation for large values of n. By a change of notation for (2.2), we can write for the closed Newton-Cotes rules,

$$\int_a^b f(x)dx = \frac{(b-a)}{n} \sum_{k=0}^{n} a_{nk} f(x_0+kh) + T_n(f) \qquad (2.7)$$

where

$$a_{nk} = \frac{n}{(b-a)} \int_a^b \frac{p_n(x)}{(x-x_0-kh)p_n'(x_0+kh)} dx \qquad (2.8)$$

and $$p_n(x) = \prod_{j=0}^{n} (x-x_0-jh) \quad .$$

Then substituting $x = x_0 + th \ (0 \le t \le n)$ we get

$$a_{nk} = \frac{(-1)^{n-k}}{k!(n-k)!} \int_0^n t(t-1) \ldots (t-k+1)(t-k-1)\ldots(t-n)dt \qquad (2.9)$$

$$= \frac{(-1)^k}{k!(n-k)!} \int_0^n \frac{\Gamma(t+1)\Gamma(n+1-t)\sin \pi t}{\pi\cdot(t-k)} dt \qquad (2.10)$$

with $\Gamma(z)$ being the Gamma function. Now letting the integral in (2.10) be a sum of three integrals (with limits $0 \le t \le 3$, $3 \le t \le n-3$, and $n-3 \le t \le n$ respectively), and estimating each integral for large values of n, we obtain [29, p.86]

$$a_{nk} = \begin{cases} \dfrac{1}{\log n}\left[1+O\left(\dfrac{1}{\log n}\right)\right] & k = 0, n \\ \dfrac{(-1)^{k-1}}{[\log n]^2} \dbinom{n}{k} \left[\dfrac{1}{k} + \dfrac{(-1)^n}{n-k}\right]\left[1+O\left(\dfrac{1}{\log n}\right)\right] & 1 \le k \le n-1 \end{cases} \quad . \qquad (2.11)$$

Hence by using equation (2.9), the coefficients a_{nk} can be computed and their limiting values estimated by equation (2.11). The computed weights for small values of $n(n = 1(1)\ 10)$ are given in table 2.1.

On examining equations (2.9) and (2.11), several characteristics for the Cotes numbers are evident. The weights are symmetric with $a_{nk} = a_{n(n-k)}$ for $0 \le k \le \left[\frac{n}{2}\right]$. Also all the weights are rational numbers with the denominator increasing non-uniformly as n increases and the numerator increasing as k approaches $\left[\frac{n}{2}\right]$. A check of the weights given in table 2.1 shows that all weights are positive for odd values of n less than eleven and even values of n less than eight. For large n, there exist both positive and negative weights which exceed in absolute value any arbitrarily large number. Moreover, the first two weights a_{n1} and a_{n2} are always positive with each successive weight up to $a_{n\left[\frac{n}{2}\right]}$ alternating in sign. Finally, in view of the fact that the Newton-Cotes rules are interpolatory they must integrate a constant function exactly and so we must have $\sum_{k=0}^{n} a_{nk} = n$.

The preceding characteristics of the quadrature weights apply only to those interpolatory quadrature rules having a constant weight function. Since it is usually quite difficult to calculate the weights associated with non-constant weight functions, someone who is intent on using this type of quadrature will, if necessary, modify the integrand in (2.2) to change the problem to one of integrating with respect to a constant weight function. In some instances such an approach will have harmful effects. It has been shown by Kaplan [27] that whenever there is a singularity in the integrand,

$$\int_a^b f(x)dx = H \sum_{k=0}^{n} a_{nk}\, f(a+kh) + T_n(f)$$

# of intervals n	coefficient H	$a_{0,n}$ $a_{n,n}$	$a_{n,1}$ $a_{n,n-1}$	$a_{n,2}$ $a_{n,n-2}$	$a_{n,3}$ $a_{n,n-3}$	$a_{n,4}$ $a_{n,n-4}$	$a_{n,5}$ $a_{n,n-5}$	Quadrature Error $T_n(f)=Cf^{(p)}(\xi)$, $a<\xi<b$
1	$\frac{h}{2}$	1						$-\frac{h^3}{12} f^{(2)}(\xi)$
2	$\frac{h}{3}$	1	4					$-\frac{h^5}{90} f^{(4)}(\xi)$
3	$\frac{3h}{8}$	1	3					$-\frac{3h^5}{80} f^{(4)}(\xi)$
4	$\frac{2h}{45}$	7	32	12				$-\frac{8h^7}{945} f^{(6)}(\xi)$
5	$\frac{5h}{288}$	19	75	50				$-\frac{275h^7}{12096} f^{(6)}(\xi)$
6	$\frac{h}{140}$	41	216	27	272			$-\frac{9h^9}{1400} f^{(8)}(\xi)$
7	$\frac{7h}{17280}$	751	3577	1323	2989			$-\frac{8183h^9}{518400} f^{(8)}(\xi)$
8	$\frac{4h}{14175}$	989	5888	-928	10496	-4540		$-\frac{2368h^{11}}{467775} f^{(10)}(\xi)$
9	$\frac{9h}{89600}$	2857	15741	1080	19344	5778		$-\frac{173h^{11}}{14620} f^{(10)}(\xi)$
10	$\frac{5h}{299376}$	16067	106300	-48525	272400	-260550	427368	$-\frac{1346350h^{13}}{326918592} f^{(12)}(\xi)$

Table 2.1 Low-order closed Newton-Cotes quadrature formulae

those quadrature rules based on weight functions of the form $w(x)=x^p$ $(p\neq 0)$ produce better approximations than those obtained from the conventional Newton-Cotes formulae.

Also, it is to be noted that equations (2.7) and (2.9) apply only to closed Newton-Cotes quadrature formulae. Calculation of the weights for open-type formulae requires the changes $x_0=a+h$, $x_n=b-h$, and $h = \frac{b-a}{n+2}$ so that the range of integration in (2.9) becomes $-1 \leq t \leq n+1$. The quadrature rule is now of the form

$$\int_a^b f(x)dx = \frac{(b-a)}{n+2} \sum_{k=0}^{n} a_{nk} f\left(a+ \left(\frac{k+1}{n+2}\right)(b-a)\right) + T_n(f) \qquad (2.12)$$

with $$a_{nk} = \frac{(-1)^{n-k}}{k!(n-k)!} \int_{-1}^{n+1} t(t-1)\ldots(t-k+1)\ldots(t-n)dt \quad k=0,1,\ldots,n$$

and $$T_n(f) = \int_a^b r_n(x)dx \quad .$$

In view of the fact that a different interpolating polynomial is used to obtain (2.12), the truncation error associated with open Newton-Cotes formulae will not be the same as that obtained for the closed-type quadrature rules. A comparison of both open and closed type rules using an equivalent number of abscissae will show that the latter invariably has a smaller quadrature error. This drawback has resulted in the infrequent use of the open-type formulae in general integration problems. Nevertheless, open-type formulae have been used successfully as predictors in predictor-corrector methods associated with obtaining numerical solutions of ordinary differential equations [20].

2.4 Representation of quadrature errors for Newton-Cotes formulae

For the general interpolatory quadrature rule given by (2.2) and (2.6) there is an error $r_n(x)$ associated with the interpolation of the integrand $f(x)$. By using divided differences a representation [29, p.44]

$$r_n(x) = p_n(x)f[x_0,x_1,\ldots,x_n,x] \tag{2.13}$$

can be derived where $f[x_0,x_1,\ldots,x_n,x]$ represents the divided difference for $f(x)$ over the set of points $\{x_0,x_1,\ldots,x_n\}$ and $p_n(x) = \prod_{i=0}^{n} (x-x_i)$. The subsequent integration

$$T_n(f) = \int_a^b w(x)r_n(x)dx = \int_a^b w(x)p_n(x)f[x_0,x_1,\ldots,x_n,x]dx \tag{2.14}$$

yields the truncation error of the associated quadrature rule. To derive more appropriate representations for $T_n(f)$ it is convenient (and sometimes necessary) to make the following assumptions. First, let it be assumed that $f\epsilon C^{(n+2)}[a,b]$ where $C^{(n+2)}[a,b]$ denotes the class of all functions having continuous derivatives of order n+2 or less on the interval [a,b]. Secondly, suppose that the weight function w(x) is constant over the entire interval [a,b]. On the basis of these assumptions, the following theorem is true.

<u>Theorem 2.2</u>

For the closed Newton-Cotes quadrature formulae having an even number of equal intervals on [a,b], the error $T_n(f)$ can be expressed by

$$T_n(f) = \frac{K_n}{(n+2)!} f^{(n+2)}(\eta), \qquad a < \eta < b \tag{2.15}$$

where $\qquad K_n = \int_a^b x\, p_n(x)dx$

Proof: In the derivation of (2.15) some preliminary results are required. First consider the function $g_n(x) = \int_a^x p_n(\xi)d\xi$ where $p_n(x) = (x-x_0)\ldots(x-x_n)$. By definition $g_n(a) = 0$ and because n is even, $p_n(\xi)$ is antisymmetric about the midpoint of the interval $[a,b]$ (i.e. $p_n\left(\frac{x_0+x_n}{2} + \xi\right) = -p_n\left(\frac{x_0+x_n}{2} - \xi\right)\Big)$. Hence we also have $g_n(b) = 0$. Also, because $x_0, x_1, \ldots, x_n$ are the only zeros of $p_n(x)$ we have $p_n(x) > 0$ for $x_0 < x < x_1$ so that $g_n(x)$ does not change sign for $x_0 \le x \le x_1$. But $|p_n(\xi+h)| < |p_n(\xi)|$ for $x_0 < \xi + h < \frac{x_0 + x_n}{2}$, $\xi \ne x_j$ $(j=0,1,\ldots,n)$ and consequently $g_n(x) > 0$ for $x_0 < x < \frac{x_0 + x_n}{2}$. By the antisymmetric property of $p_n(x)$ we also have $g_n(x) > 0$ for $\frac{x_0 + x_n}{2} < x < x_n$. Therefore $g(x)$ is of constant sign in the interval $[a,b]$.

Now considering the second integral in (2.14) we have

$$T_n(f) = \int_a^b \frac{dg_n(x)}{dx} f[x_0,x_1,\ldots,x_n,x]dx \tag{2.16}$$

It can be shown [29, p.41] that $\frac{d}{dx} f[x_0,x_1,\ldots,x_n,x]$ is continuous and

$\frac{f^{(n+2)}(\xi(x))}{(n+2)!} = \frac{d}{dx} f[x_0, \ldots, x_n, x]$ with $\xi(x)$ being dependent on x and $a<\xi(x)<b$. Hence by integrating (2.16) by parts once and using the fact that $g(a) = g(b) = 0$ we obtain

$$T_n(f) = -\int_a^b g_n(x) \frac{f^{(n+2)}(\xi(x))}{(n+2)!} dx \ . \qquad (2.17)$$

By assumption, $f^{(n+2)}(x)$ is continuous on [a,b] and as earlier proved $g_n(x)$ does not change sign on the interval [a,b]. An application of the second mean value theorem for integrals now gives

$$T_n(f) = -\frac{f^{(n+2)}(\eta)}{(n+2)!} \int_a^b g_n(x)dx, \quad a < \eta < b \ . \qquad (2.18)$$

Finally, noting the properties of $g_n(x)$ and integrating (2.18) by parts we get

$$\int_a^b g_n(x)dx = xg_n(x)\Big|_a^b - \int_a^b x \frac{d}{dx} g_n(x)dx = -\int_a^b x\, p_n(x)dx.$$

The theorem is proved by letting $K_n = -\int_a^b x\, p_n(x)dx$.

The above theorem gives the following corollary.

Corollary 2.1

For the closed Newton-Cotes formula defined by (2.2) and (2.6), the error for the case of an even number n of intervals is given by

$$T_n(f) = \frac{K_n h^{n+3}}{(n+2)!} f^{(n+2)}(\eta) \ , \quad h = \frac{b-a}{n} \ , \ a < \eta < b \qquad (2.19)$$

where $K_n = \int_0^n t(t-1)\ldots(t-n)dt$

Proof: In this case $w(x) \equiv 1$ and the fact that $g_n(x) > 0$ on $[a,b]$ gives $K_n = -\int_a^b g_n(x)dx < 0$. Then making the substitution $x = x_0 + th \; (0 \le t \le n)$ gives (2.19).

By following the same procedure as given in theorem 2.2 and corollary 2.1, the following theorem can be shown to be true.

Theorem 2.3

If $f(x)$ has a continuous derivative of order $n+1$ on $[a,b]$ and the number n of equal intervals is odd then the closed Newton-Cotes quadrature formula defined by (2.2) and (2.6) has an error of the form

$$T_n(f) = \frac{K_n h^{n+2}}{(n+1)!} f^{(n+1)}(\eta), \quad a < \eta < b, \qquad h = \frac{b-a}{n} \tag{2.20}$$

where $$K_n = \int_0^n (t-1)(t-2)\ldots(t-n)dt < 0$$

In the case of open-type Newton-Cotes quadrature formulae the errors are

$$T_n(f) = \frac{M_n h^{n+1}}{(n+2)!} f^{(n+2)}(\xi) \qquad a < \xi < b \tag{2.21}$$

$$M_n = \int_{-1}^{n+1} t(t-1)\ldots(t-n)dt > 0$$

when n is even and

$$T_n(f) = \frac{M_n\ h^{n+2}}{(n+1)!} f^{(n+1)}(\xi) \ , \quad a < \xi < b \qquad (2.22)$$

$$M_n = \int_{-1}^{n+1} (t-1)(t-2)\ldots(t-n)dt > 0$$

when n is odd.

2.5 Composite Newton-Cotes rules

For Newton-Cotes rules based on a large number of abscissae, the characteristic high degree of precision is offset by the presence of non-uniform and negative weights which induce a possibly significant roundoff error in the quadrature approximation. In contrast, the truncation error associated with the low-order rules restricts their use to problems requiring only a limited degree of accuracy. The problem of overcoming these limitations has been resolved by the development of composite Newton-Cotes rules which maximize the desirable characteristics and minimize the undesirable qualities of both the low-order and high-order formulae. The techniques used to generate such formulae usually involve taking linear combinations of the simple Newton-Cotes rules and/or adding a high-order difference to one of them so that the leading term of the truncation error is eliminated or else some weights vanish.

For example, taking $\frac{3}{40}$ times the Newton-Cotes formula for $n=2$ and $-\frac{1}{30}$ times the rule for $n=3$ and subsequently adding them together, we get Weddle's rule [39] where

$$\int_a^b f(x)dx = \frac{(b-a)}{20}(f_0+5f_1+f_2+6f_3+f_4+5f_5+f_6) - \frac{h^7}{1400}(10f^{(6)}(\eta_1)+9h^2f^{(8)}(\eta_2)) \quad (2.23)$$

$a < \eta_1, \eta_2 < b$ and $f_i = f(a + ih)$, $(i = 0, 1, \ldots, 6;\ h = \frac{(b-a)}{6})$.

This rule has a higher degree of precision than do both Simpson's three-eighths and one-third rule. Also Weddle's rule has simple coefficients which makes it usful for integrating functions when hand calculations are involved. Another notable quadrature formula is Hardy's rule

$$\int_a^b f(x)dx = \frac{b-a}{300}\,(14f_0+81f_1+110f_3+81f_5+14f_6)+\frac{9h^7}{1400}(2f^{(6)}(\eta_1)-h^2f^{(8)}(\eta_2)) \quad (2.24)$$

which is obtained by adding the difference $-\frac{(b-a)}{300}\Delta^6 f_0$ to Weddle's rule. By adding this difference, the number of function evaluations has decreased to five while the truncation error remains of the same order of magnitude. Other modified Newton-Cotes formulae of both the open and closed type can be derived by these methods. In general, for $N=(2j+1)(2j+2)$ nodes the quadrature rule obtained by eliminating the leading term of the truncation error between the pairs of Newton-Cotes formulae using $2j+1$ and $2j+2$ terms represents a family of quadrature formulae of precision $N-2j$. They have less precision than the corresponding N^{th} order Newton-Cotes formulae but their particular structure makes them more suitable for calculating quadrature approximations.

Another class of quadrature formulae based on the simple Newton-Cotes formulae are the so-called compound rules. If we consider a Newton-Cotes closed formula using n nodes over the interval $[a,b]$ then the resulting compound rule will be that formula obtained by applying the simple rule to m $(m>1)$ equal subintervals.

In total there will be $mn+1$ nodes and mn equal intervals partitioning $[a,b]$. The most commonly used compound rules are

$$\int_a^b f(x)dx = h\left[\frac{f(a)}{2} + \sum_{j=1}^{m-1} f(a+jh) + \frac{f(b)}{2}\right] - \frac{(b-a)^3}{12m^2} f^{(2)}(\xi) \qquad (2.25)$$

$$a < \xi < b, \quad h = \frac{b-a}{m}$$

which is based on the trapezoidal rule and

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a)+4\sum_{j=0}^{m-1} f(a+(2j+1)h)+2\sum_{j=1}^{m-1} f(a+2jh)+f(b)\right] - \frac{(b-a)^5}{90(2m)^5} f^{(4)}(\xi) \qquad (2.26)$$

$$a < \xi < b, \; h = \frac{b-a}{2m}$$

which is based on Simpson's one-third rule. For the case of equation (2.25) the truncation error decreases at a rate of proportional to $\frac{1}{m^2}$ and for (2.26) the error decreases in a way proportional to $\frac{1}{(2m)^5}$. Generally, a compound rule using m subintervals and based on a Newton-Cotes formula having $n+1$ nodes has an error of the form [24]

$$T_{m,n+1}(f) = \begin{cases} \dfrac{(b-a)M_n}{(n+2)!\,n^{n+3}} H^{n+2} f^{(n+2)}(\xi) & n\text{:even}, \; H = \dfrac{b-a}{nm} \\[2ex] \dfrac{(b-a)M_n}{(n+1)!\,n^{n+2}} H^{n+1} f^{(n+1)}(\xi) & n\text{:odd}, \; H = \dfrac{b-a}{nm} \end{cases} \qquad (2.27)$$

The convergence of the compound rules is guaranteed whenever the integrand is a bounded, Riemann-integrable function and the simple quadrature rule integrates the constant function exactly [14, p.25]. The main advantage of these composite rules is that a high degree of accuracy can be obtained for integrals even though the integrand may not have continuous high order derivatives.

An extension of the procedure used for generating compound rules yields another family of quadrature formulae which are based on the Newton-Cotes formulae. Starting with an initial sequence of approximations based on a simple Newton-Cotes formula, the Richardson extrapolation procedure is used to generate a sequence of higher order formulae. In the limit, each sequence of approximations for a specific order p converge and the sequence obtained by considering only the r^{th} $(r = 0, 1, \ldots, p)$ approximation of each order also converges.

Historically, a recursive algorithm for generating the sequences of formulae based on a Newton-Cotes rule was first devised by Romberg [14, p.166]. Starting with a sequence of approximations

$$T_{k,0} = h\left(\frac{f(0)}{2} + \sum_{j=1}^{2^k-1} f(a+jh) + \frac{f(1)}{2}\right), \quad h = 2^{-k} \qquad (2.28)$$

for the integral $I = \int_0^1 f(x)dx$, successive higher order approximations are obtained from the formula

$$T_{k,m} = \frac{4^m T_{k+1,m-1} - T_{k,m-1}}{4^m - 1} \quad . \qquad (2.29)$$

When $m=1$ and $m=2$ the resulting quadrature approximations correspond to the compound closed Newton-Cotes rules for $n=3$ and $n=4$ nodes respectively. The above procedure generates a sequence of quadrature approximations which may be arranged in a triangular array

$$
\begin{array}{llll}
T_{0,0} & & & \\
T_{0,1} & T_{1,0} & & \\
\vdots & \vdots & \ddots & \\
T_{0,m} & T_{1,m} & \cdots & T_{k,m}
\end{array}
$$

The principle error expression for $T_{k,m}$ increases its power of 2 each time we go one column to the right while h itself halves each row downward. If $f(x)$ is a bounded, Riemann-integrable function on $[0,1]$ such that $f \in C^{2m+2}[0,1]$, then each column and each diagonal of the given array converges to I and for any approximation $T_{k,m}$

$$
|T_{k,m} - I| \leq \left| \frac{B_{2m+2}}{(2m+2)!\,2^{(2k+m)(m+2)}} f^{(2m+2)}(\xi) \right| \qquad 0 \leq \xi \leq 1 \tag{2.30}
$$

where B_{2m+2} is the $(2m+2)^{th}$ Bernoulli number satisfying

$$
1 + \binom{n}{1} B_1 + \binom{n}{2} B_2 + \dots + \binom{n}{n-1} B_{n-1} = 0 \qquad n > 1,\ B_1 = -1/2 \tag{2.31}
$$

Other sequences of approximations have been obtained [14, p.167] when the initial approximations are of the form

$$T_{k,0} = h_k \left[\frac{f(0)}{2} + \sum_{j=1}^{n_k-1} f(jh_k) + \frac{f(1)}{2} \right] \tag{2.32}$$

$$h_k = \frac{1}{n_k} , \quad \frac{n_{k+1}}{n_k} > 1 .$$

More recently, however, modifications to Romberg's method have been considered whereby the rate of convergence is accelerated with the use of high-order Newton-Cotes formulae for the initial approximations. Meir and Sharma [33] show that for the case of initial approximations $S_{k,0}$ based on the closed Newton-Cotes formulae for $n=3$ nodes , i.e.

$$S_{k,0} = \frac{h}{3} \left[f(0) + 4 \sum_{j=0}^{2^k-1} f((2j+1)h) + 2 \sum_{j=1}^{2^k-1} f(2jh) + f(1) \right] \tag{2.33}$$

$$h = 2^{-k-1}$$

then the error in the approximation $S_{k,m}$ is

$$|S_{k,m} - I| \le \left| \frac{B_{2m+4}(2-2^{-2m-1})\, f^{(2m+4)}(\xi)}{(m+1)(m+2)(2m+3)[(2m+4)!][(m+1)!]^2} \right| \quad 0 \le \xi \le 1 \tag{2.34}$$

with B_{2m+4} being the $(2m+4)^{th}$ Bernoulli number satisfying (2.31).

The only other quadrature rules which are of the Newton-Cotes type are those which have been developed for specific applications. In particular consider the problem of approximating integrals of the form

$$I(f) = \int_a^b w(x)f(x)dx = \int_a^b \cos mx\ f(x)dx. \qquad (2.35)$$

The integrand $f(x)\cos(mx)$ in general oscillates with the weight function $w(x) = \cos(mx)$ changing sign over the interval $[a,b]$. Common quadrature rules (i.e. Newton-Cotes or Gauss rules) require the weight function to be of constant sign over the interval of integration and consequently application of such rules to approximate (2.35) necessitates the partitioning of $[a,b]$ into many subintervals. The resulting quadrature formula will then involve an excessive number of function evaluations. This problem can be overcome by using Filon's formula.

If the interval $[a,b]$ is partitioned into $2n$ equal subintervals then Filon's formula is

$$\int_a^b f(x)\cos mx\ dx = h[\alpha(f(a)\cos am - f(b)\cos bm) + \beta S + \gamma T] + T_{2n}(f) \qquad (2.36)$$

where $S = -f(a)\sin am - f(b)\sin bm + 2\sum_{k=0}^{n} f(a+2kh)\sin(am+2kmh)$, $h = \dfrac{b-a}{2n}$

$$T = \sum_{k=1}^{n} f(a+(2k-1)h)\sin(am+(2k-1)mh)$$

$$\alpha = \frac{1}{\theta} + \frac{\sin(2\theta)}{2\theta^2} - \frac{2\sin^2\theta}{\theta^3}, \qquad \theta = mh$$

$$\beta = \frac{1+\cos^2\theta}{\theta^2} - \frac{\sin 2\theta}{\theta^3}$$

$$= 4\left[\frac{\sin\theta}{\theta^3} - \frac{\cos\theta}{\theta^2}\right]$$

and $$T_{2n}(f) = \frac{(b-a)}{12}\ h^3\left[1 - \frac{1}{16\cos\frac{\theta}{4}}\right]\sin\frac{\theta}{2}\ f^{(4)}(\xi), a < \xi < b\ .$$

When θ is small $\alpha, \beta,$ and γ have expansions of the form

$$\alpha = \frac{2}{45}\theta^3 - \frac{2}{215}\theta^5 + \frac{2}{4525}\theta^7 - \dots$$

$$\beta = 1/3 + \frac{\theta^2}{15} - \frac{2\theta^4}{105} + \frac{\theta^6}{567} - \dots$$

$$\gamma = 4/3 - \frac{2\theta^2}{15} + \frac{2\theta^4}{270} - \frac{\theta^6}{11340} + \dots$$

These power series expansions are required in order to avoid loss of significant figures due to cancellation when using the trigonometric representations for $\alpha, \beta,$ and γ. For small θ a connection between Filon's formula and a compound Simpson's rule is obtained by ignoring powers of θ greater than the first in the power series expansions for $\alpha, \beta,$ and γ. When θ is large, a modification to Simpson's rule is obtained if the Simpson coefficients 1/3 and 4/3 are replaced by β and γ respectively and an appropriate multiple of $f(a)\cos(am) - f(b)\cos(bm)$ is added.

2.6 Integration of arbitrarily spaced data

All of the above quadrature rules require that the nodes be uniformly distributed within the interval of integration. If the nodes are not equally spaced then it is possible to apply interpolation techniques using all the given nodes. In addition to the difficulty of generating quadrature formulae based on direct Lagrangian interpolation, high order formulae are discouraged because they magnify any errors which may be present in the function values. The method of interpolating to find the values of the function at the nodes of known quadrature rules and the subsequent use of such formulae is also not encouraged because of the compounding of errors. A possible recourse is to use low order interpolation based on the given abscissae.

A method for integrating functions specified at arbitrarily spaced nodes has been suggested by Hennion [23]. This method is based on di-parabolic interpolation. If $a = x_0 < x_1 < x_2 < \dots < x_n = b$ are the given nodes, quadratic interpolating polynomials $p_i(x) = a_i x^2 + b_i x + c_i$ $(i = 1, 2, \dots, n-1)$ are constructed where a_i, b_i and c_i are suitable scalars such that $p_i(x_j) = f(x_j)$ for $j = i-1, i, i+1$. Then for $i = 1, 2, \dots, n-2$ the quadrature approximations are of the form

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx 1/2 \int_{x_i}^{x_{i+1}} [p_i(x) + p_{i+1}(x)]dx$$
$$= \left(\frac{a_i+a_{i+1}}{2}\right)\left(\frac{x_{i+1}^3-x_i^3}{3}\right) + \left(\frac{b_i+b_{i+1}}{2}\right)\left(\frac{x_{i+1}^2-x_i^2}{2}\right) + \left(\frac{c_i+c_{i+1}}{2}\right)(x_{i+1}-x_i). \quad (2.37)$$

For the first and last intervals the approximations are

$$\int_{x_0}^{x_1} f(x)dx \approx \int_{x_0}^{x_1} p_1(x)dx = \frac{a_1^3}{3}(x_1-x_0)^3 + \frac{b_1}{2}(x_1-x_0)^2 + c_1(x_1-x_0) \quad (2.38)$$

and

$$\int_{x_{n-1}}^{x_n} f(x)dx \approx \int_{x_{n-1}}^{x_n} p_{n-1}(x)dx = \frac{a_{n-1}^3}{3}(x_n-x_{n-1})^3 + \frac{b_{n-1}}{2}(x_n-x_{n-1})^2 + c_{n-1}(x_n-x_{n-1}) \quad (2.39)$$

respectively. The approximating value for the integral $\int_a^b f(x)dx$

is obtained by summing the values computed by equations (2.37) to (2.39). When a does not correspond to any of the abscissae (say $x_i < a < x_{i+1}$ $(1 \le i \le n-2)$ then the approximation

$$\int_a^{x_{i+1}} f(x)dx \approx 1/2 \int_a^{x_{i+1}} [p_i(x)+p_{i+1}(x)]dx$$

is used. If $x_0 < a < x_1$ or if $a < x_0$ then the approximation

$$\int_a^{x_1} f(x)dx \approx \int_a^{x_1} p_1(x)dx$$

is made. Similar approximations are obtained for cases where $x_i < b < x_{i+1}$ $(1 \le i \le n-2)$, $x_{n-1} < b < x_n$, or $x_n < b$.

A defect of Hennion's algorithm is that it fails to provide any indication of the quadrature error, which, in the absence of any knowledge of the behaviour of the integrand, results in the quadrature approximation being of limited use. An algorithm by Gill and Miller [21] attempts to measure the reliability of a quadrature approximation obtained from a piecewise interpolatory scheme by approximating the theoretical derivative term in the quadrature error by a divided difference based on the given abscissae. For the case of quartic interpolation, their numerical experiments show that such an error estimate gives a favourable indication of the true quadrature error.

Another approach to the problem of integration of arbitrarily spaced data has been suggested by Secrest [47]. If one considers the set of evaluation points (x_i, f_i) $(i = 1, 2, \ldots, n)$ and defines

I_{min} and I_{max} to be the minimum and maximum values of the integral

$$I = \int_{x_1}^{x_n} f(x)dx \tag{2.40}$$

for a restricted class of functions passing through the evaluation points,then a reasonablequadrature approximation is

$$I_{best} = 1/2\ (I_{max}+I_{min}). \tag{2.41}$$

This is precisely the most desirable approximation proposed by Secrest. Associated with the quadrature approximation (2.41) is the "best error bound"

$$E_{best} = 1/2\ (I_{max}-I_{min})\ . \tag{2.42}$$

For the case of functions in the space $L_\infty^{(r)}$, the determination of the quadrature approximation I_{best} and its associated error estimate are dependent on the knowledge of bounds for the r^{th} derivative of the integrand. When the space $L_2^{(r)}$ is considered, a matrix-orientated algorithm is used to generate I_{best} and E_{best}. By this latter method, bounds on the derivatives of the integrand are not required although a knowledge of such bounds could give better estimates of the quadrature error.

2.7 Quadrature formulae with multiple nodes

Up to this point, the quadrature formulae have not included linear combinations of derivatives of the integrand. It is possible, however, to have quadrature formulae using derivatives by considering osculating polynomials. The most common interpolation of this type is Hermite interpolation. Suppose that the first derivatives of $f(x)$ are given at $r(0 \le r \le n+1)$ of the nodes $a \le x_0 < x_1 < \dots < x_{n-1} < x_n \le b$ and define

$$p_n(x) = \prod_{i=0}^{n} (x-x_i) \quad , \quad \ell_{jn}(x) = \frac{p_n(x)}{(x-x_j)p_n'(x_j)}$$

$$j = 0, 1, \dots, n$$

$$p_r(x) = \prod_{j=0}^{r} (x-x_j) \quad , \quad \ell_{jr}(x) = \frac{p_r(x)}{(x-x_j)p_r'(x_j)}$$

$$j = 0, 1, \dots, r .$$

Then the Hermite interpolation formula for $f(x)$ is

$$f(x) = \sum_{j=0}^{n} u_j(x)f(x_j) + \sum_{j=0}^{r} v_j(x)f'(x_j) + e_{nr}(x) \tag{2.43}$$

where $\quad v_j(x) = (x-x_j)\ell_{jn}(x)\ell_{jr}(x)$

$$u_j(x) = \begin{cases} \left\{1-(x-x_j)[\ell_{jn}'(x_j)+\ell_{jr}'(x_j)]\right\}\ell_{jn}(x)\ell_{jr}(x) & j = 0, 1, \dots, r \\ \ell_{jn}(x)\,\dfrac{p_r(x)}{p_r(x_j)} & j = r+1, \dots, n \end{cases}$$

and $\quad u_j(x_k) = \delta_{jk} \quad j,k=0,1,\dots,n \quad u_j'(x_k) = 0 \quad j=0,1,\dots,n;\ k=0,1,\dots,r$

$\quad v_j(x_k) = 0 \quad j=0,1,\dots,r;\ k=0,1,\dots,n \quad v_j'(x_k)=\delta_{jk} \quad j,k=0,1,\dots,r$

as required. The error in the interpolation is

$$e_{nr}(x) = \frac{p_n(x)p_r(x)}{(n+r+2)!} f^{(n+r+2)}(\xi(x)) ,$$

$$a < \xi(x) < b \qquad (2.44)$$

Now multiplying each side of (2.43) by $w(x)$ and integrating over [a,b] we get

$$\int_a^b w(x)f(x)dx = \sum_{j=0}^{n} a_{nj}f(x_j) + \sum_{j=0}^{r} a_{rj}f'(x_j) + T_{nr}(f) \qquad (2.45)$$

where $$a_{nj} = \int_a^b w(x)u_j(x)dx \qquad j = 0, 1, \ldots, n$$

$$a_{rj} = \int_a^b w(x)v_j(x)dx \qquad j = 0, 1, \ldots, r$$

and $$T_{nr}(f) = \int_a^b \frac{w(x)p_n(x)p_r(x)f^{(n+r+2)}(\xi(x))}{(n + r + 2)!} dx.$$

Since the interpolation formula (2.43) is exact for all polynomials of degree $(n+r+1)$ or less we have that the quadrature formula (2.45) is also exact for all polynomials of degree $(n+r+1)$ or less. For the case of formulae with derivatives of arbitrary order, the precision of the quadrature rule is directly proportional to the number and order of the multiple nodes. As in the case of the standard quadrature rules of sections 2.2 and 2.3, the quadrature rule (2.45) is also of maximum precision.

If derivatives of the integrand are either given at the endpoints of the interval of integration or else are convenient to calculate, then it is possible to generate families of simple high-order quadrature rules without using the technique of interpolation. The development of such quadrature rules is based on the fact that an arbitrary function having continuous derivatives of a specified order on the finite interval [a,b] can be represented there in terms of the Bernoulli polynomials. The Bernoulli polynomials $B_n(x)$ $(n \geq 0)$ are defined by the generating function $\frac{te^{xt}}{e^t-1} = \sum_{n=0}^{\infty} B_n(x) \frac{t^n}{n!}$.

Given the function $f(x)$ with continuous derivatives of order p, its representation over $[a,b]$ in terms of the Bernoulli polynomials is [29, ch.11]

$$f(x) = \frac{1}{b-a} \int_a^b f(t)dt + \sum_{k=1}^{p-1} \frac{(b-a)^{k-1}}{k!} B_k \left(\frac{x-a}{b-a}\right) [f^{(k-1)}(b)-f^{(k-1)}(a)]-$$

$$\frac{(b-a)^{p-1}}{p!} \int_a^b f^{(p)}(t) [B_p^* \left(\frac{x-t}{b-a}\right) - B_p^* \left(\frac{x-a}{b-a}\right)]dt \tag{2.46}$$

where $B_n^*(x) = B_n(x)$ for $0 \leq x < 1$ and $B_n^*(x+1) = B_n^*(x)$. By considering a quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=1}^{n} a_j f(x_j) + T_n(f) = S_n + T_n(f) \tag{2.47}$$

having a precision of degree m, it is possible to apply (2.46) and obtain the quadrature rule

$$\int_a^b w(x)f(x)dx = S_n + \sum_{k=0}^{p} C_k[f^{(m+k)}(b)-f^{(m+k)}(a)] + R_{m,p}(f) \qquad (2.48)$$

where

$$C_i = \frac{(b-a)^{m+i}}{(m+i+1)!} \quad \int_a^b w(t)B_{m+i+1}\left(\frac{t-a}{b-a}\right)dt - \sum_{k=1}^{n} a_k B_{m+i+1}\left(\frac{x_k-a}{b-a}\right)$$

and

$$R_{m,p}(f) = -\frac{(b-a)^p}{(p+1)!}\int_a^b f^{(m+p+2)}(x)\int_a^b K_m(t)[B^*_{p-1}\left(\frac{t-x}{b-a}\right) - B^*_{p-1}\left(\frac{t-a}{b-a}\right)]dt\ dx$$

whenever $f(x)$ has a continuous derivative of order $m+p+2$ on $[a,b]$. The function $K_m(t)$ is the kernel function of the quadrature error associated with the quadrature rule given in (2.47) and as such is dependent on the specified quadrature weights and abscissae.

Those quadrature rules which are of the form (2.48) are called Euler-Maclaurin formulae. The characteristic advantage, and in some sense optimal quality of these rules over other interpolatory quadrature rules, is that a sequence of increasingly high-order formulae are obtained without increasing the number of quadrature nodes. In addition, if the integration limits and quadrature weights and nodes in (2.47) are rational numbers then the coefficients C_i will also be rational numbers. This fact combined with the restricted selection of multiple nodes results in the preservation of the simplicity of the quadrature rules. A third advantage sometimes enjoyed by the Euler-Maclaurin formulae is the ease with which an accurate error estimate can be calculated. For a wide variety of the quadrature rules given by (2.47) the corresponding coefficients C_i in (2.48) alternate in sign. Hence for those integrands having successive derivatives of similar sign over the interval [a,b], the Euler-Maclaurin formulae are partial sums of an alternating series. In such instances the quadrature error will be less than the first neglected term in the series. Such an error estimate will often be easier to obtain than that of calculating sharp bounds for $R_{m,p}(f)$ in (2.48).

The most commonly used Euler-Maclaurin formula is the modified trapezoidal rule

$$\int_a^b f(x)dx = S_n - \sum_{k=1}^{p-1} \frac{h^{2k}}{(2k)!} B_{2k}[f^{(2k-1)}(b)-f^{(2k-1)}(a)]+R_{m,2p}(f) \tag{2.49}$$

where $S_n = h[f(a)/2 + \sum_{i=1}^{n-1} f(a+ih)+f(b)/2]$, $h = \frac{b-a}{n}$, and B_{2k} is the $2k^{th}$ Bernoulli number satisfying (2.31). For $f^{(2p)}(x)$ continuous on [a,b] the truncation error $R_{m,2p}(f)$ can be written as $R_{m,2p}(f) = -\frac{nh^{2p+1}}{(2p)!} B_{2p}f^{(2p)}(\xi)$ for some ξ , $a < \xi < b$. If S_{2n} represents the compound Simpson's rule (2.26) then the corresponding Euler-Maclaurin formula is

$$\int_a^b f(x)dx = S_{2n} + \sum_{k=2}^{p} \frac{(2h)^{2k}}{3(2k)!} (1-2^{-2k+2})B_{2k}[f^{(2k-1)}(b)-f^{(2k-1)}(a)] + R_{2m,2p+2}(f) \tag{2.50}$$

for all functions $f(x)$ having continuous derivatives of order 2P+2 on $[a,b]$. In such instances the quadrature error has a definite form

$$R_{2m,p+2}(f) = \frac{nh(2h)^{2p+2}}{2(2p+2)!}(1-2^{-2p})B_{2p+2}f^{(2p+2)}(\xi) \quad \text{where} \quad h = \frac{b-a}{2n} .$$

Other Euler-Maclaurin expansions can be determined for both higher-order Newton-Cotes rules and Gaussian quadrature rules (cf. [29,ch. 11]).

When the derivatives in (2.48) are difficult to determine, it is still possible to obtain quadrature rules with improved accuracy by approximating the derivatives with finite differences. The practicality of such a method, however, is limited to those cases where the underlying quadrature rule (2.47) is based on uniformly distributed abscissae. If such a procedure is used, the derivative approximations are usually based on the Newton-Gregory interpolating polynomials, that is,

$$f(x) = f(a+th) = f_0 + t\Delta f_0 + \ldots + \frac{t(t-1)\ldots(t-k)}{k!}\Delta^k f_0 + \ldots \tag{2.51}$$

for values of x close to a and

$$f(x) = f(a+th) = f_n + t\Delta f_{n-1} + \ldots + \frac{t(t+1)\ldots(t+k)}{k!}\Delta^k f_{n-k} + \ldots \tag{2.52}$$

for values of x close to b. The notation used in (2.51) and (2.52) is $f_i = f(a+ih)$, $\Delta f_i = f_{i+1} - f_i$, $f_n = f(b)$, and $\Delta f_{n-i} = f_{n-i+1} - f_{n-1}$ where $h = \frac{b-a}{n}$.

By differentiating both (2.51) and (2.52) with respect to

t and setting $t = 0$ and $t = n$ for the respective derivatives, accurate approximations to the derivatives of $f(x)$ at $x = a$ and $x = b$ can be obtained. If all the differences beyond the first term are neglected, then the approximations $h^k f^{(k)}(a) \approx \Delta^k f_0$ and $h^k f^{(k)}(b) \approx \Delta^k f_{n-k}$ are obtained. When the approximations are used in the modified trapezoidal rule (2.49) the resulting quadrature rule is

$$\int_a^b f(x)dx = S_n - h \sum_{k=1}^{p} C_k[\Delta^k f_{n-k}+(-1)^k \Delta^k f_0]+R_{n,p}(f) \qquad (2.53)$$

where $C_k = \frac{(-1)^k}{(k+1)} \int_0^1 x(x-1)\ldots(x-k)dx$. This quadrature approximation is called Gregory's formula. A convenient representation for $R_{n,p}(f)$ in (2.53) is difficult to determine and as such presents a handicap for the use of Gregory's formula. However, when the successive terms in the expansion of (2.53) alternate in sign, a suitable bound for the error can be computed.

CHAPTER III

QUADRATURE RULES WITH THE HIGHEST DEGREE OF PRECISION

3.1 Gaussian quadrature rules

In the previous chapter it was shown that for a given set of nodes $\{x_0, x_1, \ldots, x_n\}$ in $[a,b]$, a quadrature formula

$$\int_a^b w(x)f(x)dx = \sum_{j=0}^{n} a_j f(x_j) + R_n(f) \tag{3.1}$$

could be constructed such that (3.1) has a degree of precision of at least n. The resulting quadrature formulae are based on Lagrangian interpolation and are optimal in the sense of having the greatest precision. If the abscissae are not fixed then it is to be expected that a judicious choice of nodes in the interval $[a,b]$ could increase the precision of the quadrature formula. This is indeed possible and quadrature formulae of the form (3.1) can be derived so that they have a degree of precision at most $2n+1$. These formulae are also of the interpolatory type with the weights a_j $(j = 0, 1, \ldots, n)$ being uniquely defined by the integration limits, weight function, and chosen abscissae.

Quadrature formulae derived by choosing the abscissae so that a maximum degree of precision is obtained are called Gaussian (or Gauss-Jacobi) quadrature formulae. These quadrature formulae are related to the theory of orthogonal polynomials and as such have different names attached to families of quadrature rules with each name depending on the weight function and the corresponding set of orthogonal polynomials.

If $w(x)$ is an arbitrary integrable weight function of constant sign on $[a,b]$ then a set of polynomials $\{p_m(x)\}$ orthogonal with respect to $w(x)$ on the interval $[a,b]$ can be constructed by using the Gram-Schmidt method [3]. These orthogonal polynomials satisfy the recurrence relation [51, p. 42]

$$p_{n+1}(x) = (A_{n+1}x + B_{n+1})p_n(x) - C_{n+1}p_{n-1}(x) \qquad (3.2)$$

$$n = 1, 2, 3, \ldots$$

where the quantities A_{n+1}, B_{n+1}, C_{n+1} are constants and A_{n+1}, $C_{n+1} > 0$. If the coefficient of the highest power of x in $p_{n+1}(x)$ is denoted by k_{n+1} we have

$$A_{n+1} = \frac{k_{n+1}}{k_n} . \qquad (3.3)$$

Since $p_{n+1}(x)$ is an orthogonal polynomial of degree $n+1$ it has $n+1$ real roots $x_0 < x_1 < \ldots < x_n$. Each root is located in the interior of the interval $[a,b]$ and if $x_{-1} = a$ and $x_{n+1} = b$ then each interval $[x_v, x_{v+1}]$ $v = -1, 0, \ldots, n$ contains exactly one zero of $p_{n+2}(x)$.

By considering the orthogonal polynomial $p_{n+1}(x)$ of degree $n+1$, a Gaussian quadrature formula having precision of degree $2n+1$ can be constructed. The existence and actual construction of the Gaussian quadrature rules are established on the basis of the following theorem

Theorem 3.1:

The quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=0}^{n} a_j f(x_j) \tag{3.4}$$

is exact for all polynomials of degree $\le 2n+1$ iff (3.4) is interpolatory and $x_0 < x_1 < \dots < x_n$ are the roots of the orthogonal polynomial $p_{n+1}(x)$ of degree $n+1$. The weights a_j ($j = 0, 1, \dots, n$) are given by

$$a_j = \int_a^b \frac{w(x)p_{n+1}(x)}{(x-x_j)p'_{n+1}(x_j)} dx \;. \tag{3.5}$$

Proof

Let $f(x)$ be an arbitrary polynomial of degree $\le 2n+1$ and let $x_0 < x_1 < \dots < x_n$ be the roots of the orthogonal polynomial $p_{n+1}(x)$. As previously mentioned, the roots are real and are located in the interior of $[a,b]$. Constructing the Lagrangian interpolation polynomial $L(x)$ of degree n for $f(x)$ and the points $x_0, x_1, \dots, x_n$ we have

$$L(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x) = \sum_{j=0}^{n} f(x_j) \frac{p_{n+1}(x)}{(x-x_j)p'_{n+1}(x_j)} \tag{3.6}$$

Since $L(x_j) = f(x_j)$ ($j = 0, 1, \dots, n$) then $f(x)-L(x) = p_{n+1}(x)r(x)$ with $r(x)$ being a polynomial of degree $\le n$. Hence multiplying both sides by $w(x)$ and integrating we have

$$\int_a^b w(x)f(x)dx = \int_a^b w(x)L(x)dx + \int_a^b w(x)p_{n+1}(x)r(x)dx \qquad (3.7)$$

$$= \int_a^b w(x)L(x)dx = \sum_{j=0}^{n} a_j f(x_j) \ ,$$

$$a_j = \int_a^b \frac{w(x)p_{n+1}(x)}{(x-x_j)p'_{n+1}(x_j)} \quad j=0(1)n \ . \qquad (3.8)$$

The second integral on the right side of (3.7) vanishes because of the assumed orthogonality of $p_{n+1}(x)$. Therefore (3.4) is exact for all $f(x)$ of degree $\le 2n+1$. Then (3.4) is exact for all polynomials of degree $\le n$ and by theorem (2.1) it must be interpolatory. Conversely, assume that (3.4) is exact for all polynomials of degree $\le 2n+1$ and let $f(x)=(x-x_0)(x-x_1)\cdots(x-x_n)=\ell(x)r(x)$ where $r(x)$ is of degree $\le n$. Then $f(x_k)=0$ $(k=0(1)n)$ and $\int_a^b w(x)\ell(x)r(x)dx=0$ so that $\ell(x) = K\cdot p_{n+1}(x)$ for some constant K. Therefore $x_0 < x_1 < \ldots < x_n$ are roots of the orthogonal polynomial $p_{n+1}(x)$.

So far the only requirements for the weight function $w(x)$ are that it be integrable and be of constant sign over the interval $[a,b]$. If $w(x)$ does change sign in $[a,b]$ then orthogonal polynomials for $w(x)$ may not exist and even if they do, the roots may be complex or lie outside the interval of integration. Therefore Gaussian quadrature rules are derived only for weight functions of constant sign in the interval of integration. For such weight functions, we obtain quadrature rules of maximum precision as indicated in the following Theorem [29, p.102]

Theorem (3.2):

If $w(x)$ does not change sign on $[a,b]$ then no matter how we choose x_j and a_j, equation (3.4) cannot be exact for all polynomials of degree $2n+2$. Therefore Gaussian quadrature rules are optimal in the sense of having the maximum possible degree of precision when the choice of quadrature abscissae is unrestricted.

The weights a_j $(j = 0, 1, \ldots, n)$ given in (3.4) are often called Christoffel numbers and are characterized by the following theorem

Theorem (3.3):

The Christoffel numbers a_j are positive, and

(i) $$\sum_{j=0}^{n} a_j = \int_a^b w(x)dx \qquad (3.9)$$

(ii) The following representations hold;

$$a_j = \int_a^b \left[\frac{p_{n+1}(x)}{(x-x_j)\,p'_{n+1}(x_j)}\right]^2 w(x)dx \qquad (3.10a)$$

$$a_j = \left(\frac{k_{n+1}}{k_n}\right) \frac{1}{p_n(x_j)\,p'_{n+1}(x_j)} \qquad j = 0, 1, \ldots, n\,;$$

k_{n+1}, k_n as given in equation (3.3) (3.10b)

$$a_j = \frac{1}{\sum_{i=0}^{n+1} \left(p_i(x_j)\right)^2} \qquad j = 0, 1, \ldots, n \qquad (3.10c)$$

(iii) $$a_{0,n+1}+\ldots+a_{k-1,n+1} < a_{0,n+2}+\ldots+a_{k,n+2} < a_{0,n+1}+\ldots+a_{k,n+1}$$

$k=0,1,\ldots,n$ (3.11)

where $\{a_{j,n+1}\}_{j=0}^{n}$ are the Christoffel numbers for (3.4) having n+1 nodes and $\{a_{j,n+2}\}_{j=0}^{n+1}$ are the Christoffel numbers for (3.4) having n+2 nodes.

On the basis of theorems (3.1) and (3.2) and equation (3.2), Gaussian quadrature rules can be constructed for any arbitrary weight function $w(x)$ having the previously mentioned restrictions and for any degree of precision. In practice, however, only certain weight functions have been chosen for developing Gaussian quadrature rules. These weight functions generate sets of classical orthogonal polynomials whose behaviour and characteristics are well known ([1],[51]). Also the known quadrature rules seldom have large number of abscissae because much computational effort is required to calculate the quadrature weights and nodes.

Tables 3.1 and 3.2 give the names and properties of families of quadrature rules for the more commonly used weight functions. Gaussian type quadrature rules for the weight functions $w(x) = \sqrt{x}$ and $w(x) = \frac{1}{\sqrt{x}}$ have been discussed by Krylov [51] and properties for quadrature formulae with weight functions $w(x) = -\log x$ $(0 \leq x \leq 1)$ and $w(x) = x^n$ $n \geq 0,\ -1 \geq x \geq 1$ have been investigated by Anderson [3] and Rothman [42] respectively.

Name of Quadrature Rules	Weight Function $w(x)$	Associated Orthogonal Polynomials of Degree n	Recurrence Relation for Orthogonal Polynomials
Gauss-Hermite	e^{-x^2}	Hermite Polynomials $H_n(x) = n! \sum_{k=0}^{[\frac{n}{2}]} \frac{(-1)^k (2x)^{n-2k}}{k!(n-2k)!}$	$H_0(x) = 1$, $H_1(x) = x$ $H_n(x) = 2x\, H_{n-1}(x) - 2(n-1)H_{n-2}(x)$, $n > 1$
Gauss-Laguerre	$x^{\alpha} e^{-x}$, $\alpha > -1$	Laguerre Polynomials $L_{\alpha,n}(x) = \sum_{k=0}^{n} \binom{n+\alpha}{n-k} \frac{(-x)^k}{k!}$	$L_{\alpha,0}(x) = 1$, $L_{\alpha,1}(x) = -x + \alpha + 1$ $nL_{\alpha,n}(x) = (-x+2n+\alpha-1)L_{\alpha,n-1}(x)-(n+\alpha-1)L_{\alpha,n-2}(x)$, $n > 1$
Gauss-Jacobi	$(1-x)^{\alpha}(1+x)^{\beta}$ $\alpha, \beta > -1$	Jacobi Polynomials $P_n^{(\alpha,\beta)}(x) = \sum_{k=0}^{n} \left[\binom{n+\alpha}{n-k}\binom{n+\beta}{k} \left(\frac{x-1}{2}\right)^k \left(\frac{x+1}{2}\right)^{n-k}\right]$	$P_0^{(\alpha,\beta)}(x) = 1$, $P_1^{(\alpha,\beta)}(x) = \tfrac{1}{2}(\alpha+\beta+2)x + \tfrac{1}{2}(\alpha-\beta)$ $2n(n+\alpha+\beta)\ (2n+\alpha+\beta-2)P_n^{(\alpha,\beta)}(x) = (2n+\alpha+\beta-1)\Big\{(2n+\alpha+\beta)(2n+\alpha+\beta-2)x + (\alpha^2-\beta^2)\Big\} P_{n-1}^{(\alpha,\beta)}(x)$ $-2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)P_{n-2}^{(\alpha,\beta)}(x)$, $n > 1$

Table 3.1 Families of Gaussian quadrature rules $\int_a^b w(x)f(x) = \sum_{k=1}^{n} a_k\, f(x_k)$

Name of Quadrature Rules	Integration Limits	Quadrature Weights*	Quadrature Error $a < \eta < b$
Gauss-Hermite	$a = -\infty \quad , b = \infty$	$a_k = \dfrac{n!\sqrt{\pi}\ 2^{n+1}}{[H_n'(x_k)]^2}$	$\dfrac{n!\sqrt{\pi}}{2^n(2n)!}\ f^{(2n)}(\eta)$
Gauss-Laguerre	$a = 0 \quad , b = \infty$	$a_k = \dfrac{\Gamma(n+\alpha+1)}{n!\,x_{kn}[L_{\alpha,n}'(x_k)]^2}$	$\dfrac{(n!)^2}{(2n)!}\ f^{(2n)}(\eta)$
Gauss-Jacobi	$a = -1 \quad , b = 1$	$a_k = \dfrac{2^{\alpha+\beta+1}\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{n!\,\Gamma(n+\alpha+\beta+1)(1-x_k^2)[P_n^{(\alpha,\beta)'}(x_k)]^2}$	$\dfrac{2^{2n+\alpha+\beta+1}\Gamma(n+\alpha+1)\Gamma(n+\beta+1)\Gamma(n+\alpha+\beta+1)n!}{\Gamma(2n+\alpha+\beta+1)\Gamma(2n+\alpha+\beta+2)(2n)!}\ f^{(2n)}(\eta)$

* x_{kn} denotes the k^{th} largest root of the associated orthogonal polynomial of degree n

Table 3.1 (continued)

Name of Quadrature Rules	Weight Function $w(x)$	Associated Orthonormal Polynomials of Degree n	Recurrence Relation for Orthogonal Polynomials
Gauss-Legendre ($\alpha = \beta = 0$)	1	Legendre polynomials $P_n^{(\alpha,\beta)}(x)=P_n(x)= \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2-1)^n$	$P_0(x) = 1$, $P(x) = x$ $(n+1)P_{n+1}(x)=(2n+1)P_n(x)-nP_{n-1}(x)$, $n>1$
Gauss-Chebyshev ($\alpha = \beta = 1/2$)	$(1-x^2)^{-1/2}$	Chebyshev polynomials of 1^{st} kind $P_n^{(\alpha,\beta)}(x) = \frac{(2n-1)!}{2^{2n-1} n!(n-1)!} T_n(x)$ $T_n(x) = \cos n\theta$, $x = \cos\theta$	$T_0(x) = 1$, $T_1(x) = x$ $T_{n+1}(x)=2xT_n(x)-T_{n-1}(x)$, $n>1$
Gauss-Chebyshev ($\alpha = \beta = 1/2$)	$(1+x^2)^{1/2}$	Chebyshev polynomials of 2^{nd} kind $P_n^{(\alpha,\beta)}(x) = \frac{(2n+1)!}{2^{2n} n!(n+1)!} U_n(x)$ $U_n(x) = \frac{\sin((n+1)\theta)}{\sin\theta}$, $x = \cos\theta$	$U_0(x) = 1$, $U_1(x) = x$ $U_{n+1}(x)=2xU_n(x)-U_{n-1}(x)$, $n>1$
Gauss-Jacobi $\alpha = \beta = 1/2$	$\left(\frac{1-x}{1+x}\right)^{1/2}$	Jacobi polynomials $P_n^{(\alpha,\beta)}(x) = \frac{(2n)!}{2^{2n}(n!)^2} \frac{\sin((n+1)\theta/2)}{\sin(\theta/2)}$, $x = \cos\theta$	$P_0^{(1/2,-1/2)}(x)=1$, $P_0^{(1/2,-1/2)}(x)=x$ $2n(n+1)P_{n+1}^{(1/2,-1/2)}(x)=2n(n+1)\times$ $xP_n^{(1/2,-1/2)}(x)-(4n^2-1P_{n-1}^{(1/2,-1/2)}(x)$, $n>1$

Table 3.2 Families of Gauss-Jacobi quadrature rules $\int_{1}^{1} w(x)f(x) = \sum_{k=1}^{n} a_k f(x_k)$

Name of Quadrature Rules	Quadrature Abscissae*	Quadrature Weights	Quadrature Error $-1 < \eta < 1$
Gauss-Legendre ($\alpha = \beta = 0$)	$x_k = 1 - \frac{4\xi_k}{2n+1+\xi_k}$; $\xi_k = \frac{j_{0,k}^2}{4n+2}\left(1+\frac{j_{0,k-2}^2}{12(n+1)^2}\right) + O\left(\frac{1}{n^5}\right)$	$a_k = \frac{2}{(1-x_k^2)[P_n'(x_k)]^2}$	$\frac{2^{2n+1}}{(2n+1)}\left(\frac{n!}{(2n)!}\right)^2 f^{(2n)}(\eta)$
Gauss-Chebyshev ($\alpha = \beta = 1/2$)	$x_k = \cos\left(\left(\frac{2k-1}{2n}\right)\pi\right)$	$a_k = \frac{\pi}{n}$	$-\frac{\pi}{2^{2n-1}(2n)!} f^{(2n)}(\eta)$
Gauss-Chebyshev ($\alpha = \beta = 1/2$)	$x_k = \cos\left(\frac{k\pi}{n+1}\right)$	$a_k = \frac{\pi}{n+1}\sin^2\left(\frac{k\pi}{n+1}\right)$	$\frac{\pi}{2^{2n}(2n)!} f^{(2n)}(\eta)$
Gauss-Jacobi ($\alpha = -\beta = 1/2$)	$x_k = \cos\left(\frac{2k\pi}{2n+1}\right)$	$a_k = \frac{4\pi}{2n+1}\sin^2\left(\frac{k\pi}{2n+1}\right)$	$\frac{\pi}{2^{2n}(2n)!} f^{(2n)}(\eta)$

*$j_{0,k}$ denotes the k^{th} largest root of Bessel's function of the 1^{st} kind of order 0.

Table 3.2 (continued)

A variation of the Gaussian quadrature rule (3.1) is the combination of both fixed and variable abscissae. For such rules a set of abscissae $\{x_{fj}\}_{j=1}^{m}$ in $[a,b]$ are preassigned and a set of parametric abscissae $\{x_{vk}\}_{k=1}^{n}$ in $[a,b]$ are to be determined so that the quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=1}^{n} a_j\ f(x_{vj}) + \sum_{j=1}^{m} b_j f(x_{fj}) + E_{m,n}(f) \tag{3.12}$$

has maximum precision. In the case of (3.12), there are $2n + m$ parameters x_{vj}, a_j $(j = 1, \ldots, n)$ and b_j $(j = 1, \ldots, m)$ so that the maximum possible degree of precision is $2n + m-1$. In deriving the quadrature rule (3.12) an interpolating polynomial $g(x)$ of degree $\leq 2n+m-1$ is constructed from the conditions

$$\begin{aligned} g(x_{fj}) &= f(x_{fj}) = j = 1, \ldots, m \\ g(x_{vk}) &= f(x_{vk}),\ g'(x_{vk}) = f'(x_{vk}) \\ &\qquad k = 1, \ldots, n \end{aligned} \tag{3.13}$$

By multiplying the interpolating polynomial by the weight function $w(x)$ and integrating over the interval $[a,b]$ the quadrature rule (3.4) is obtained with

$$a_j = \int_a^b \frac{w(x)\alpha(x)\beta(x)}{(x-x_{vj})\alpha'(x_{vj})\beta(x_{vj})} dx \qquad j = 1, \ldots, n$$

$$b_k = \int_a^b \frac{w(x)\alpha(x)\beta(x)}{(x-x_{fk})\alpha(x_{fk})\beta'(x_{fk})}\,dx \qquad k = 1, \ldots, m$$

$$E_{m,n}(f) = \frac{f^{(2n+m)}(\eta)}{(2n+m)!}\int_a^b w(x)\alpha^2(x)\beta(x)dx \qquad a < \eta < b$$

and where $\alpha(x) = \prod_{j=1}^{n} (x - x_{vj})$, $\beta(x) = \prod_{k=1}^{m} (x - x_{fk})$. The abscissae $x_{vk}(k = 1, \ldots, n)$ correspond to the roots of the n^{th} degree polynomial $p_n(x)$ which is orthogonal to $w(x)\beta(x)$. For the special cases of $m = 1$, $x_{v1} = a$ and $m = 2$, $x_{v1} = a$, $x_{v2} = b$ the resulting quadrature rules are called Radau and Lobatto quadratures respectively.

Calculation of the Gaussian quadrature weights and abscissae invariably depends on the existence of polynomials orthogonal with respect to the weight function $w(x)$ and any method for calculating the parameters of the quadrature formulae will explicitly or implictly use the orthogonal polynomials and their associated properties.

Direct calculation of the quadrature weights and nodes can be accomplished by using equation (3.2) for generation of the orthogonal polynomials, equation (3.10c) for calculation of the weights, and Newton's method or a modification thereof[14,p.45] for determination of the abscissae. From the nesting property of the zeros of successive orthogonal polynomials, accurate initial approximations for their roots can be obtained so that few iterations of Newton's method are required. This direct approach is useful only when the explicit form of the recurrence relation (3.2) is known and generation of the orthogonal polynomials is easily accomplished.

For non-standard weight functions giving non-classical orthogonal polynomials it is often advisable to use indirect methods for accurate determination of the quadrature nodes and weights. Several such methods have been proposed by various authors [43,], [18], [19], [46], [22]. The method suggested by Rutishauser [43] uses a continued fraction expansion.

$$\int \frac{w(x)}{z - x} \, dx = \frac{1|}{|z} - \frac{q_1|}{|1} - \frac{e_1|}{|z} - \frac{q_2|}{|1} - \frac{e_2|}{|z} - \cdots \qquad (3.14)$$

For the successive convergents $\frac{R_n(z)}{S_n(z)}$ of (3.12), $S_n(z)$ corresponds to the orthogonal polynomial of degree n on $[a,b]$. Hence the poles of the convergents give the abscissae of the quadrature formulae and from the identity

$$\frac{R_n(z)}{S_n(z)} = \text{const.} \sum_{k=1}^{n} \frac{w_{kn}}{z-x_{kn}} \qquad (3.15)$$

the weights and abscissae can be uniquely determined when a modified quotient-difference algorithm [44] is applied to (3.15). The drawback of this method is that numerical stability is not always guaranteed [14, p. 48].

If neither the moments

$$\mu_k = \int_a^b x^k w(x) dx \qquad k = 1, 2, \ldots \qquad (3.16)$$

nor the recurrence relationships (3.2) for generation of the orthogonal polynomials to $w(x)$ are known, then an indirect method due to Gautschi can be used for calculating the quadrature weights and nodes. Gautschi's algorithm [18] and ALGOL 60 procedure [34] generate the abscissae and weights by making the inner product which defines the orthogonal polynomials a discrete inner product. The desired abscissae and weights are then approximated by the zeros and weight factors of the orthogonal polynomials of a discrete variable. This discretization is made so as to enable the calculation of the coefficients A_n, B_n, C_n in the recurrence relation (3.2) to be made. Having obtained these coefficients the quadrature abscissae are determined by applying Newton's method to the orthogonal polynomials. The quadrature weights are subsequently calculated using equation (3.10c). Provided that any singularities for the weight function are simple and located at the endpoints of the interval of integration, convergence of the method is obtained when the number of points used in the discretization becomes infinite.

If the weight function $w(x)$ in (3.1) is not known explicitly but a set of $2N+1$ moments are known then a three term recurrence relation for polynomials $\{p_k(x)\}_{k=0}^{N}$ orthogonal to $w(x)$ can be determined. Golub and Welsch [22] show that the Cholesky decomposition of a Grammian for the moments μ_k $(k = 0, 1, \ldots, 2N)$ uniquely determines the three term recurrence relations for a set of polynomials $\{p_k(x)\}_{k=0}^{N}$ orthogonal to $w(x)$. Using the generated recurrence relation, a matrix equation

$$x\vec{P}_N(x) = T_{N\times N}\,\vec{P}_N(x) + \frac{1}{A_0}\,\vec{e}_N \tag{3.17}$$

is obtained where $T_{N\times N}$ is a tridiagonal matrix, $\vec{P}_N(x) = (p_0(x), \ldots, p_{N-1}(x))^T$, and $\vec{e}_N = (0, 0, \ldots, 0, 1)^T$. The tridiagonal matrix $T_{N\times N}$ is either initially symmetric or else converted into a symmetric tridiagonal matrix by applying a diagonal similarity transformation. Since the eigenvalues of $T_{N\times N}$ correspond to the zeros of $p_N(x)$ and the weights of the quadrature rule are calculated from the first component of the orthonormal eigenvectors [52], an application of the QR algorithm of Sack [45] produces the weights and abscissae of the quadrature rule. The drawback of this method, as Golub and Welsch point out, is that the Gram matrix usually becomes ill-conditioned as N increases.

To overcome the problem of ill-conditioning either of the methods of Gautschi [34] or Sack and Donovan [46] can be used. Each is based on the method of modified moments but the algorithm of Sack and Donovan requires less storage and on the average requires less computation time. On that basis, only the latter algorithm will be considered here. Using the weight function $w(x)$, a set of normalized modified moments

$$v_k = \frac{\int_\alpha^\beta p_k(x)w(x)dx}{\int_a^b p_k^2(x)dx}, \qquad k = 0, 1, \ldots, 2n \tag{3.18}$$

can be constructed with $\{p_k(x)\}_{k=0}^{2n}$ being a set of polynomials not necessarily orthogonal with respect to $w(x)$ on the interval

$[\alpha,\beta]\subseteq[a,b]$. To ensure numerical stability for the method, it is suggested by Sack and Donovan that these polynomials be orthogonal to some "standard" weight function $w_0(x)$. These polynomials satisfy a known recurrence relation

$$xp_j(x) = a_j p_{j+1}(x) + b_j p_j(x) + c_j p_{j-1}(x) \tag{3.19}$$

If $\{T_k(x)\}_{k=0}^{\infty}$ denotes the set of polynomials orthogonal with respect to the weight function $w(x)$, we have a recurrence relationship

$$xT_i(x) = \alpha_i T_{i+1}(x) + \beta_i T_i(x) + \gamma_i T_{i-1}(x) \tag{3.20}$$

(with $\alpha_i = \gamma_{i+1}$) whenever the polynomials are orthonormal. For computational convenience assume that they are in fact orthonormal. As in the previous method it is desired to find the eigenvalues of the matrix

$$A_n = \begin{bmatrix} \beta_0 & \gamma_1 & & & & \\ \alpha_0 & \beta_1 & \gamma_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \\ & & & \alpha_{n-3} & \beta_{n-2} & \gamma_{n-1} \\ & & & & \alpha_{n-2} & \beta_{n-1} \end{bmatrix} \tag{3.21}$$

for they correspond to the abscissae of the n-point Gaussian quadrature rule for the weight function $w(x)$. Since each $T_i(x)$ can be expressed as a linear combination of polynomials $p_j(x)$, $j \leq i$ a matrix equation

$$\vec{T}(x) = Q\,\vec{P}(x) \tag{3.22}$$

is obtained where Q is an (in)finite lower triangular matrix with non-zero elements and $\vec{T}(x)$ and $\vec{P}(x)$ are (in)finite vectors. The characteristic equation for (3.21) is $|A_n - \lambda I_n| = 0$ and in view of (3.22) we obtain

$$|A_n - \lambda I_n| = |Q_n(M_n - \lambda I)Q_n^T| = 0 \tag{3.23}$$

where $$M_n = (\xi_{ij}) = (\int x\, p_i(x) p_j(x) dx) \tag{3.24}$$

and $$N_n = (\eta_{ij}) = (\int p_i(x) p_j(x) dx) \tag{3.25}$$

Since Q_n is nonsingular $|M_n - \lambda I_n| = 0 = |N_n^{-1} - \lambda I_n|$ so that the matrix A_n has the same eigenvalues as the asymmetric matrix $N_n^{-1}M_n = X_n + Y_n$ where Y_n has nonzero elements in the last column only and

$$X_n = \begin{bmatrix} b_0 & c_1 & & & & \\ a_0 & b_1 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \\ & & & a_{n-3} & b_{n-2} & c_{n-1} \\ & & & & a_{n-2} & b_{n-1} \end{bmatrix} \tag{3.26}$$

From the equality of the traces $T_r(A_n) = T_r(X_n + Y_n)$ and $T_r(A_n^2) = T_r((X_n + y_n)^2)$ we find that

$$\beta_i = T_r(A_{i+1}) - T_r(A_i) = b_i + y_i^{(i+1)} - y_{i-1}^{(i)} \tag{3.27}$$

and $$\alpha_i^2 = 1/2\ [T_r(A_{i+2}^2) - T_r(A_{i+1}^2) - \beta_{i+1}^2]$$

$$= a_i(c_{i+1} + y_i^{(i+2)}) + y_i^{(i+1)}(b_{i+1} - y_{i+1}^{(i+2)} - y_i^{(i+1)}) - a_{i-1}y_{i-1}^{(i+2)} \tag{3.28}$$

To find the corresponding components of $\vec{y}^{(j)}$ we note that construction of a matrix

$$S_n = (s_{ij}) = (\int q_i(x)p_j(x)w(x)dx) \tag{3.29}$$

where $$s_{ij} = 0 \quad j < i\ ; \qquad s_{ii} = 1 \tag{3.30}$$

gives $$\rho_i q_{i+1}(x) = (x-\tau_i)q_i(x) - \sigma_i q_{i-1}(x) \tag{3.31}$$

$$\text{and} \quad \rho_i s_{i+1,j} = (b_j - \sigma_i) s_{ij} + a_j s_{i,j+1} + a_j s_{i,j-1} - \tau_i s_{i-1,j} \tag{3.32}$$

for which
$$\rho_i = b_{i+1} - \sigma_i$$

$$\sigma_i = a_i s_{i,i+1} + b_i - a_{i-1} s_{i-1,i} \tag{3.33}$$

$$\tau_i = a_{i-1}$$

Then from the starting values $s_{-1,j}=0$, $s_{0,j} = \dfrac{v_i}{v_0}$, $j = 0, 1, \ldots, n-1$ the recurrence relation (3.32) yields s_{ij} for $i > j$ (j = 0, 1, ..., n-1). The components of the matrices Y_n are given in terms of S_n and

$$y_{n-1}^{(n)} = a_{n-1} s_{n-1,n} \quad , \tag{3.34}$$

and
$$y_{n-2}^{(n)} = a_{n-1} (s_{n-2,n} - s_{n-2,n-1} s_{n-1,n})$$

Hence using (3.33) and (3.34) we obtain $\beta_i = \sigma_i$ and $\alpha_i^2 = \rho_i \tau_{i+1}$. Having computed the matrix A_n, its eigenvalues (which correspond to the quadrature nodes) can be determined by using a rational variant of the QR tranformation with explicit shift. An Algol-60 procedure for this method has been published by Reinsch [40].

Extensive tables of Gaussian quadrature weights and abscissae for standard weight functions have been published. In Stroud and Secrest's book on Gaussian quadrature rules [50], tables of weights and abscissae for Gauss-Jacobi, Legendre, Laguerre, Hermite, Radau, and Lobatto quadrature formulae of various orders are given. Tables which augment those found in [50] and extend them to the cases of other weight functions have been published in journals such as Mathematics of Computation and U.S. National Bureau of Standards, Journal of Research.

3.2 Error-estimates for Gaussian quadrature formulae

From the results of theorem (3.1), every n-point Gaussian quadrature formula has an associated truncation error, $E_n(f)$, such that $E_n(f) = 0$ whenever $f(x)$ is a polynomial of degree $\leq 2n-1$. In the case of less regular integrands, exact representation of the truncation error can be derived by appealing to analytical methods of which the most common are (i) use of Peano's theorem, (ii) use of complex variable theory, and (iii) use of the theory of Hilbert spaces. The resulting representations for $E_n(f)$ are applicable to a large class of functions but unfortunately involve integrals and/or summations which do not easily allow reduction to a usable form. Therefore the best that one can hope for are estimates or bounds on the truncation error with sharp bounds being obtained only in special cases.

To obtain bounds for $E_n(f)$ on the basis of Peano's theorem the following restrictions for the integrand $f(x)$ are required;

(i) $f(x)$ has a continuous derivative of order $(2n-1)$ on $[a,b]$

(ii) $f^{(2n)}(x)$ is piecewise continuous on $[a,b]$

and (iii) $\max_{a\leq x\leq b}|f^{(2n)}(x)| \leq M_{2n} < \infty$

Then under the above restrictions, the error functional $E_n(f)$ associated with the Gaussian quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=1}^{n} a_j\, f(x_j) + E_n(f) \qquad (3.35)$$

has a representation [14,p.109]

$$E_n(f) = \int_a^b K_{2n-1}(t)\, f^{(2n)}(t)dt \qquad (3.36)$$

where $$K_{2n-1}(t) = E_n\left(\frac{(x-t)_+^{2n-1}}{(2n-1)!}\right), \quad (x-t)_+^{2n-1} = \begin{cases} (x-t)^{2n-1} & x \geq t \\ 0 & x < t \end{cases} \qquad (3.37)$$

From equations (3.33) and (3.35) the Peano kernel $K_{2n-1}(t)$ depends on the weight function $w(x)$ and

$$K_{2n-1}(t) = \int_a^b \frac{w(x)(x-t)_+^{2n-1}}{(2n-1)!}dx - \sum_{j=1}^{n} a_j \frac{(x_j-t)_+^{2n-1}}{(2n-1)!} \qquad (3.38)$$

For the case when the kernel function is of constant sign over $[a,b]$, the application of the mean-value theorem for integrals yields

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} E_n(x^{2n}) \quad a < \xi < b \qquad (3.39)$$

for which an error bound

$$|E_n(f)| \leq \frac{M_{2n}}{(2n)!} |E_n(x^{2n})| \qquad (3.40)$$

is obtainable by virtue of condition (iii). Error expressions of the form (3.37) for the most common Gaussian rules are given in tables (3.1) and (3.2). If $K_{2n-1}(t)$ has a varying sign in [a,b] then an upper bound of the form

$$|E_n(f)| \le e_{2n-1} M_{2n} \quad , \quad e_{2n-1} = \int_a^b |K_{2n-1}(t)| dt \tag{3.41}$$

can be calculated.

The above error bounds (3.38) and (3.39) hold whenever conditions (i), (ii), and (iii) of page 58 are satisfied over the finite, semi-finite, or infinite interval [a,b]. In certain cases, sharper estimates can be obtained from Peano's theorem. For example, under less stringent conditions such as $|f^{(2n)}(x)| \le \gamma x^\beta$ for $x \ge x_n$ and $\max_{a \le x \le x_n} |f^{(2n)}(x)| \le M_{2n}$, Stroud and Chen [49] show that the n-point Gauss-Laguerre quadrature formula has an error bound of the form

$$|E_n(f)| \le \left[\frac{(n!)^2}{(2n!)} - e^{-x_n} \right] M_{2n} + \gamma \Gamma(\beta+1, x_n)$$

where $\Gamma(\beta+1,x_n) = \int_{x_n}^{\infty} t^\beta e^{-t} dt$ is the incomplete Gamma function and x_n is the largest root of the Laguerre polynomial of degree n. The resulting error bound applies to a wider class of functions but the practical use of their estimate depends on whether or not one can obtain suitable constants M_{2n}, γ and β for the error bounds.

If it happens that finite bounds for low-order or even high-order derivatives of the integrand cannot be established, then error bounds derived from Peano's theorem are not useful and derivative-free error estimates must be obtained. Such error bounds can be calculated by using Hilbert spaces. Letting $\xi_\rho\ (\rho>1)$ designate the ellipse having foci at $z = \pm 1$ and semi-major and semi-minor axes $\alpha = 1/2(\rho+1/\rho)$ and $\beta = 1/2(\rho-\rho^{-1})$ respectively, and letting $A(\xi_\rho)$ denote the family of functions analytic on $[-1,1]$ and continuably analytic so as to be single-valued and regular in the closure of ξ_ρ, then the completion of $A(\xi_\rho)$ under the norm

$$||f|| = (f,f)^{1/2} = \left(\iint_{\xi_\rho} f(z)\overline{f(z)}dxdy\right)^{1/2} \tag{3.42}$$

generates a Hilbert space $H^2(\xi_\rho)$. An orthonormal basis for $H^2(\xi_\rho)$ is the sequence of orthonormal Chebyshev polynomials of the second kind [12, p. 241]

$$U_n^*(z) = \frac{2\sqrt{\frac{n+1}{\pi}}\ \sin((n+1)\arccos(z))}{\sqrt{\rho^{2n+2}-\rho^{-2n-2}}\ \sqrt{1-z^2}}. \tag{3.43}$$

Then for an n-point Gaussian quadrature rule over $[-1,1]$ the error functional E_{G_n} is a bounded linear functional over $H^2(\xi_\rho)$ and by the Schwartz inequality we get [12, p.241]

$$|E_n(f)| \le ||E_{G_n}||\ ||f|| \tag{3.44}$$

where $||E_{G_n}||^2 = \frac{4}{\pi}\sum_{k=2n}^{\infty}(k+1)\frac{|E_{G_n}(U_k)|^2}{\rho^{2k+2}-\rho^{-2k-2}}$,

$$U_n(z) = \frac{\sin((n+1)\arccos(z))}{\sqrt{1-z^2}} \tag{3.45}$$

and $$||f|| \leq 1/2\sqrt{\pi(\rho^2-\rho^{-2})}\; M_\rho(f),\quad M_\rho(f) = \max_{z\in\xi_\rho} |f(z)| \,. \tag{3.46}$$

Conveniently the summation (3.45) reduces to a simple form for special cases of the weight function $w(x)$. If the weight function is $w(x) = \frac{1}{\sqrt{1-x^2}}$ then we have a Gauss-Chebyshev quadrature of the first kind and from the results of Chawla [8]

$$|E_n(f)| \leq \frac{2\pi\sqrt{2n+2(1-\rho^{-4})^{-1}}}{\sqrt{1-\rho^{-4}}\,\sqrt{\rho^{4n}-\rho^{-4n}}}\; M_\rho(f) \quad . \tag{3.47}$$

For the weight function $w(x) = \sqrt{1-x^2}$, the Gauss-Chebyshev quadrature rule of the second kind has an error bound of the form [8]

[8]

$$|E_n(f)| \leq \frac{\pi}{2}\;\frac{\sqrt{2n+2\,(\rho^4+1)}}{\sqrt{\rho^{2n+2}-\rho^{-2n-2}}}\; M_\rho(f) \quad . \tag{3.48}$$

For other weight functions on $[-1,1]$, the boundedness of E_{G_n} in $H^2(\xi_\rho)$ permits the calculation of an error bound of the form (3.44) subject to the restriction of f being a member of $A(\xi_\rho)$. Then, even though the summation in (3.45) may not be easily calculated, appropriate bounds on the factors can be used to obtain a bound for $||E_n||$. Therefore, in the case of general applications, the usefulness of this method in obtaining an error estimate depends on the availability of bounds for $||E_{G_n}||$.

If bounds for the summation in (3.45) cannot be easily determined, then derivative-free error estimates for $E_n(f)$ can be derived by considering contour integral representation of $E_n(f)$. To establish a contour integral representation for $E_n(f)$ associated with the Gaussian quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=1}^{n} a_j f(x_j) + E_n(f) \tag{3.49}$$

we consider a closed contour C containing $x_1, \ldots, x_n$ and the points a and b in its interior. Then for any function $f(z)$ analytic in C we have [14, p. 118]

$$E_n(f) = \frac{1}{2\pi i}\int_C \frac{f(z)v_n(z)}{p_n(z)}\,dz \tag{3.50}$$

where

$$v_n(z) = \int_a^b \frac{p_n(t)\,w(t)}{z-t}\,dt \tag{3.51}$$

with $p_n(x)$ being the polynomial of degree n orthogonal to $w(x)$ on the interval $[a,b]$. By obtaining upper and lower bounds on $v_n(z)$ and $p_n(z)$ respectively, an upper bound for $E_n(f)$ of the form

$$|E_n(f)| \le e_{2n-1}M_c(f) \quad , \quad M_c(f) = \max_{z\epsilon C} |f(z)| \tag{3.52}$$

can be established.

An example of such an estimate is considered by Kambo [26] where upper and lower bounds for $v_n(z)$ and $p_n(z)$ are considered for the case of Gauss-Legendre quadrature. By taking the closed contour C to be the ellipse ξ_ρ given above, the n^{th} degree Legendre polynomial $p_n(z)$ is bounded below by

$$|p_n(z)| \geq \frac{(2n)!}{2^{2n}(n!)^2} \frac{(\rho^2-2)\rho^n}{(\rho^2 - 1)} \tag{3.53}$$

whenever $z \in \xi_\rho$. Furthermore, for $v_n(z) = \int_{-1}^{1} \frac{p_n(t)}{z-t}\,dt$, a lemma proved by Kambo shows that

$$|v_n(z)| \leq \frac{2^{2n+2}(n!)^2}{(2n+1)!\,\rho^n(\rho^2-1)} \tag{3.54}$$

for $z \in \xi_\rho$. Then from (3.50), (3.53), and (3.54) an error estimate for the truncation error associated with an n-point Gauss-Legendre quadrature rule is

$$|E_n(f)| \leq \frac{2^{4n+1}(n!)^4\,(\rho^2+1)}{(2n)!(2n+1)!\,\rho^{2n}(\rho^2-2)} M_\rho(f)$$

provided $f(z)$ is analytic in the interior of the ellipse ξ_ρ, $\rho > 2$.

CHAPTER IV

OPTIMAL QUADRATURE RULES DEFINED BY MINIMAL PROPERTIES

4.1 Quadrature formulae characterized by the minimum norm property

In the previous chapter it was shown that bounds for the error functional associated with certain Gaussian quadrature rules can be established by considering a Hilbert space Ω containing all functions analytic in the interior of the unit circle, B. By using either of the line integral or area integral inner products

$$(f,g) = \frac{1}{2\pi}\int_0^{2\pi} f(e^{i\phi})g(e^{i\phi})d\phi \qquad (4.1)$$

and

$$(f,g) = \iint_B f(z)g(z)dx\,dy \qquad (4.2)$$

respectively, a bound of the form

$$|T_n(f)| \le e_n\,||f||\,, \qquad ||f|| = (f,f)^{1/2} \qquad (4.3)$$

where $e_n = ||T_n|| = \sup_{f\in\Omega} \frac{|T_n(f)|}{||f||}$ can be determined such that e_n depends only on the closed contour C or region B which contains the unit interval [-1,1]. Although bounds of the form (4.3) were given only for Gaussian quadrature rules, such bounds are obtainable for Newton-Cotes type quadrature rules (cf.[13],[26]).

By using the error bound (4.3) for a given Hilbert space it is possible to define an optimal quadrature formula to be one for which the error functional T_n has a minimum norm over the entire family of functions analytic on (-1,1). Then, for fixed n, an optimal quadrature rule would be

$$\int_a^b f(x)dx = \sum_{j=1}^{n} a_{mj} f(x_{mj}) + T_{mn}(f) \tag{4.4}$$

such that $|T_{mn}(f)| \leq e_{mn} \; ||f||$ and $e_n \geq e_{mn}$ for any other choice of weights and abscissae. Such a quadrature rule is said to satisfy the minimal norm property. When the quadrature nodes in (4.4) are preassigned, then the quadrature rule (4.4) is said to satisfy the minimum norm property with respect to the abscissae $x_1, \ldots, x_n$.

If a_{mj} and x_{mj} ($j = 1, 2, \ldots, n$) are the weights and abscissae corresponding to the minimal error norm e_{mn}, then for all functions $f(x)$ analytic in $|x| < 1$ and square integrable on the unit circle,

$$f(x) = \sum_{n=0}^{\infty} b_n x^n \quad , \; |x| < 1$$

and $$||f||^2 = \frac{1}{2\pi} \int_0^{2\pi} |f(e^{i\phi})|^2 d\phi \; = \sum_{n=0}^{\infty} |b_n|^2 < \infty$$

so that the error made in integrating $f(x)$ over $[a,b] = [0,1]$ by using (4.4) is

$$|T_{mn}(f)| = \left|\int_0^1 f(x)dx - \sum_{j=1}^{n} a_{mj} f(x_{mj})\right|$$

$$= \left|\sum_{k=0}^{\infty} b_k\left(\frac{1}{k+1} - \sum_{j=1}^{n} a_{mj} x_{mj}^k\right)\right|$$

$$\leq \left[\sum_{k=0}^{\infty} \left(\frac{1}{k+1} - \sum_{j=1}^{n} a_{mj} x_{mj}^k\right)^2\right]^{1/2} \left[\sum_{k=0}^{\infty} |b_k|^2\right]^{1/2}$$

$$= e_{mn} \, ||f|| \, .$$

Hence the minimization of e_{mn} requires the minimization of

$$e_{mn}^2 = \sum_{k=0}^{\infty} \left(\frac{1}{k+1} - \sum_{j=1}^{n} a_{mj} x_{mj}^k\right)^2 \qquad (4.5)$$

subject to the choice of a_{mj} and x_{mj} ($j = 1, 2, \ldots, n$). To obtain this minimization the derivatives

$$\frac{\partial e_{mn}}{\partial a_{mj}} = 0 \; , \quad \frac{\partial e_{mn}}{\partial x_{mj}} = 0 \quad j = 1, 2, \ldots, n \qquad (4.6)$$

are calculated. Noting the identities

$$\frac{1}{x} \log \frac{1}{1-x} = \sum_{n=0}^{\infty} \frac{x^n}{n+1} \qquad (4.7)$$

and

$$\frac{1}{x(1-x)} - \frac{1}{x^2} \log \frac{1}{1-x} = \sum_{n=0}^{\infty} \frac{nx^{n-1}}{n+1} \qquad (4.8)$$

the derivatives (4.6) yield a nonlinear system of $2n$ equations

$$\frac{1}{x_{mj}} \log \frac{1}{1-x_{mj}} = \sum_{k=1}^{n} \frac{a_{mk}}{1-x_{mj}x_{mk}} \qquad j=1,2,\ldots,n$$

$$\frac{1}{x_{mj}(1-x_{mj})} - \frac{1}{x_{mj}^2} \log \frac{1}{1-x_{mj}} = \sum_{k=1}^{n} \frac{a_{mk}x_{mk}}{1-x_{mj}x_{mk}} \qquad j=1,2,\ldots,n \tag{4.9}$$

for the $2n$ unknowns $a_{m1}, \ldots, a_{mn}, x_{m1}, \ldots, x_{mn}$. The existence of a unique solution for the system of equations (4.9) for fixed n has been established by Eckhardt [15]. Solutions for the system (4.9) and values of e_{mn} for $n = 1(1)5$ have been computed by Eckhardt and are included in Table 4.1.

For the case of the Hilbert space $H^2(B)$ with the area integral inner product (4.2) and B corresponding to the unit circle, Richter [41] extends the results of Eckhardt and shows that for the interval $[a,b] = [-1,1]$ the abscissae are all interior points, the weights are positive, and $\sum_{j=1}^{n} a_{mj} < 2$ if the origin is not one of the abscissae. Under these conditions the resulting quadrature rule is not exact even for constant functions. If one of the quadrature nodes corresponds to the origin then $\sum_{j=1}^{n} a_{mj} = 2$, $\sum_{j=1}^{n} a_{mj}x_{mj} = 0$, and the resulting quadrature rule is exact for all polynomials of degree ≤ 1. For classes of functions which extend the region of analyticity, Barnhill [4] shows that the limiting case of a family of entire functions results in the convergence of the weights and abscissae of the minimum norm quadrature rule to those of the corresponding n-point Gaussian quadrature rule.

When the quadrature nodes are fixed in (4.4) and

n	Quadrature Nodes x_{mj}	Quadrature Weights a_{mj}	Minimal Error Coeff. e_{mn}
1	0.68380262	0.89648004	0.36802
2	0.48470056	0.83640681	0.14096
	0.95579191	0.14739637	
3	0.37187816	0.71983972	0.06276
	0.88486952	0.25057401	
	0.99226589	0.02626972	
4	0.30065913	0.61996725	0.03075
	0.81300049	0.31438445	
	0.97407840	0.05915685	
	0.99833314	0.00567889	
5	0.25191541	0.54068166	0.01622
	0.74719512	0.35069805	
	0.94940738	0.09181342	
	0.99342517	0.01515203	
	0.99958104	0.00142326	

Table 4.1. Quadrature parameters for minimal norm quadrature rules

$0 = a < x_1 < \ldots < x_n < b = 1$, the error coefficient is minimized when the weights a_{mk} $(k = 1, \ldots, n)$ are chosen to be solutions of the linear system

$$\frac{1}{x_j} \log \frac{1}{1-x_j} = \sum_{k=1}^{n} \frac{a_{mk}}{1-x_j x_k} \qquad j = 1,2,\ldots, n. \qquad (4.10)$$

The quadrature rule (4.4) with weights defined by (4.10) is applicable only to those functions $f(x)$ which are analytic on $(-1,1)$. For the case of all preassigned nodes being interior to $[-1,1]$ it can be shown that as the region in which the integrand in (4.4) is analytic extends to include the entire real line, the quadrature weights associated with the minimum norm quadrature rule will converge to those of the Newton-Cotes quadrature rule based on the points $x_1, x_2, \ldots, x_n$.

On the basis of the above development of quadrature rules satisfying the minimum norm property, the question of their practicality is reflected in the significance one places in the following facts. First, the existence and uniqueness of a quadrature satisfying the minimum norm property is dependent on the Hilbert space being considered. The restriction of the class of functions for which the quadrature rule is applicable may be severe enough to prohibit its use in many practical problems. Secondly, the determination of such quadrature rules for a large number of abscissae may be difficult. If it is necessary to solve a system of equations such as (4.9) then the effort expended to determine the quadrature weights and abscissae would be unwarranted. There may also exist the problem of determining

$||f||$ if only a set of ordinate values $\{f(x_1), \ldots, f(x_n)\}$ is given. Finally, because it is known that there exists a connection between minimum norm quadrature rules and the Gaussian and Newton-Cotes type rules, the preferential use of these quadrature rules can be justified.

Those instances for which it may be desirable to use the minimum norm quadrature rules include (i) the integrand having a small norm which is easy to calculate and (ii) the value of e_{mn} is known a priori such that $|T_{mn}(f)| \leq e_{mn} \; ||f||$ is an acceptable value for some small value of n.

4.2 Optimal quadrature formulae characterized by minimum roundoff error

For a given quadrature rule

$$\int_a^b w(x)f(x)dx = \sum_{j=1}^{n} a_j f(x_j) + T_n(f) \qquad (4.11)$$

and an unavoidable error ϵ_j associated with either the experimental determination or calculation of $f(x_j)$ $(j = 1, 2, \ldots, n)$, the total roundoff error $R_n(f)$ for equation (4.11) is

$R_n(f) = \sum_{j=1}^{n} a_j \epsilon_j$. From the Cauchy-Schwartz inequality we have $|R_n(f)| \leq \left(\sum_{j=1}^{n} |\epsilon_j|^2 \right)^{1/2} \left(\sum_{j=1}^{n} |a_j|^2 \right)^{1/2}$ so that the problem of minimizing R_n leads to the determination of a_j $(j=0,1,\ldots,n)$ such that

$$\sum_{j=1}^{n} a_j^2 = \text{minimum} . \qquad (4.12)$$

If it is also required that the quadrature rule (4.11) be exact for constant functions then the added condition

$$\sum_{j=1}^{n} a_j = \int_a^b w(x)dx \tag{4.13}$$

is obtained. Quadrature weights which satisfy (4.12) and (4.13) simultaneously also satisfy the system of equations

$$\frac{d}{da_k}\left\{\sum_{j=1}^{n} a_j^2 - \lambda\left[\sum_{j=1}^{n} a_j - \int_a^b w(x)dx\right]\right\} = 0 \ , \ k=1,2,\ldots,n \tag{4.14}$$

where λ is some nonzero constant. Solving (4.14) yields $2a_k - \lambda = 0$ $(k = 1,2,\ldots,n)$ so that all the weights are equal and in view of (4.13), $a_k = \frac{1}{n}\int_a^b w(x)dx$. The resulting quadrature rule for minimum roundoff error is

$$\int_a^b w(x)f(x)dx = C_n \sum_{j=1}^{n} f(x_j) \ ,$$
$$C_n = \frac{1}{n}\int_a^b w(x)dx \tag{4.15}$$

and is referred to as being a Chebyshev quadrature formula.

The Chebyshev quadrature rule given by (4.15) is exact for at least all polynomials of degree less than one and is defined whenever the weight function is integrable over the interval [a,b]. When the quadrature nodes x_j $(j = 1,2,\ldots,n)$ are preassigned then it is likely that the highest degree of precision has been reached.

On the other hand, if the n quadrature nodes are treated as parameters then it may be possible to increase the degree of precision of the quadrature rule to n. To obtain this level of precision the quadrature nodes must be chosen such that the system of equations

$$\sum_{j=1}^{n} x_j^k = \frac{1}{C_n} \int_a^b x^k w(x)dx \quad k = 1,2,\ldots,n \tag{4.16}$$

is satisfied.

The solution set for the system (4.16) is usually obtained indirectly by solving for the roots of the polynomial

$$p_n(x) = \prod_{j=1}^{n} (x-x_j) = x^n + A_1x^{n-1} + A_2x^{n-2} + \ldots + A_n. \tag{4.17}$$

The connection between the polynomial $p_n(x)$ in (4.17) and the system of equations (4.16) is that of [24, p. 324]

$$\begin{aligned}
A_1 &= -(x_1 + x_2 + \ldots + x_n) = -b_1 \\
A_2 &= x_1x_2 + x_1x_3 + \ldots + x_1x_n + x_2x_3 + \ldots + x_2x_n + x_{n-1}x_n = \\
&\qquad - 1/2(b_2 + A_1b_1) \\
&\;\;\vdots \qquad\qquad\qquad\qquad\qquad\qquad . \quad . \\
A_n &= (-1)^n x_1x_2x_3 \ldots x_{n-1}x_n = -\frac{1}{n}(b_n + A_1b_{n-1} + \ldots + A_{n-1}b_1)
\end{aligned} \tag{4.18}$$

where $b_k = \sum_{j=1}^{n} x_j^k$, $(k = 1,2,\ldots,n)$. Now noting also that

$b_k = \frac{1}{C_n} \int_a^b x^k w(x)dx$ it is possible to calculate sequentially the coefficients A_i $(i = 1,2,\ldots,n)$ for $p_n(x)$. The roots of $p_n(x)$ and hence the corresponding quadrature nodes for (4.15) can be computed numerically by using for example either Newton's or Bairstow's method.

For particular weight functions and certain values of n, it may happen that some or all of the roots are either complex or lie outside the interval of integration. In such instances, the Chebyshev quadrature rule having a precision of order n is said to be nonexistent. It is known, for example, that Chebyshev quadrature formulae of maximum precision exist for the weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$, $1 \le x \le 1$ and for all $n \ge 0$. For such quadrature rules the degree of precision is in fact $2n-1$. On the other hand, for the weight function $w(x) = 1$ there can exist Chebyshev quadrature rules having only a precision of order $n = 1,2,3,4,5,6,7,$ or 9.

Since there are known to be only a few quadrature rules which are defined by (4.15) and satisfy (4.16), there have been some attempts to expand the class of Cheybshev quadrature rules by modifying either or both of the requirements (4.15) and (4.16). As an example of the limitation of the defining characteristic (4.15), the occurrence of an odd (or periodic) weight function and a quadrature interval $[a,b] = [-\alpha,\alpha]$ (or $[a,b] = [o,p]$; p = period of $w(x)$) gives $C_n = 0$. This difficulty has been overcome by defining a Chebyshev quadrature rule for an odd weight function to be

$$\int_a^b w(x)f(x)dx = C_n \sum_{j=1}^{\frac{n}{2}} \{f(x_j) - f(-x_j)\} \tag{4.19}$$

where n is even and $C_n, x_1, x_2, \ldots, x_{\frac{n}{2}}$ satisfy the system of equations

$$C_n \sum_{j=1}^{\frac{n}{2}} x_j^{2k+1} = \int_a^b x^{2k+1} w(x)dx, \quad k = 0,1,\ldots,\frac{n}{2} \tag{4.20}$$

and
$$\sum_{j=1}^{\frac{n}{2}} x_j^{2k} = 0 \qquad k = 1,2,\ldots,\frac{n}{2} \ .$$

The precision of the Chebyshev quadrature rule (4.19) is $n+1$.

Alternately the requirement (4.16) often yields complex quadrature nodes or nodes outside the interval of integration. A slight modification to (4.16) as proposed by Barnhill et al. [5] is to define a Chebyshev quadrature rule by (4.15) subject to the requirement

$$\sum_{k=0}^{n} (C_n \sum_{j=1}^{n} x_j^k - \int_a^b x^k w(x)dx)^2 = \text{minimum} \ . \tag{4.21}$$

Although a high level of precision is not always associated with this type of Chebyshev quadrature, it is possible to obtain fairly accurate quadrature approximations using only a small number of quadrature nodes. For example, the Chebyshev quadrature rule satisfying (4.15) and (4.19) for the weight function $w(x) = 1$ and

$n = 8, 10$ and 11 exists with all quadrature nodes being symmetric about the midpoint of the interval of integration and the midpoint being a multiple node. For $n = 8$ the minimum value associated with (4.21) is 0.79×10^{-6}. The corresponding Chebyshev quadrature rules for $n = 8, 10,$ and 11 satisfying (4.16) do not exist.

4.3 Quadrature formulae with least upper bound error.

If a quadrature rule of the form (4.11) is being considered for the integration of a class of functions F, then the previously mentioned situations of (i) fixed quadrature abscissae and variable weights, (ii) fixed weights and variable abscissae, or (iii) both variable quadrature weights and abscissae may be encountered. Corresponding to these three cases an optimal quadrature rule can be defined as one in which the variable quadrature quantities are chosen such that a least upper bound for the quadrature error is obtained with the least upper bound for the quadrature error being defined by

$$T_{\ell.u.b.,i} = \inf_{\{a_j\}_{j=1}^{n}} \{ \sup_{f \epsilon F} |T_n(f)| \} \tag{4.22}$$

$$T_{\ell.u.b.,ii} = \inf_{\{x_j\}_{j=1}^{n}} \{ \sup_{f \epsilon F} |T_n(f)| \} \tag{4.23}$$

or

$$T_{\ell.u.b.,iii} = \inf_{\{x_j\},\{a_j\}} \{ \sup_{f \epsilon F} |T_n(f)| \} \tag{4.24}$$

respectively. On the basis of these definitions the magnitude of the least upper bound quadrature error will depend significantly on the specified weight function $w(x)$ and the "smoothness" of the family of functions being considered.

The ability to analytically measure the smoothness of the class of functions is important, for it will often enable the specification of the parameters which give a least upper bound to the quadrature error. This fact has been well illustrated by the investigations of Lebed [30] into the determination of optimal quadrature rules which satisfy either (4.22), (4.23), or (4.24). The two classes of functions which Lebed considered are F_δ and $F_{\delta,n}$. A function $f \epsilon F_\delta$ satisfies $|f(x_2)-f(x_1)| \leq \delta\ (x_2-x_1)$ for any x_1, x_2 $\epsilon[a,b]$ and $a \leq x_1 < x_2 \leq b$ with $\delta(x)$ being the modulus of continuity. The class $F_{\delta,n}$ depends on a given set of abscissae $\{x_1, x_2, \ldots, x_n\}$ where $a \leq x_1 < x_2 < \ldots < x_n \leq b$ and $f \epsilon F_{\delta,n}$ if $|f(x)-f(x_k)| \leq \delta\ (|x-x_k|)$, $\beta_k \leq x \leq \beta_{k+1}$ $(k = 1,2,\ldots,n)$ and $\beta_1 = a$, $\beta_k = \dfrac{x_{k-1}+x_k}{2}$ $(k = 2,\ldots,n)$, $\beta_{n+1} = b$. It is noted that $F_\delta \subset F_{\delta,n}$ for any set of abscissae and for any n. Without loss of generally it is assumed that $[a,b] = [0,1]$.

If the modulus of continuity $\delta(x)$ is known for all functions in F_δ or $F_{\delta,n}$ then an optimal quadrature formula satisfying (4.22) with respect to the fixed set of abscissae $\{x_1,x_2,\ldots,x_n\}$ can be determined. The dependence of the quadrature rule on the weight function, modulus of continuity, and specified abscissae is given by the following theorem due to Lebed

<u>Theorem 4.1.</u> The optimal quadrature rule satisfying (4.22) for the fixed quadrature nodes $x_1, x_2, \ldots x_n$ has quadrature weights

$$a_k = \int_{\beta_k}^{\beta_{k+1}} w(x)dx \quad k = 1,2,\ldots,n \tag{4.25}$$

and
$$T_{\ell.u.b.,i} = \sum_{k=1}^{n} \int_{\beta_k}^{\beta_{k+1}} w(x)\ \delta\ (|x-x_k|)dx. \tag{4.26}$$

Corresponding to case (iii) of this section, Lebed shows that the optimal quadrature rule satisfying (4.24) with respect to the class of functions $F_{\delta,n}$ has quadrature abscissae $x_1^*, x_2^*, \ldots, x_n^*$ which minimize the function

$$m(x_1,x_2,\ldots,x_n) = \sum_{k=1}^{n} \int_{\beta_k}^{\beta_{k+1}} w(x)\ \delta\ (|x-x_k|)dx \tag{4.27}$$

and quadrature weights

$$a_k = \int_{\beta_k^*}^{\beta_{k+1}^*} w(x)dx \qquad k = 1,2,\ldots,n \tag{4.28}$$

where β_k $(k = 1,2,\ldots,n)$ is as defined above and $\beta_1^* = a$, $\beta_k^* = \dfrac{x_{k-1}^* + x_k^*}{2}$ $(k = 1,2,\ldots,n)$, $\beta_{n+1}^* = b$. The least upper bound quadrature error in this case is

$$T_{\ell.u.b.,iii} = \sum_{k=1}^{n} \int_{\beta_k^*}^{\beta_{k+1}^*} w(x)\ \delta\ (|x-x_k^*|)dx. \tag{4.29}$$

CHAPTER V

THE APPROXIMATION OF IMPROPER INTEGRALS

5.1 Integration over infinite intervals.

A convergent improper integral having an infinite quadrature interval may have any one of the forms

$$I_1(f) = \int_a^{\infty} w(x)f(x)dx \ , \quad |a| < \infty \tag{5.1a}$$

$$I_2(f) = \int_{-\infty}^{b} w(x)f(x)dx \ , \quad |b| < \infty \tag{5.1b}$$

or

$$I_3(f) = \int_{-\infty}^{\infty} w(x)f(x)dx \tag{5.1c}$$

with the integrand $w(x)f(x)$ being bounded and integrable over the quadrature interval. Each of the above improper integrals is a special problem in numerical integration because any numerically computed quadrature approximation can be based on at most a finite number of function evaluations with all quadrature abscissae being located in a finitely bounded interval. Except for the cases of (5.1a) with weight function $w(x) = e^{-x}$ and (5.1c) with weight function $w(x) = e^{-x^2}$, none of the standard quadrature methods can generate accurate quadrature approximations for a large class of functions.

If any of the conventional Newton-Cotes or Gaussian quadrature rules are to be used to approximate an integral having infinite limits, then one approach is to make a transformation of the independent variable so that the integral will be reduced to one having removable singularities at one or both endpoints of a finite interval. Often many transformations can be used but the one preferred will be that transformation which results in the integrand being least complicated, both computationally and structurally. For (5.1a) a transformation can be chosen such that

$$I_1(f) = \int_{-1}^{1} v_1(y)g_1(y)dy \tag{5.2a}$$

where $v_1(y) = \dfrac{1}{(1-y)^2} \; w\left(\dfrac{(1-a)y+(1+a)}{1-y}\right)$ and $g_1(y) = f\left(\dfrac{(1-a)y+(1+a)}{1-y}\right)$.

A suitable transformation for (5.1c) results in

$$I_3(f) = \int_{-1}^{1} v_3(y)g_3(y)dy \tag{5.2b}$$

where $v_3(y) = \dfrac{1+y^2}{(1-y^2)^2} \; w\left(\dfrac{y}{1-y^2}\right)$ and $g_3(y) = f\left(\dfrac{y}{1-y^2}\right)$. Techniques for handling the singularities in (5.2a) and (5.2b) are discussed in Section 5.2.

Alternately by noting that the theoretical definitions of (5.1a) and (5.1c) are

$$I_1(f) = \lim_{x\to\infty} \int_{a}^{x} f(t)dt$$

and

$$I_3(f) = \lim_{x\to\infty} \int_{-x}^{0} f(t)dt + \lim_{x\to\infty} \int_{0}^{x} f(t)dt$$

respectively then it is reasonable to derive quadrature approximations based on finite interval quadrature rules by ignoring the "tail" of the integral. For the case of (5.1a), Davis and Rabinowitz [14, p.90] suggest the integral be approximated by a sum of integrals

$$I_1(f) \approx I_{1,n}(f) = \sum_{i=0}^{n} \int_{r_i}^{r_{i+1}} f(x)dx \qquad (5.3)$$

where $\{r_0, r_1, \ldots\}$ is a sequence of numbers converging to infinity. In practical applications, the upper limit n is chosen such that

$\int_{r_n}^{r_{n+1}} f(x)dx \leq \epsilon$ and the sequence $\{r_i\}_{i=0}^{n}$ is a geometric sequence such as $\{2^i+a\}_{i=0}^{n}$, for example. For each integral in the sum (5.3), a closed-type Newton-Cotes or Gaussian quadrature rule is applied. A more direct approach proposed by Isaacson and Keller [25, p.350] is to choose some large number L such that the improper integral becomes a sum of two integrals, one of which is proper,

$$I_1(f) = \int_a^L f(x)dx + \int_L^\infty f(x)dx = I_{1,L} + I_{1,\infty} .$$

The integral $I_{1,L}$ is approximated by an interpolatory quadrature rule, either of a simple or compound nature, and the integral $I_{1,\infty}$ is transformed into an integral similar to (5.2a). The parameter L is chosen such that the value of $I_{1,\infty}$ is significantly smaller than $I_{1,L}$.

If it is known that the integrand in (5.1a) is monotonic and $\lim_{x\to\infty} f(x) = 0$ then a simple Reimann sum can be used to approximate the improper integral. The fact that a

Riemann sum will suffice is based on the result that the monotonicity of the integrand gives [14,p.91]

$$0 \le \int_a^\infty f(x)dx - h \sum_{k=1}^\infty f(a+kh) \le h\, f(a) \qquad (5.4)$$

for any h and $f(a) \ge 0$. From (5.4) one obtains

$$\lim_{h\to 0} h \sum_{k=1}^\infty f(a+kh) = \int_a^\infty f(x)dx.$$

Since use of this method on a digital computer will require the Riemann sum to have a finite number of terms this method is just another variation of the technique of ignoring the "tail" of the integrand. The advantages of this approach, however, are that the quadrature error is readily available and the quadrature approximation can be easily computed. A disadvantage of the Riemann sum method is that any sequence of quadrature approximations will converge slowly.

A somewhat different and in fact more direct approach than any of the above techniques for the approximation of either (5.1a) or (5.1c) is to modify the integrand so that the appropriate Gaussian quadrature weight function appears as part of the integrand. The required modifications could typically be

$$\int_0^\infty w(x)f(x)dx = \int_0^\infty e^{-x}\, g_1(x)dx$$

and

$$\int_{-\infty}^{\infty} w(x)f(x)dx = \int_{-\infty}^{\infty} e^{-x^2} g_3(x)dx$$

where $g_1(x) = e^x w(x)f(x)$ and $g_3(x) = e^{x^2} w(x)f(x)$. The obvious disadvantage in using this method is that $g_1(x)$ and $g_3(x)$ will more than likely be unbounded so that the standard expressions for the Gaussian quadrature errors will not be useful.

5.2 The integration of singular integrands.

A different type of improper integral to that considered in section 5.1 involves a convergent integral for which the integrand is unbounded at one or more points within a finite quadrature interval. As noted in the last section the connection of this type of improper integral with that of an improper integral having an infinite quadrature interval is made through a transformation of variables. However there is a distinction to be made in the problem of choosing a quadrature method for the two types of improper integrals. Aside from the usual question of truncation and roundoff errors, the problem of choosing an approximation method for integrals with singular integrands is compounded by the fact that the type of singularity (either logarithmic or algebraic of a fixed degree) affects the degree of convergence of the quadrature method [16]. The other consideration for approximation of integrals with singular integrands is that some of the commonly used techniques are theoretically unjustified, either for some particular class of integrands or for any integrand.

A typical improper integral with singular integrand will be of the form

$$I(f) = \int_a^b f(x)dx \tag{5.5}$$

where $-\infty < a < b < \infty$ and $f(x)$ is continuous for all x in either of the half-open intervals $(a,b]$ or $[a,b)$ or the open interval (a,b). If singularities appear in the interval (a,b) the improper integral can be transformed into a sum of improper integrals with singularities at one or both of the endpoints of the quadrature subintervals. Without loss of generality it can be assumed that the interval $[a,b]$ corresponds to the interval $[-1,1]$.

An investigation into the techniques of approximating improper integrals such as (5.5) reveals the following methods for handling the singularities; (i) change of variable, (ii) elimination of singularity by subtraction, (iii) use of integration formulae of the interpolatory type with the singularity (or singularities) being absorbed into a fixed weight function, (iv) proceeding to the limit, (v) removal of singularity, and (vi) ignoring the singularity.

For the first technique, an algebraic singularity is removed by a transformation of variables, thus converting the improper integral into a proper integral. The proper integral which arises from the transformation can be approximated by any of the standard interpolatory quadrature rules. Such an approach would work for integrals of the form

$$\int_0^1 x^{-1/n} f(x)dx = n \int_0^1 f(t^n)\, t^{n-2}dt$$

where $n > 1$ and $f(x)$ has no singularities in the interval $[0,1]$.

The method of subtracting out the singularity involves choosing a function $g(x)$ such that $f(x)-g(x)$ is bounded and $g(x)$ has a closed form integral. A frequently quoted example for this method is

$$\int_0^1 \frac{\cos(x)}{\sqrt{x}}\,dx = \int_0^1 \frac{dx}{\sqrt{x}} + \int_0^1 \frac{\cos(x)-1}{\sqrt{x}}\,dx = 2 + \int_0^1 \frac{\cos(x)-1}{\sqrt{x}}\,dx \;.$$

The integrand $\frac{\cos(x)-1}{\sqrt{x}}$ is bounded at the origin because

$\lim_{x\to 0^+} \frac{\cos(x)-1}{\sqrt{x}} = 1$. The third technique mentioned will be applied to integrals of the form $\int_a^b w(x)f(x)dx$ where $w(x)$ has a singularity at one or both of the quadrature limits. The only requirement for the use of $w(x)$ as a weight function for interpolatory quadrature rules is that $\int_a^b w(x)x^k dx$ $(k \geq 0)$ must exist.

If the integral (5.5) has a singularity at $x = a$ then by definition $\lim_{x\to a^+} \int_x^b f(x)dx = \int_a^b f(x)dx$ if the limit exists. For a convergent integral with singularity at $x = a$ a quadrature approximation can be obtained by either truncating the quadrature interval or else by computing a sum of integrals. For the former approach, a sufficiently small number r is chosen such that

$I(f) \approx \int_r^b f(t)dt$ and $\int_a^r f(t)dt \leq \epsilon$ for some prescribed error level ϵ. The integral for the truncated interval $[r,b]$ would be approximated by an interpolatory quadrature rule. The other approach gives $I(f) \approx \sum_{i=0}^{n} \int_{r_{i+1}}^{r_i} f(t)dt$ where $b = r_0 > r_1 > r_2 > \ldots$ is a sequence of numbers converging to $x = a$. The limit n is usually chosen so that $\int_{r_{n+1}}^{r_n} f(t)dt < \epsilon$, and all integrals in the sum are approximated by closed-type interpolatory quadrature formulae. These two methods are essentially based on the idea of "proceeding to the limit". Analogous procedures can be used for integrands with a singularity at $x = b$.

The method of removing the singularity will be used for integrals in which there is an apparent or removable singularity in the integrand. The singularity is "removed" by defining the integrand to be a finite (usually limiting) value. An example of such an integral is $\int_0^1 \frac{\sin(x)}{x} dx$, where in effect the integrand $f(x) = \frac{\sin(x)}{x}$ is bounded for all x because $\lim_{x \to 0} \frac{\sin(x)}{x} = 1$. By defining $f(0) = 1$ a closed type quadrature rule could be used to approximate the integral.

The method of ignoring the singularity is perhaps the simplest but at the same time most hazardous approach to generating an approximation to an improper integral having singularities. Essentially the method is one of assigning a finite value (usually 0) to the integrand for any singularity which appears at the end-point of the quadrature interval. Then either a sequence of open or closed type interpolatory quadrature formulae are used to generate a (hopefully convergent) sequence of quadrature approximations. The reason why this method is a hazardous one, is that absurd results can be sometimes obtained [13]. Conditions required for the convergence of the sequence of quadrature approximations are described in [13], [38], and [35] with the degrees of convergence for integrands having either logarithmic or algebraic singularities for different sequences of approximations being given in [16].

CHAPTER VI

COMPUTER ALGORITHMS FOR NUMERICAL INTEGRATION

6.1 The advantages and limitations in using a digital computer for numerical integration.

The arithmetical computations required of any numerical quadrature method may be performed by hand or else by using either a desk calculator or digital computer. For each of these computational methods any quadrature approximation computed by using floating-point arithmetic will have an associated truncation error due to the finite representation of constants and results of intermediate calculations. This type of error is usually several orders of magnitude less than the truncation error associated with the quadrature rule and poses no serious problem.

When a digital computer is used to compute a quadrature approximation, there is an additional error, called roundoff error, due to the representation of a decimal number in terms of a finite binary fraction. The magnitude of the roundoff error depends on the base of the number system, the number of bits used to store the binary fraction, and the rounding procedure used to obtain the binary floating-point approximation (cf. [53], [10]). If several calculations are made, this roundoff error is magnified and can become a problem of such magnitude as to "destroy an efficient and sound computational procedure predicted on arithmetic in the real number field." [33] The problem is further compounded by the fact that the roundoff error for a quadrature method cannot be computed a priori without considerable difficulty. Thus the two factors of limited accuracy and uncertainty of the "correctness" of the computed quadrature approximation present the most severe limitation in using a digital computer for the calculation of a quadrature approximation.

A second limitation in using a digital computer for numerical quadrature concerns the unadaptable nature of a digital computer. In the practical use of a digial computer there can be only a limited amount of human intervention in the computational process. Therefore any tests for undesirable characteristics such as growth of roundoff error,nonconvergence or even spurious convergence in the quadrature approximations must be incorporated in the computer algorithm. Any efficiency techniques such as the use of low-order quadrature rules for regions where the integrand is very smooth and the avoidance of redundant function evaluations for compound quadrature rules add to the complexity of a computer algorithm for numerical quadrature. As a result the most accurate, efficient, and infallible quadrature algorithms are long and complex.

In spite of the above mentioned difficulties,there are distinct advantages in using a digital computer for the calculation of quadrature approximations. The most notable advantage is the short time in which an approximation or even sequence of approximations can be computed. What may take minutes or even hours of human effort to calculate a sequence of quadrature approximations may require only a few seconds of computer processing time. Secondly, there is little chance of computational mistakes being made when using a digital computer. In performing arithmetical computations by hand or else by using a desk calculator, there is a good chance of human error due to miscalculation or wrong entry of numbers. Except for machine failure, such gross errors will not occur when using a digital computer. The third advantage is that a computer algorithm for the integration of a large class of integrals can be coded in a computer language which is virtually machine-independent.

Therefore, there can be an almost universal implementation of a computer algorithm for numerical quadrature with the algorithm being usable by those who are either well-acquainted or only cursorily familiar with numerical quadrature methods.

6.2 The characterization of an ideal automatic integrator

An automatic integrator is defined as being a computer algorithm for the numerical approximation of an integral with user input consisting of a pair of integration limits, a function routine for the integrand or a set of function values, an error tolerance level, and usually an integer representing some computational attribute which, if exceeded, causes the termination of the quadrature process. Output from the algorithm consists of the "best" quadrature approximation and possibly a measure of the quadrature error. Automatic integrators are classified as being adaptive or non-adaptive and can be either iterative or non-iterative. For an adaptive quadrature scheme, the quadrature abscissae are chosen on the basis of the behaviour of the integrand over various segments of the quadrature interval. In an interative scheme, successive quadrature approximations are computed until two successive approximations agree to a specified number of digits or else until a maximum number of quadrature approximations have been computed.

The qualities associated with a successful automatic integrator centre on the practical aspects of numerical integration. One of these attributes concerns the prevention and detection of the roundoff errors. It is desirable for a quadrature scheme to be based on a quadrature rule or sequence of quadrature rules having quadrature weights which are positive rational numbers, for these rules help prevent the growth of such errors. However, as figure 6.1 indicates, the choice of such rules does not guarantee that roundoff error will not accumulate. Therefore there also should be a capability in the quadrature scheme for the detection of accumulating roundoff error.

Secondly, in the interest of avoiding unnecessary calculations, the computer algorithm should be able to detect and report non-convergence or even slow convergence of the quadrature approximations after calculating only a few quadrature approximations. Figure 6.2 illustrates that there is no advantage in doing more than thirty-one or even sixty-three function evaluations, for no significant gain in accuracy of the quadrature approximations is realized. Another problem of this type concerns the possibility of spurious convergence; that is, the convergence of successive quadrature approximations to the wrong number. Although the numerical detection of this fault is almost impossible, it is desirable for a quadrature scheme to be based on quadrature rules which are unlikely to generate such approximations.

A third characteristic of a good automatic integrator is a lack of complexity. Here complexity is construed to include

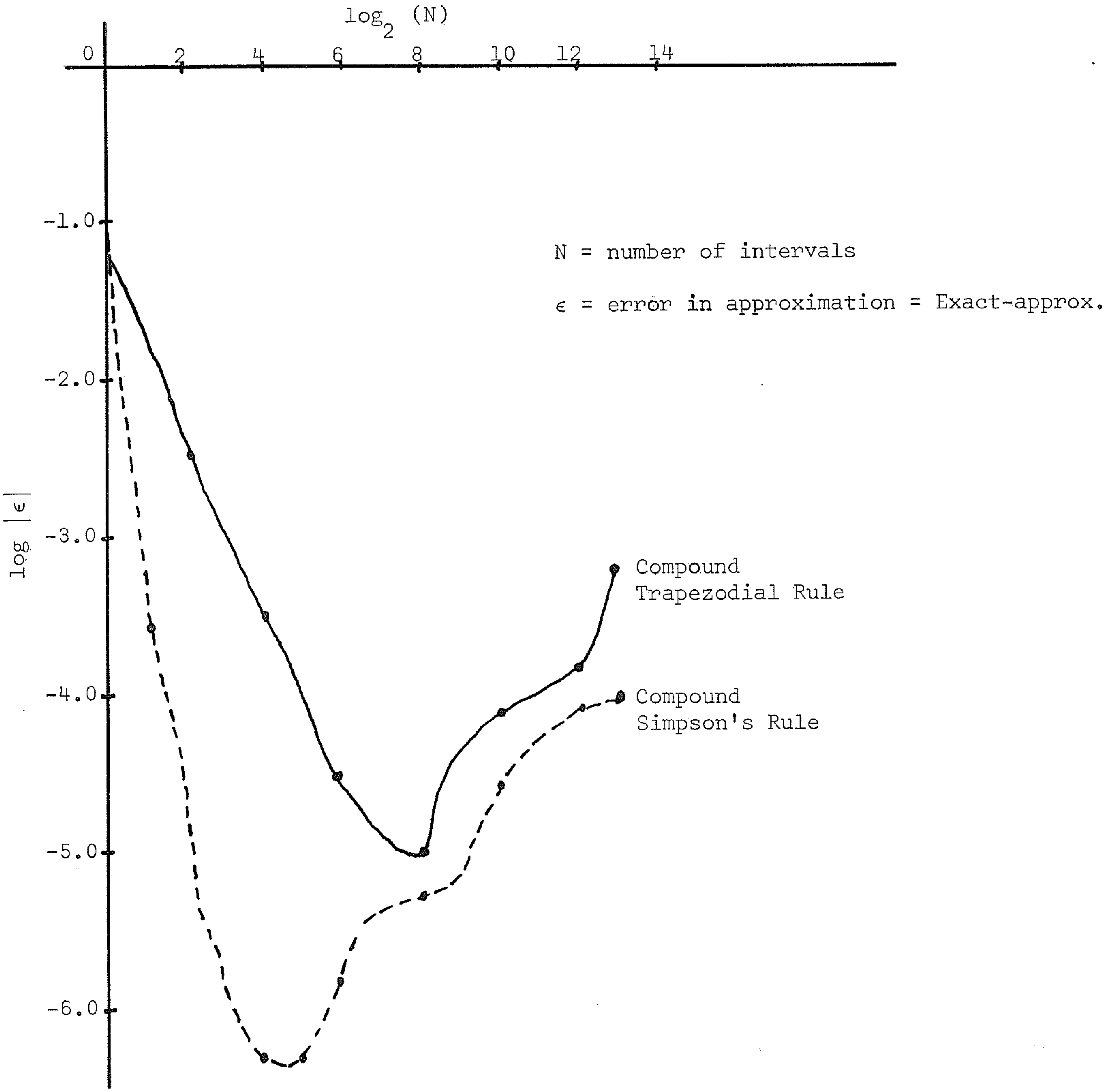

Figure 6.1

Error in Numerical Approximation of $\int_0^1 e^{-x^2}dx = \frac{\sqrt{\pi}}{2}$ erf(1.0)

Number of Function Evaluations	Relative Quadrature Errors
3	9.2×10^{-3}
7	7.2×10^{-3}
15	6.5×10^{-3}
31	6.2×10^{-3}
63	6.0×10^{-3}
127	5.9×10^{-3}
255	5.9×10^{-3}

Figure 6.2

The approximation of $\int_0^1 e^{.01\left|x + \frac{\sqrt{3}}{20}\right|}dx$ using the midpoint rule.

rule.

both computational and algorithmic complexity. Computational complexity concerns the number of computations required to generate a quadrature approximation and the occurrence of irrational quadrature weights and abscissae. The problem of algorithmic complexity concerns the coding of the computing program. One aspect of this problem is the avoidance of redundant function evaluations when using compound quadrature rules such as those of the Newton-Cotes type. The second form of algorithmic complexity concerns the generation of successive quadrature approximations and the procedure used to terminate the quadrature process. This type of difficulty ranges from the simple termination after two approximations differ in absolute value less than some specified number, to the recursive terminating procedure proposed by McKeeman [14, p.165].

Finally, a good automatic integrator should give some indication of the error in the resulting quadrature approximation. The fact that a sequence of quadrature approximations is terminated when either

$$|A_{m+1}(f) - A_m(f)| \leq \epsilon \tag{6.1}$$

or

$$|A_{m+1}(f) - A_m(f)| \leq \epsilon \, |A_{m+1}(f)| \tag{6.2}$$

for two successive quadrature approximations $A_{m+1}(f)$ and $A_m(f)$ and for some preassigned tolerance level ϵ, does not guarantee that the quadrature approximation $A_{m+1}(f)$ has a quadrature error less than or equal to ϵ. Numerous examples ([11], [14,p.161]) can be given to illustrate successive approximations satisfying either (6.1) or (6.2) while in fact having errors in the approximation several orders of magnitude greater than ϵ. The automatic integrator, after having terminated the quadrature procedure, should attempt to estimate the quadrature error on the basis of theoretical information of the integrand. One fairly successful attempt in this direction has been the approximation of successive derivatives of the integrand by finite differences and the use of such approximations in computing a theoretical estimate of the quadrature error on the basis of formulae such as those given in Table 2.1.

In attempting to satisfy the above requirements for a good automatic quadrature scheme, there has been a trend towards the development of iterative, adaptive methods based on low-order quadrature formulae. One reason for the selection of low-order formulae, is that by using a high-order quadrature rule, an improvement in the quadrature approximation requires the application of at least one other quadrature rule. Since formulae of different orders do not usually have common nodes and quadrature weights, the method is inefficient. Even if accurate results are obtained there is the problem of estimating quadrature errors directly when the expressions involve high-order derivatives. Another reason for not using high-order formulae is that such quadrature rules usually have weights and abscissae which are irrational numbers. In the case of high-order Newton-Cotes quadrature rules there is the problem of negative quadrature weights. Therefore the requirement of minimal roundoff error necessitates the rejection of such rules. The iterative aspect is beneficial because some information can be extracted about the convergence of the quadrature approximations. Finally, adaptive quadrature methods often have the ability to concentrate on regions where the integrand is badly behaved. When the integrand has erratic behaviour over one or more regions more accurate results can be obtained by using adaptive methods.

Based on the reasons given above, adaptive quadrature methods using low-order formulae are to be preferred in automatic integration. This is not to mean that adaptive quadrature methods have to use only low-order quadrature rules. Oliver [37] has claimed that many of the proposed disadvantages of using high-order quadrature rules in adaptive schemes are either irrelevant or else only of minor significance and as such do not warrant the exclusion of high-order formulae. Further support for his view is drawn from the fact that there have not been any conclusive numerical experiments which illustrate these undesirable characteristics.

6.3 Some computer programs for numerical quadrature

The automatic integrators characterized in Section 6.2 are translations of mathematically defined algorithms into computer source programs. Usually the computer source programs are written in either FORTRAN IV or ALGOL 60 since these languages are well suited for programs involving numerical computations and to a large extent are machine-independent. Also, in using ALGOL 60, there is an added benefit arising from the structure of the source program. By using nested blocks it is possible to write a compact program with almost no branching. This capability allows for better program readability.

As mentioned in the last section, an automatic integrator often arrives at an approximation to an integral when either two successive approximations differ in absolute value less than some prescribed tolerance or else when a certain number of interval subdivisions or function evaluations are made. Based on that stopping criteria, the automatic integrators are function procedures having a formal parameter list consisting of one or more of the following entities; (i) the integration limits, (ii) an error tolerance level, (iii) a procedure declaration for the integrand, and (iv) a number giving the maximum number of interval subdivisions or function evaluations. If recursion is used or if arrays for the abscissae and ordinate values are given then (iv) corresponds to the maximum level of recursion or the number of elements in each array. Some computer programs illustrating these characteristics are given below. All computer programs are written using ALGOL W [48].

One of the most successful and widely used automatic integrators is that due to McKeeman [32]. The automatic integrator is a recursive adaptive quadrature method based on Simpson's rule with the range (and each subrange) of integration being divided into subintervals of progressively smaller widths. Subdivision is stopped if the absolute value of the relative differences between successive approximations is less than a specified allowable tolerance or if the level of recursion reaches twenty. The reason for taking this approach is that there is a minimization in the number of function evaluations over a given subrange of integration. The minimum number of quadrature abscissae used is nineteen and the number of abscissae is always of the form $7 + 12k$ $(k \geq 1)$. The only input requirements are the integration limits, a relative error tolerance level, and a procedure for the integrand. A source program for this algorithm is

```
REAL PROCEDURE INTEGRAL(REAL VALUE A,B,EPS; REAL PROCEDURE FCT);
BEGIN  REAL ABSAREA;  INTEGER LEVEL;
REAL PROCEDURE SIMPSON( REAL VALUE A,DA,FA,FM,FB,ABSAREA,EST,EPS;
                         REAL PROCEDURE FCT                        );
  BEGIN
  REAL DX,X1,X2,EST1,EST2,EST3,F1,F2,F3,F4,SUM,APPROX;
  DX:=DA/3.0;  X1:=A+DX;  X2:=X1+DX;
  F1:=4.0*FCT(A+DX/2.0);  F2:=FCT(X1);
  F3:=FCT(X2);  F4:=4.0*FCT(A+2.5*DX);
  EST1:=(FA+F1+F2)*DX/6.0;
  EST2:=(F2+FM+F3)*DX/6.0;
  EST3:=(F3+F4+FB)*DX/6.0;
  ABSAREA:=ABSAREA-ABS(EST)+ABS(EST1)+ABS(EST2)+ABS(EST3);
  SUM:=EST1+EST2+EST3;
  LEVEL:=LEVEL+1;
  IF (((ABS(EST-SUM) <= EPS*ABSAREA) AND (EST ¬= 1.0)) OR
  (LEVEL >= 20))    THEN APPROX:=SUM    ELSE
       APPROX:=SIMPSON(A,DX,FA,F1,F2,ABSAREA,EST1,EPS/1.7,FCT)+
               SIMPSON(X1,DX,F2,FM,F3,ABSAREA,EST2,EPS/1.7,FCT)+
               SIMPSON(X2,DX,F3,F4,FB,ABSAREA,EST3,EPS/1.7,FCT);
  LEVEL:=LEVEL-1;
  APPROX
  END;
LEVEL:=1;  ABSAREA:=1.0;
SIMPSON(A,B-A,FCT(A),4.0*FCT((A+B)/2.0),FCT(B),ABSAREA,1.0,EPS,FCT)
END;
```

A non-adaptive quadrature algorith proposed by Bauer [14, p.167] approximates $\int_a^b f(x)dx$ by using Romberg's method.

The integration procedure is terminated when the quadrature error term is of the order 2*ORD+2 (ORD ≥ 0) where ORD is an integer having a value specified by the user. The only other input requirements are the integration limits and a procedure for the integrand. In analyzing this algorithm it has been noted that computation time will approximately double when ORD is increased by one, that roundoff error could become significant, and that it is not foolproof. The conditions under which the algorithm will fail are for either (i) a discontinuous integrand, (ii) an integrand with odd-derivatives which vanish or else are equal at the endpoints of the quadrature interval, or (iii) an error term which contains an odd-power of h where $h = \frac{b-a}{2^n}$ and a and b are the integration limits. A short program for this algorithm is as follows:

```
REAL PROCEDURE ROMBERG(REAL VALUE A,B;    INTEGER VALUE ORD;
                       REAL PROCEDURE FCT            );
BEGIN
  REAL ARRAY T(1::ORD+1);
  REAL LEN,SUM,M;   INTEGER F,I,J,N;
  LEN:=B-A;
  T(1):=(FCT(A)+FCT(B))/2;
  N:=1;
  FOR I:=1 STEP 1 UNTIL ORD DO
     BEGIN
       SUM:=0.0;   M:=LEN/(2*N);
       FOR J:=1 STEP 2 UNTIL 2*N-1 DO SUM:=SUM+FCT(A+J*M);
       T(I+1):=(SUM/N+T(I))/2;
       F:=1;
       FOR J:=I STEP -1 UNTIL 1 DO
          BEGIN  F:=4*F;
                 T(J):=T(J+1)+(T(J+1)-T(J))/(F-1)
          END;
       N:=2*N;
     END;
T(1)*LEN
END;
```

When an exact formulation for the integrand cannot be specified and all that is available is a set of ordinate values which correspond to a set of non-uniformly distributed abscissae, then a nonadaptive automatic integrator for the integration of arbitrarily spaced data must be used. For such problems, either of the two techniques of Section 2.6 can be used. If Hennion's method of di-parabolic interpolation is used, then the input requirements for a computer program are the integration limits, a number specifying the number of quadrature abscissae, and two arrays; one for the abscissae and one for the ordinate values. It is assumed that the quadrature abscissae are in ascending order. A computer program based on Hennion's algorithm is as follows:

```
REAL PROCEDURE AVINT(REAL VALUE XMIN,XMAX;  INTEGER VALUE NOP;
                     REAL ARRAY XA,YA(*)                              );
BEGIN
  REAL CA,CB,CC,A,B,C,TERM1,TERM2,TERM3,SUM,SYL,SYU;
  INTEGER JM,JS,JUL,IA,IB;
  SUM:=0.0;  SYL:=XMIN;  JUL:=NOP-1;  IB:=2;
  FOR IA:=1 STEP 1 UNTIL NOP DO WHILE XA(IA) < XMIN DO IB:=IB+1;
  FOR IA:=1 STEP 1 UNTIL NOP DO WHILE XMAX < XA(JUL) DO JUL:=JUL-1;
  JUL:=JUL-1;
  FOR JM:=IB STEP 1 UNTIL JUL DO
   BEGIN
    TERM1:=YA(JM-1)/((XA(JM-1)-XA(JM))*(XA(JM-1)-XA(JM+1)));
    TERM2:=YA(JM)/((XA(JM)-XA(JM-1))*(XA(JM)-XA(JM+1)));
    TERM3:=YA(JM+1)/((XA(JM+1)-XA(JM-1))*(XA(JM+1)-XA(JM)));
    A:=TERM1+TERM2+TERM3;
    B:=-(XA(JM)+XA(JM+1))*TERM1-(XA(JM-1)+XA(JM+1))*TERM2-
                                   (XA(JM-1)+XA(JM))*TERM3;
    C:=XA(JM)*XA(JM+1)*TERM1+XA(JM-1)*XA(JM+1)*TERM2+
                                   XA(JM)*XA(JM-1)*TERM3;
    IF JM ¬= 2  THEN
       BEGIN  CA:=(A+CA)/2;  CB:=(B+CB)/2;  CC:=(C+CC)/2;   END
    ELSE
       BEGIN   CA:=A;  CB:=B;  CC:=C;  END;
    SYU:=XA(JM);
    SUM:=SUM+CA*(SYU**3-SYL**3)/3+CB*( SYU**2-SYL**2)/2+
                                          CC*( SYU-SYL);
    CA:=A;  CB:=B;  CC:=C;   SYL:=SYU;
   END;
SUM+CA*(XMAX**3-SYL**3)/3+CB*(XMAX**2-SYL**2)/2+CC*(XMAX-SYL)
END;
```

Finally, for the integration of a particular class of integrals it may be the case that a special quadrature technique will be used. As an example, in the approximation of integrals with weight functions such as $w(x) = \sin(Tx)$ or $w(x) = \cos(Tx)$ Filon's method is a suitable quadrature technique. A computer program based on Filon's algorithm will require for input the integration limits, a value for T, a code indicating whether $w(x) = \sin(Tx)$ or $w(x) = \cos(Tx)$, and a procedure for the integrand or else a set of function values. A computer program for Filon's method using a set of function values is as follows:

```
REAL PROCEDURE FILON(REAL VALUE A,B,T; INTEGER VALUE KEY,M;
                          REAL ARRAY F(*)                       );
 BEGIN
REAL H,S,TH,AL,C,BE,GA,SUM,A1,SU,SU1;  INTEGER I;
  H:=(B-A)/(M-1);   TH:=T*H;
  S:=SIN(TH);   C:=COS(TH);
  AL:=1.0/TH+S*C/TH**2-2.0*S*S/TH**3;
  BE:=(2.0+2.0*C*C-4.0*S*C/TH)/TH**2;
  GA:=4.0*(S/TH-C)/TH**2;
  SUM:=0.0;   A1:=A+H;
  IF KEY =1 THEN
     BEGIN
       SU:=F(M)*SIN(B*T)-F(1)*SIN(A*T);
       SU1:= -(F(M)*COS(B*T)-F(1)*COS(A*T))/2.0;
       FOR I:=2 STEP 2 UNTIL M-1 DO
          BEGIN
            SUM:=SUM+F(I-1)*COS(A1*T);   A1:=A1+H;
            SU1:=SU1+F(I)*COS(A1*T);     A1:=A1+H;
          END
     END
  ELSE
     BEGIN
       SU:=F(M)*COS(B*T)-F(1)*COS(A*T);
       SU1:=-(F(M)*SIN(B*T)-F(1)*SIN(A*T))/2.0;
       FOR I:=2 STEP 2 UNTIL M-1 DO
          BEGIN
            SUM:=SUM+F(I-1)*SIN(A1*T);   A1:=A1+H;
            SU1:=SU1+F(I)*SIN(A1*T);     A1:=A1+H;
          END
     END;
H*(AL*SU+BE*SU1+GA*SUM)
END;
```

The computer programs given above are representative of those usually written for a digital computer. It is to be noted that these programs fail to incorporate all of the desirable qualities of an automatic integrator. Therefore, in selecting a program for numerical integration, the limitations of the program and its possible effects on the accuracy of the quadrature approximation should be taken into account.

CHAPTER VII

CONCLUSION

7.1 The problem of defining and identifying an optimal quadrature rule

In this study an optimal quadrature rule for numerical integration has been characterized as having the attributes of (i) maximum precision, (ii) least upper bound quadrature error, (iii) minimum norm quadrature error, (iv) minimum roundoff error, (v) applicability to a large class of integrals, and (vi) suitability for use in a computer algorithm. It is implicit in this definition that properties (i)-(iv) could possibly have an associated constraint such as being defined relative to a fixed set of quadrature abscissae, a fixed quadrature weight, etc. The meanings associated with attributes (i), (ii), (iii), and (iv) have been characterized in Chapters II, III, and IV. Attribute (v) is intended to mean being applicable for the integration of all functions of a particular function space as well as being applicable to improper integrals. For (vi), the suitability for use in a computer algorithm coincides with the extent to which an ideal computer algorithm as characterized in Section 6.2 can be constructed by using the specified quadrature rule.

In attempting to assess the merits of this characterization of an optimal quadrature rule and the extent to which known quadrature rules ascribe to this characterization the following observations are given. First it is noted that a quadrature rule can be constructed to satisfy a specific criterion of optimality. Therefore the attributes given above are considered to be valid criteria for an optimal quadrature formula. Secondly, the criteria specified are general enough so that almost all of the known and commonly used quadrature rules satisfy at least one of the attributes mentioned. Thirdly, the criteria, if satisfied, define a quadrature rule which has sufficient accuracy for a broad range of applications. Finally, it is noted that the criteria are restrictive enough to prevent most quadrature rules from being classified as an optimal quadrature rule. On these points, it is concluded that the above characterization of an optimal quadrature rule is acceptable.

In attempting to isolate known quadrature rules which best conform to the given characterization of an optimal quadrature rule, the following observations are noted. First it is known that Gauss-Chebyshev quadrature rules satisfy both (i) and (iv). Also, from Chapter IV, it is known that quadrature rules satisfying (iii) converge to Gaussian quadrature rules when functions analytic on the entire real line and variable quadrature abscissae are considered. As yet, no connection has been made between (ii) and Gaussian quadrature rules. Therefore the Gauss-Chebyshev rules are likely candidates for being called optimal quadrature rules. The only drawback of Gauss-Chebyshev quadrature rules and Gaussian rules in general is that they are not well-suited for use in adaptive iterative automatic quadrature schemes. On the other hand, the Newton-Cotes rules satisfy (i) and in a limiting sense also satisfy (iii). However they do not give minimum roundoff error. This shortcoming is offset by their suitability for use in adaptive iterative automatic quadrature schemes. Most of the other known quadrature rules fail to adhere to at least three of the attributes mentioned here.

On the basis of the above remarks, it is concluded that there is no known quadrature rule which can truly be classified as being optimal for the criteria set forth. One of the reasons as to why this is the case is that some of the attributes (e.g. maximum precision) introduce undesirable traits in the associated quadrature rule. In other instances the attributes are mutually exclusive as in (i) and (iii) for example. Therefore, unless a set of compatible characteristics which satisfy the four properties mentioned in the second paragraph of this section can be found, it is unlikely that a truly optimal quadrature rule can be formulated. At best, some quadrature rules are optimal only in some sense.

BIBLIOGRAPHY

[1] Abramowitz, M. and I. Stegun eds. (1964), Handbook of Mathematical Functions, National Bur. Stand., Washington, ch. 22.

[2] Anderson, D.G. (1965), Gaussian Quadrature Formulae for $\int_0^1 -\ell n(x)f(x)dx$, Math. Comp. Vol. 19, pp. 477-480.

[3] Apostol, T. (1957), Mathematical Analysis, Addison-Wesley Publishing Co., Inc., Reading, Mass., p. 306.

[4] Barnhill, R.E. (1968), Asymptotic Properties of Minimal Norm and Optimal Quadrature, Numer. Math., Vol. 12, pp. 384-393.

[5] Barnhill, R.E. et al. (1969), A New Type of Chebyshev Quadrature, Math. Comp., Vol. 23, pp. 437-441.

[6] Bell, E.T. (1940), Development of Mathematics, McGraw-Hill Book Co., Inc., New York, p. 21.

[7] Cajori, F. (1895), A History of Mathematics, MacMillan Company, New York, p. 26.

[8] Chawla, M.M. (1969), On Davis's Method For the Estimation of Gauss-Chebyshev Quadratures, Math. Comp., Vol. 24, pp. 261-269.

[9] Cheney, E.W. (1966), Introduction to Approximation Theory, McGraw-Hill Book Co., Inc., New York, p. 58.

[10] Cody, W.J., Jr. (1973), Static and Dynamic Numerical Characteristics of Floating-Point Arithmetic, IEEE Trans. Comp., Vol. C-19, pp. 598-607.

[11] Cranly, R. and T.N.L. Patterson (1971), On the Automatic Numerical Evaluation of Definite Integrals, Comp. J., Vol. 14, pp. 189-198.

[12] Davis, P.J. (1963), Interpolation and Approximation, Blaisdell Publishing Co., New York, p. 241.

[13] Davis, P.J. and P. Rabinowitz (1965), Ignoring the Singularity in Approximation Integration, SIAM J. Num. Anal., Vol. 2, pp. 367-382.

[14] Davis, P.J. and P. Rabinowitz (1967), Numerical Integration, Blaisdell Publishing Co., New York.

[15] Eckhardt, U. (1968), Einige Eigenschaften Wilfscher Quadraturformeln, Numer. Math., Vol. 12, pp. 1-7.

[16] El-Tom, M.E.A. (1971), On Ignoring the Singularity in Numerical Quadrature, SIAM J. Num. Anal., Vol. 8, pp. 412-424.

[17] Filon, L. (1928), On a Quadrature Formula for Trigonometric Integrals, Proc. Roy. Soc. Edinburgh, Vol. 49, pp. 38-47.

[18] Gautschi, W. (1968), Construction of Gauss-Christoffel Quadrature Formulas, Math. Comp., Vol. 22, pp. 251-270.

[19] Gautschi, W. (1970), On the Construction of Gaussian Quadrature Rules from Modified Moments, Math. Comp., Vol. 24, pp. 245-260.

[20] Gear, C.W. (1971), Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, N.J.

[21] Gill, P.E. and G.F. Miller (1972), An Algorithm for the Integration of Unequally Spaced Data, Comp. J., Vol. 15, pp. 81-84.

[22] Golub, G.H. and J.H. Welsch (1969), Calculation of Gauss Quadrature Rules, Math. Comp., Vol. 23, pp. 221-230.

[23] Hennion, P.E. (1962), Algorithm 77; Interpolation, Differentiation, Integration, C.A.C.M., Vol. 5, p. 96.

[24] Isaacson, E. and H.B. Keller (1966), Analysis of Numerical Methods, John Wiley and Sons, New York.

[25] Johnson, W.W. (1915), On Cotesian Numbers: Their History, Computation and Values to $n = 20$, Quart. J. Pure Appl. Math., Vol. 46, pp. 52-65.

[26] Kambo, N.S. (1970),Error of the Newton-Cotes and Gauss-Legendre Quadrature Formulas, Math. Comp., Vol. 24, pp. 261-269.

[27] Kaplan, E.J. (1952), Numerical Integration Near a Singularity, J. Math. And Physics, Vol. 31, pp. 1-28.

[28] Kopal, Z. (1961), Numerical Analysis, 2nd. ed., Chapman and Hall Ltd., London.

[29] Krylov, I.V. (1962), Approximate Calculation of Integrals, Trans. by A.H. Stroud, The MacMillan Co., New York.

[30] Lebed, G.K. (1968), Quadrature Formulas with Minimum Error for Certain Classes of Functions, Math. Notes of the U.S.S.R. Academy of Sciences, Vol. 3, pp. 368-373.

[31] McKeeman, W.M. (1962), Adaptive Numerical Integration by Simpson's Rule, C.A.C.M., Vol. 5, p. 604.

[32] Marasa, J.D. and D.W. Matula (1973), A Simulative Study of Correlated Error Propagation in Various Finite-Precision Arithmetics, IEEE Trans. Comp., Vol. C-19, pp. 587-597.

[33] Meir, A. and A. Sharma (1965), On the Romberg Quadrature Method, SIAM J. Num. Anal., Vol. 2, pp. 250-258.

[34] Gautschi W. (1968), Gaussian Quadrature Formulas, CACM, Vol. 11, p. 432-436.

[35] Miller, R.K. (1971), On Ignoring the Singularity in Numerical Quadrature, Math. Comp., Vol. 25, pp. 521-533.

[36] Montucla, J.F. (1968), Historie des Mathematiques, Vol. 4, A Blanchard, Paris.

[37] Oliver, J. (1968), On the Numerical Evaluation of Definite Integrals Using High-Order Integration Formula, Comp. J., Vol. 14, pp. 301-306.

[38] Rabinowitz, P. (1967), Gaussian Integration in the Presence of a Singularity, SIAM J. Num. Anal., Vol. 4, pp. 191-201.

[39] Ralston, A. (1965), A First Course in Numerical Analysis, McGraw-Hill Book. Co., New York.

[40] Reinsch, C.H. (1973), Eigenvalues of a Real, Symmetric, Tridiagonal Matrix, Algorithm 464, C.A.C.M., Vol. 16, p. 689.

[41] Richter, N. (1970), Properties of Minimal Integration Rules, SIAM J. Num. Anal., Vol. 7, pp. 67-79.

[42] Rothman, H.A. (1961), Gaussian Quadrature Rules with Weight Function x^n on the Interval (-1,1), Math. Comp., Vol 15, pp. 163-168.

[43] Rutishauser, H. (1962), Algorithm 125, Weightcoeff, C.A.C.M. Vol. 5, pp. 510-511.

[44] Rutishauser, H. (1963), On a Modification of the QD-Algorithm with Graeffetype Convergence, ZAMP, Vol. 13, pp. 493-496.

[45] Sack, R.A. (1962), A Fully Stable Rational Version of the QR Algorithm for Tridiagonal Matrices, Numer. Math., Vol. 18, pp. 432-441.

[46] Sack, R.A. and A.F. Donovan (1972), An Algorithm for Gaussian Quadrature Given Modified Moments, Numer. Math., Vol. 18, pp. 465-478.

[47] Secrest, D. (1965), Numerical Integration of Arbitrarily Spaced Data and Estimation of Errors, SIAM J. Numer. Anal., Vol. 2, pp. 52-68.

[48] Sites, R.L. (1972), Algol W Reference Manual, Stanford University.

[49] Stroud, A.H. and K. Chen (1972), Peano Error Estimates for Gauss-Laguerre Quadrature Formulae, SIAM J. Num. Anal., Vol. 9, pp. 333-340.

[50] Stroud, A.H. and D. Secrest (1966), <u>Gaussian Quadrature Formulae</u>, Prentice-Hall, Englewood Cliffs, N.J.

[51] Szego, G.(1959), <u>Orthogonal Polynomials</u>, 2nd. ed., American Math. Soc. Colloquium Publications, Vol. 23.

[52] Wilkinson, J.H. (1965), <u>The Algebraic Eigenvalue Problem</u>, Oxford, Clareden Press.

[53] Yoke, J.M. (1973), Roundings in Floating-Point Arithmetic, IEEE Trans. Comp., Vol. C-19, pp. 577-587.