NUMERICAL QUADRATURE: THEORY AND COMPUTATION

by
Lingyun Ye

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
September 2006

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**NUMERICAL QUADRATURE: THEORY AND COMPUTATION**" by **Lingyun Ye** in partial fulfillment of the requirements for the degree of **Master of Computer Science**.

Dated: August 17th 2006

Supervisor: _____
Jonathan M. Borwein

Readers: _____
Andrew Rau-Chaplin

_____
David H. Bailey

DALHOUSIE UNIVERSITY

Date: **August 17th 2006**

Author: **Lingyun Ye**

Title: **NUMERICAL QUADRATURE: THEORY AND COMPUTATION**

Department: **Computer Science**

Degree: **M.C.Sc.** Convocation: **October** Year: **2006**

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

_____

Signature of Author

Table of Contents

Abstract

The double exponential 'tanh-sinh' transformation has proved to be a very useful tool in numerical integration. It can often exhibit a remarkable convergence speed even for the integrands with end-point singularities. Under certain circumstances, it can be shown that the tanh-sinh method has a quadratic convergence ratio of $O\left(\exp\left(-CN/\ln N\right)\right)$, where N is the number of abscissas. The tanh-sinh scheme is also well-suited for parallel implementation. These merits make the tanh-sinh scheme the best quadrature rule when high precision results are desired. This thesis surveys the development of the double exponential transformation and its (parallel) implementations in one and two dimensions. We examine the application of the tanh-sinh methods in relation to many other scientific fields. We also provide a proof to the quadratic convergence property of the tanh-sinh method over the Hardy space. This proof can be generalized to analyze the performance of other transformations.

To my God and my family

Acknowledgements

I would like to thank my supervisor Jonathan Borwein, for introducing me to the area of numerical integration, for his inspiring guidance, encouragement and great patience. I would also like to thank David Bailey, who generously shares his code and provides expert advices on the implementation issues, and Andrew Rau-Chaplin, who spend his valuable time serving on my committee. I am indebted as well to many of my colleagues, especially Mason Macklem, Herre Wiersma, Isabel Gao, Scott Wilson and David Langstroth for their kind help and support. Finally, I want to express my thanks to my family and friends who silently support me in their own ways.

Chapter 1

Introduction

The theory of integration is a fundamental branch of mathematics. Although there are various notions of integral based on measure theory, we will limit our discussion to Riemann integration in this thesis.

Formally speaking, for a given function $f : B \subseteq \mathbb{R}^N \to \mathbb{R}$, we will consider the evaluation of the definite integral

$$I(f) = \int_B f(x)dx, \tag{1.1}$$

which is defined in terms of a Riemann sum.

In the evaluation of (1.1), the first tool that comes to our mind might be the Fundamental Theorem of Calculus, which plays a central role in calculus and integration theory. It states that for any continuous function f on a closed interval $[a, b]$,

$$\int_a^b f(x)dx = F(b) - F(a) \tag{1.2}$$

Here, $F(x)$ is an antiderivative of $f(x)$. If $F(x)$ is available and simple enough, (1.2) can be applied to get a satisfactory result. However, (1.2) doesn't always work well in practice for several reasons.

First, the indefinite integral is not always available since some $F(x)$ can't be represented as combinations of elementary functions. There exist numerous integrals which look very simple but can not be evaluated exactly [16], such as:

$$\int_{-\pi}^{0} \left(a^2 \sin^2 x + b^2 \cos^2 x\right)^{1/2} dx,$$

$$\int_1^2 \frac{dx}{\ln x},$$

and

$$\int_0^\pi \frac{\sin x}{x} dx.$$

The practical integration problems arising from various scientific areas and industry are often far more complicated. It is almost never possible to get their analytical expressions. Numerical integration is almost always the only choice.

Second, even if the antiderivative of an integrand can be found, the resulting analytical form can be too complicated to be easily evaluated. In such cases, it is often easier and faster to numerically approximate the value of an integral than to directly evaluate the antiderivative.

This encourages people to develop methods for approximating numerical values of certain definite integrals. More precisely, the one dimensional numerical integration problem can be stated as follows: Given a function $f(x)$ and an interval $[a, b]$, find an approximation $A(f)$, which is often called a *quadrature rule*, such that

$$\int_a^b f(x)dx \approx A(f),$$

where A is obtained without knowing any antiderivative information of f.

An n point approximation $A(f)$ takes the form of weighted sum:

$$A(f) = \sum_{i=1}^{n} w_i f(x_i). \tag{1.3}$$

Here, the x_i's are called *abscissas*, the sample points at which the function are evaluated. The w_i's are the *weights* associated with the abscissas.

The theory of numerical integration has a very long history which can be traced back to over 2000 years ago, when Archimedes approximated the area of circles using inscribed and circumscribed polygons. In the 10th century, an Arab mathematician Ibrahim introduced a more general method of numerical integration to calculate the area of the parabola. With the advent of the calculus in the 17th century, the theory of numerical integration developed rapidly. Many brilliant mathematicians made significant contributions to this area, including Simpson, Newton and Gauss. Over the past century, the invention of the electronic computer has made it possible to apply numerical integration techniques to the practical problems arising from engineering and applied sciences.

In this thesis, we examine a class of numerical integration techniques based on Euler-Maclaurin formula [9, p.309] and variable transformation. In particular, we focus on the discussion of the so-called 'tanh-sinh' method, which is observed to be

the best quadrature rules of this type. In Chapter 2, after a brief introduction to variable transformation based quadrature rules, we examine 3 important quadrature schemes in detail: the tanh, erf and tanh-sinh rules. We conclude that the tanh-sinh rule can often provide the best performance in extreme precision integration problems (approximate an integral to over hundreds of digits). Next, we prove our central result in theorem 3.4.1, which gives an error estimation of the tanh-sinh rule. Later in Chapter 4, we examine the parallel implementation of the tanh-sinh rule and the multidimensional tanh-sinh rules. We end with a brief survey of some applications of tanh-sinh rule in Section 4.3. The conclusion and future will follow in Chapter 5.

In the following sections of this chapter, we will introduce some of the famous quadrature rules, from very simple to more advanced schemes.

1.1 Interpolatory Quadrature

A very common class of integration techniques is called *interpolatory quadrature* [56, page 246], in which the integration of an *interpolant* of f is obtained as the approximation to the original integral. A function g is called an *interpolating function* [56, chapter 7] or *interpolant* of f if it satisfies:

$$g(x_i) = f(x_i) \quad i = 1, \ldots, n,$$

for given data

$$(x_i, f(x_i)) \quad i = 1, \ldots, n.$$

A general interpolatory quadrature scheme has 3 steps: First, the integrand f is sampled at some points, then the function which interpolates f at those points is obtained, as well as the corresponding weights w_i's. Finally, an approximation of the integral of the interpolant (which is relatively easy to compute) is calculated.

In practice, the interpolant is determined independently of the integrand f. The interpolation is used to compute the weights w_i according to the preassigned nodes x_i. The values of x_i and w_i will then be stored and used in the approximation of any integrand f over the same interval.

1.1.1 Newton-Cotes formula

The Newton-Cotes formula is a widely used interpolatory scheme, in which the interpolating function is a polynomial. In this scheme, n equally spaced points x_i are chosen in the interval $[a, b]$ as abscissas and the unique interpolating polynomial with degree $n-1$ that passes through these points is found. If we choose endpoints a, b as nodes, the formula is called the *closed* Newton-Cotes rule. Otherwise, we obtain the *open* Newton-Cotes formula.

In order to find the fitting polynomials, the Newton-Cotes formula uses *Lagrange interpolating polynomials* [56, section 7.2.2]:

$$f(x) \approx P(x) = \sum_{i=1}^{n} f(x_i) \prod_{k=1, i\neq k}^{n} \frac{x - x_k}{x_i - x_k}. \tag{1.4}$$

Integrating $P(x)$ over $[a, b]$ and choosing $x_i = a + (b-a)i/n$, we have the closed Newton-Cotes rule:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{k=0}^{n} B_{nk} f\left(a + kh\right) \tag{1.5}$$

where

$$B_{nk} = \frac{n}{b-a} \int_a^b \frac{w(x)dx}{\left(x - a - kh\right) w'\left(a + kh\right)} \tag{1.6}$$

and

$$w(x) = (x-a)(x-a-h)\cdots(x-a-nh).$$

When $n = 1$, (1.5) gives

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\left(f(a) + f(b)\right). \tag{1.7}$$

It is recognized as the simple trapezoidal rule which uses the linear function to approximate the original function.

The three points interpolation (when $n = 2$)

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \tag{1.8}$$

is called *Simpson's rule*. It is one of the most frequently used quadrature rules in practice.

The *degree* of a quadrature rule is used to measure the accuracy of a polynomial interpolatory quadrature.

Definition 1.1.1 [31, section 4.2.1] *The degree of accuracy of a quadrature rule is k, if this quadrature rule is exact for all polynomial of degree $d \leq k$ and is not exact for some polynomial of degree $d = k + 1$.*

It is obvious that the accurate degree of the Newton-Cotes rule is at least $n - 1$. In fact, an odd-order Newton-Cotes rule has one degree higher than this lower bound.

Theorem 1.1.1 [31, page 74] *A n-point Newton-Cotes formula has degree $2\lfloor \frac{n+1}{2} \rfloor - 1$.*

1.1.2 Gaussian rules

The Newton-Cotes quadrature rules require all the abscissas to be equally distributed along the interval. When high degree polynomial interpolation is used, this may cause some erratic behaviors. For example, the weights in Newton-Cotes rules can be negative which leads to the cancellation in the final summation. Also, when n abscissas are used in the approximation, the Newton-Cotes rules has the degree of accuracy of $n - 1$ or n. This is not the highest degree polynomial possible for n points. In order to get more accurate results, a higher degree approximation should be considered.

Gaussian quadrature can overcome some of the problems in the Newton-Cotes rules. This scheme was discovered by Gauss in 1820. It approximates the integral of $f(x)$ (see [1]) as:

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

where x_i are the roots of a degree-n Legendre polynomial $P_n(x)$ on $[-1, 1]$, and

$$w_i = \frac{-2}{(n+1)P_n'(x_i)P_{n+1}(x_i)}.$$

Here, the abscissas and weights are chosen in such a way that Gaussian quadrature is optimal in term of the degree of accuracy.

Theorem 1.1.2 [31, Thm 4.2.5] *For an n-point approximation, the degree of accuracy of the Gaussian quadrature rule is $2n-1$. This is the maximal degree of accuracy of any n-point interpolatory quadrature rule.*

The Legendre polynomial $P_n(x)$ is defined as

$$P_n(x) = \frac{1}{2\pi i} \oint (1 - 2tx + t^2)^{-1/2} t^{-n-1} dt.$$

A practical way to compute $P_n(x)$ is to use the recurrence formula

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x) \quad \text{for} \quad k = 2, \cdots$$

with $P_0(x) = 0$ and $P_1(x) = 1$. The derivative $P'(x)$ of the n-th Legendre polynomial can be calculated by:

$$P'_n(x) = n\frac{xP_n(x) - P_{n-1}(x)}{x^2 - 1}.$$

Theorem 1.1.3 [31, Thm 4.2.6] *The weights $w_1, w_2, \cdots w_n$ of the Gaussian quadrature rule are all positive:*

$$w_i > 0 \quad i = 1, \cdots, n, \quad n = 1, 2, \cdots .$$

Gaussian quadrature has been proven quite robust and fast for smooth and bounded functions on finite intervals. An implementation of the Gaussian quadrature rule is listed in Algorithm 1.

1.2 Composite Quadrature and Adaptive Rule

In the last section, we discussed several interpolatory quadrature rules, in which an interpolatory polynomial is integrated over the corresponding interval of integration. A quadrature rule with higher order normally gives a more accurate solution. In order to increase precision, an alternative is to decompose the interval $[a, b]$ into subintervals and apply low-order quadrature formulae to each subinterval. Summing up the partial results will give a more accurate approximation. For example, assume we partition the interval $[a, b]$ into n intervals $[x_i, x_{i+i}], i = 1 \ldots n$ by

$$a = x_0 < x_1 < \ldots < x_k = b.$$

Then the *composite trapezoid rule* is

$$Q(f) = \sum_{i=1}^{n} (x_i - x_{i-1}) \frac{f(x_i) + f(x_{i-1})}{2}. \tag{1.9}$$

Algorithm 1 Gaussian quadrature rule [9]

Initialize:
for $k = 1$ to m **do**
  Set $n = 3 \cdot 2^k$
  for $j = 1$ to $n/2$ **do**
    Set $r = \cos[\pi(j - \frac{1}{4})/(n + \frac{1}{2})]$;
    repeat
      Set $t_1 = 1$ and $t_2 = 0$
      for $j_1 = 1$ to n **do**
        Set $t_3 = t_2$ and $t_2 = t_1$
        Compute $t_1 = [(2j_1 - 1)rt_2 - (j_1 - 1)t_3]/j_1$
      end for
      Calculate $t_4 = n(rt_1 - t_2)/(r^2 - 1)$
      Set $t_5 = r$ and $r = r - t_1/t_4$
    until $r = t_5$ to within the desired working precision level
    Set $x_{j,k} = r$ and $w_{j,k} = 2/[(1 - r^2)t_4^2]$
  end for
end for
Quadrature:
for $k = 1$ to m (or until the desired accuracy is obtained) **do**
  Set $n = 3 \cdot 2^k$ and set $s = 0$
  for $j = 1$ to $n/2$ **do**
    Set $s = s + w_{j,k}[f(-x_{j,k}) + f(x_{j,k})]$
  end for
  Set $S_k = s$
end for

The idea of composite quadrature can be easily extended to *automatic integration*: keep increasing the number of subintervals, (e.g by halving the length of each subinterval) until the desired precision is obtained or certain stopping criteria are satisfied. In either case, the program should output the approximation value and theoretical error achieved.

A special type of automatic integration procedure is the *adaptive procedure*, in which the abscissas are chosen according to the nature of the integrand. For example, given a tolerance ϵ and a iteration upper bound N, one can apply a quadrature rule over the entire interval at very beginning. If the estimated error exceeds the tolerance ϵ, divide the interval into 2 subintervals and apply the quadrature rule in each. If the sum of the errors is still larger than ϵ, subdivide the interval with the largest error and compute the integral over the new subintervals. The procedure will continue iteratively until a satisfactory result is obtained or the number of iterations exceeds N.

A more detailed survey of adaptive integration quadratures can be found in [28].

1.3 Romberg Rule

As we have seen in the composite rule, the approximation value and error of an integration algorithm is often based on the step size h. An algorithm is said to be convergent for a function space E if $\lim_{h\to 0} Q(f) = I(f)$ for any $f \in E$. Thus, for a convergent algorithm we can increase the accuracy of the result by decreasing the step size h, and the exact value would be obtained by taking $h \to 0$. In practice we can't make the step size arbitrarily small due to the rounding error and the excessive computational cost. However, if the integrand can be expressed as a function of the step size h, we can then 'predict' the integration value at $h = 0$ from nonzero h using *Richardson extrapolation* [56, section 8.8].

According to the Euler-Maclaurin summation formula (which will be introduced in detail later in Theorem 2.1.1), if the integrand f is sufficiently smooth, then

$$T_n(f) - I(f) = C_2 h^2 + C_4 h^4 + C_6 h^6 + \cdots , \tag{1.10}$$

where C_k's are constants independent of h, $n = (b-a)/h$ and

$$T_n(f) = h\sum_{j=0}^{n} f(x_j) - \frac{h}{2}\left(f(a) + f(b)\right)$$

is the trapezoid rule for step size h. Similarly,

$$T_{2n}(f) - I(f) = \frac{C_2 h^2}{2^2} + \frac{C_4 h^4}{2^4} + \frac{C_6 h^6}{2^6} + \cdots . \tag{1.11}$$

Eliminating h^2 from (1.10) and (1.11) gives:

$$\frac{4T_{2n}(f) - T_n(f)}{3} - I(f) = \frac{2^{-2}-1}{3}C_4 h^4 + \frac{2^{-4}-1}{3}C_6 h^6 + \cdots , \tag{1.12}$$

from which we get the new formula

$$T_n^{(1)}(f) = \frac{4T_{2n}(f) - T_n(f)}{3}.$$

Note that in (1.12), the term with h^2 is eliminated. Therefore, the value $T_n^{(1)}(f)$ should be more accurate than $T_{2n}(f)$. This process can be continued recursively

$$T_n^{(i)}(f) = \frac{4^i T_{2n}^{(i)}(f) - T_n^{(i)}(f)}{4^i - 1}, \tag{1.13}$$

which gives us the following table:

$$\begin{array}{cccccc}
T_1^0 & & & & \\
T_2^0 & T_2^1 & & & \\
T_4^0 & T_4^1 & T_4^2 & & \\
T_8^0 & T_8^1 & T_8^2 & T_8^3 & \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

Thus the convergence of the trapezoidal rule can be accelerated by the repeated application of (1.13). A detailed discussion of Romberg integration can be found in Section 6.3 of [28].

1.4 Error Estimation

In numerical integration, we use a finite summation to approximate the value of an integral:

$$I(f) = \int_a^b f(x)dx = \sum_{i=1}^{n} w_i f(x_i) + E \tag{1.14}$$

A quadrature rule is of little use unless the error $|I(f) - A(f)|$ can be effectively measured. There are two type of errors in numerical integration: *truncation error* and *roundoff error*.

Roundoff error comes from the fact that we can only compute the summation to finite precision, due to the limited accuracy of a computer's representation of floating point numbers. In general, the roundoff error is negligible compared with truncation error. However, in the case where the number of abscissas in the summation of (1.14) is huge, or the tolerance ϵ is too small, rounding error might need to be taken into account. An extensive examination on how to detect and handle roundoff error in an integration rule can be found in [28]. The roundoff error will not be considered in our error estimation, since we focus on high precision integration: the evaluation of an integral to over 50 digits and often hundreds of digits. In order to carry out the high precision arithmetic, we will use the package ARPREC [2, 6] which supports extremely high precision numbers (up to approximately ten million digits).

The truncation error arises from the fact that the summation is only an approximation of the integral. The analytical estimation of truncation error often relies on the properties of the underlying integrands and the specific quadrature rule. Many tools, such as the theory of analytic functions, asymptotic analysis, and functional analysis, are used to estimate error bounds. A survey of such techniques can be found in Chapter 4 of [28]. On the other hand, the standard analysis of truncation error often involves information about the higher-order derivatives of the integrand. In practice, such information is usually not available and very expensive to obtain. The theoretical error bound is therefore only useful for the estimation of the efficiency of a quadrature rule, but is rarely used in the implementation.

In general, the practical error estimation can't give any guarantee of success. However, it works well in most cases, since many of the integration problems arising from real-world applications are well behaved.

One of the most commonly used error estimation techniques is to compare the results by a pair of integration formulas with different accuracies. For example, assume A_1 and A_2 are two approximations to the integral and A_1 is more accurate than A_2 so that

$$|A_1 - I| \ll |A_2 - I|$$

implies

$$|A_1 - I| < |A_1 - A_2| .$$

Then $|A_1 - A_2|$ might be used as a upper bound for $|A_1 - I|$. However, if A_1 turns out to be much better than A_2, then $|A_1 - A_2|$ is indeed an overestimation to $|A_1 - I|$. In this case, the program will sometimes quit with failure even if the desired estimation is obtained.

There are many ways to choose the pair of approximation rules. In practice, most algorithms will choose two belonging to the same family of rules. In this thesis, we will mainly focus on the discussion of variable transformation based integration algorithms, which utilize the trapezoidal rule. For error estimation, we will compare the results from the two consecutive levels of computation, and use the one with smaller step size as the more accurate estimation. (see Section 2.2 of the Chapter 2 for details).

1.5 Multidimensional Integration

In contrast with the one-dimensional integration problem, multidimensional quadrature is much more complicated. First, in one dimension we only need to deal with finite and (half) infinite intervals, while in $\mathbb{R}^N$ the domain of integration can be arbitrary. The integration formulae are often different with regard to different integration areas. Multiple integration rules over some standard regions (such as the hypercube, simplex and hypersphere) can be elementary. For more general regions, a useful technique is to convert them to one of the standard regions by a proper chosen non-singular transformation [28]. However, unlike in one dimension, two arbitrary compact regions in multidimensional space are not always equivalent [48].

Second, the interpolatory integration formulae, such as the Newton-Cotes and Gaussian quadratures, rely on the theories of function interpolation and orthogonal polynomials, which are far more complex in multidimensional space. In order to construct an N-dimensional n-point interpolatory rule with accurate degree d, one needs to solve a nonlinear system consisting of $\binom{N+d}{N}$ equations with $n(N+1)$ unknowns (see Section 5.2 of [31]). In general, the solution of such systems are often too complicated to be calculated directly.

1.5.1 Monte Carlo method

The Monte Carlo numerical integration approximates an integral using the statistical approach. An estimate of the integral is obtained by multiplying the volume of the integration region with the mean value of the integrand [31]:

$$I(f) = \int_B f(\mathbf{x})d\mathbf{x} = vol(B) \int_{\mathbb{R}^N} vol(B)^{-1} c_B(\mathbf{x}) f(\mathbf{x})d\mathbf{x} = vol(B) \cdot \mu(f).$$

where $\mu(f)$ is the mean value of f and c_B and $vol(B)$ are the characteristic function and volume of domain B. The quantity $vol(B)^{-1}c_B(\mathbf{x})$ can be interpreted as the probability density.

In practice, the mean value $\mu(f)$ is calculated as the average value of $f(x)$ at n randomly sampled points x_i with density $vol(B)^{-1}c_B(\mathbf{x})$:

$$A\mu f = \frac{1}{n} \sum_{i=1}^{n} f(x_i),$$

which will converge to $\mu(f)$ as $n \to \infty$.

By the Central Limit Theorem, the convergence rate of Monte Carlo integration is $O(1/\sqrt{n})$. This implies that for one more correct digit, the number of sampling points has to increase by a factor of 100. The Monte Carlo method suffers by its slow convergence rate so that it is not very appropriate for solutions that require high accuracy. However, it can be shown that the convergence ratio $O(\sqrt{n})$ is independent of dimension N [31]. This makes Monte Carlo integration one of the most commonly used integration scheme for high dimensional applications.

1.5.2 Lattice rule

The one-dimensional composite trapezoidal rule (1.9) can be generalized into multiple dimensions by distributing the abscissas equally within the integration area. An *integration lattice* L in $\mathbb{R}^N$ is a discrete subset of $\mathbb{R}^N$ such that [43]:

- L is closed under the plus and minus operations;
- L contains N linearly independent points;
- 0 is an isolated point of L and $\mathbb{Z}^N \subseteq L$.

Consider the integrating function f on the N-dimensional unit cube $[0,1)^N$, a *lattice rule* can be written as [31]

$$A(f) = \frac{1}{n_1 \cdots n_m} \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_m=0}^{n_m-1} f\left(\left\{\frac{i_1}{n_1}\mathbf{z}_1 + \cdots + \frac{i_m}{n_m}\mathbf{z}_m\right\}\right), \tag{1.15}$$

where $\mathbf{z}_i$ is a sequence of linearly independent integer vectors and the braces $\{\mathbf{x}\}$ denotes a vector whose components are the fractional part of vector $\mathbf{x}$. In (1.15), m is called the *rank* of $A(f)$ and $n_1, \cdots n_m$ are the *invariants* of $A(f)$.

The rank m can take any values between 1 and N. The lattice rule with $d = 1$ is called *the method of good lattice points* [29]

$$A(f) = \frac{1}{n} \sum_{i=0}^{n-1} f\left(\left\{\frac{i}{n}\mathbf{z}\right\}\right). \tag{1.16}$$

When $m = N$, we have the *product trapezoidal rule*

$$A(f) = \frac{1}{n^N} \sum_{i_1=0}^{n-1} \cdots \sum_{i_N=0}^{n-1} f\left(\left\{\left(\frac{i_1}{n}, \ldots, \frac{i_N}{n}\right)\right\}\right), \tag{1.17}$$

where n is the number of sampling points in one dimension. (1.17) is an expansion of (1.9) which applies the one-dimensional trapezoidal rule along each axis and takes the Cartesian product.

The z_i's in (1.15) should be chosen in a way that minimizes the error bound for a particular class of functions. For periodic functions, it is known that the lattice rule perform very efficiently. In this case, if we assume $f \in C^\alpha$, the convergence rate of the lattice rule can be shown to be $O(\rho^{-\alpha})$, where ρ is called the *Zaremba index*, a parameter indicating the efficiency of a lattice rule [43]. On the other hand, lattice rules can only achieve $O(\rho^{-1})$ convergence ratio for non-periodic functions. In practice, in order to accelerate the approximation speed, a non-periodic integrand will often be applied with a periodizing transformation to get a periodic function. See [26, 43, 31] for a more detailed discussion.

Chapter 2

High Precision Integration

In practice, the evaluation of integrals to 16 digits is sufficient for most applications. Very few applications need very high precision quadrature. However, in the past several years high precision integration has emerged as a very useful tool in experimental mathematics. In experimental mathematics, it is often necessary to calculate definite integrals to high precision. The obtained numerical values can be used to identify definite integrals in analytic terms which may lead to formal "paper" proof. Such experimental evaluation of integrals is often combined with *integer relation detection schemes* [25]: for n real variables $x_1 \ldots x_n$, find integers a_i, not all zero, such that

$$a_1 x_1 + a_2 x_2 + \ldots a_n x_n = 0.$$

Integer relation detection is commonly used to determine whether the numerical value is given by a formula of a certain type with unknown integer or rational coefficients. The current best integer relation algorithm is the *PSLQ* algorithm, in which very high precision (normally more than hundreds or thousand digits) is necessary in order to output meaningful results.

Example 2.0.1 [8, page 60] *Consider the evaluation of the following definite integral*

$$\int_0^1 \frac{t^2 \ln(t)}{(t^2-1)(t^4+1)} dt.$$

This is the output of Maple 9.0:

$$\frac{1}{4} \lim_{t \to 0^+} \operatorname{dilog}(1+t) + \ln(t)\ln(1+t) + \operatorname{dilog}(t) + \frac{1}{48}\pi^2$$
$$-2\frac{\sum_{\_R1=RootOf\left(\_Z^4+1\right)} -1/4 \left(\_R1^2-1\right)\left(\ln\left(\_R1\right)\ln\left(\frac{t-\_R1}{\_R1}\right) - \operatorname{dilog}\left(\frac{t}{\_R1}\right)\right)}{\_R1^{-3}}$$
$$+\frac{1}{2}\frac{\sum_{\_R1=RootOf\left(\_Z^4+1\right)} -1/4 \left(\_R1^2-1\right)\left(\ln\left(\_R1\right)\ln\left(-\frac{-1+\_R1}{\_R1}\right) - \operatorname{dilog}\left(\_R1^{-1}\right)\right)}{\_R1^{-3}}.$$

Instead of evaluating this limit analytically, we compute this integral to 100 decimals:

$$\int_0^1 \frac{t^2 \ln(t)}{(t^2-1)(t^4+1)} = 0.18067126259065494279230812898167161533711457101829676626624079429375856622413300177089825415048379 97.$$

We can then use an integer relation recognition scheme (PSLQ) to obtain a closed expression for the integral based only on the 100 digits above:

$$\int_0^1 \frac{t^2 \ln(t)}{(t^2-1)(t^4+1)} = \frac{\pi^2(2-\sqrt{2})}{32}.$$

In general, if we wish to find a relation of length n, with the coefficients a_i having a maximum size of d decimal digits, both the input data x_i and the working precision of the integer relation detection scheme must be at least nd digits. While this is just a theoretical lower bound which is independent of the integer relation algorithm used, the typical working precision for PSLQ would be $10\% - 15\%$ higher than this minimum value.

Except for the Gaussian quadrature, none of the algorithms mentioned in Chapter 1 are suitable for high precision approximation. Even Gaussian quadrature can only converge quickly to high accuracy for well-behaved functions. In the case where the integrand is not differentiable or not bounded around the end-points, Gaussian quadrature is slow [7]. Also, the initialization time in Gaussian quadrature will significantly grow as the desired precision increases. When the required precision exceeds 1000 digits, the initialization procedure normally overwhelms the overall quadrature cost, even if several integrals are performed using the same initialization data [7].

In practice, there are many ways to handle the singularities of an integrand. For example, the so-called "graded mesh" method places more sample points within the area close to the singular points to improve the accuracy. Another approach is to generalize Euler-Maclaurin expansion and construct new formulae based on the asymptotic properties of the original quadrature error.

This chapter presents a class of numerical integration methods based on variable transformation and the trapezoidal rule. We start by introducing the Euler-Maclaurin summation formula. Later in Section 2.2, the technique of variable transformation is discussed and three transformations (tanh, erf and tanh-sinh) are introduced. The

analysis of their performance and some convergent results are given in Section 2.3. It can be seen that these methods are very robust and efficient in dealing with integrands with singularities.

2.1 Euler-Maclaurin Summation Formula

The tanh, erf and tanh-sinh quadrature to be introduced in the following sections are all based on the Euler-Maclaurin summation formula, which can be regarded as an extension of the trapezoidal rule. It provides a very important theoretical support to the variable transformation related integration schemes.

Definition 2.1.1 [30, Def. 9.23] *The* Bernoulli polynomials B_n *of degree* n *are defined by* $B_0(x) = 1$ *and*

$$B_n' = B_{n-1},$$

with the constraints

$$\int_0^1 B_n(x)dx = 0, \qquad n = 0, 1, 2, \ldots$$

The rational numbers

$$b_n = n!B_n(0)$$

are called Bernoulli numbers.

Theorem 2.1.1 (Euler-Maclaurin Summation Formula) [9, p.309] *Let* $m \geq 0$ *and* $n \geq 1$ *be integers, and* $h = (b-a)/n$, $x_j = a + jh$ *for* $0 \leq j \leq n$. *Assume that* $f(x)$ *is at least* $(2m+2)$*-times continuously differentiable on* $[a, b]$. *Let* T_n *designate the trapezoidal rule over interval* $[a, b]$*:*

$$T_n(f) = h\left(\frac{1}{2}f(x_0) + f(x_1) + \ldots + f(x_{n-1}) + \frac{1}{2}f(x_n)\right).$$

Then

$$\int_a^b f(x)dx = T_n(f) - \sum_{i=1}^{m} \frac{h^{2i}B_{2i}}{2i!}\left(f^{(2i-1)}(b) - f^{(2i-1)}(a)\right) - E, \tag{2.1}$$

where B_{2i} *denote the Bernouli numbers, and*

$$E = \frac{h^{2m+2}(b-a)B_{2m+2}f^{2m+2}(\xi)}{(2m+2)!} \tag{2.2}$$

for some $\xi \in (a, b)$.

If the derivatives of $f(x)$ with the same order are equal at endpoints a and b, (2.1) becomes

$$\int_a^b f(x)dx = T_n(f) - E \tag{2.3}$$

The error of approximation is just E, which is $O\left(h^{2m+2}\right)$.

Theorem 2.1.2 [28, p.137] *Let $f(x) \in C^{2m+1}[a,b]$, $f^k(a) = f^k(b)$ for $k = 1 \ldots 2m-1$ and assume $|f^{2m+1}(x)| \leq M$. Then*

$$\left| \int_a^b f(x)dx - T_n(f) \right| \leq \frac{C}{n^{2m+1}}, \tag{2.4}$$

where C is a constant independent of n and may be taken as

$$C = M(b-a)^{2m+2} 2^{-2m} \pi^{-2m-1} \zeta(2m+1)$$

and $\zeta(k) = \sum_{j=1}^{\infty} j^{-k}$ is the Riemann zeta function.

In the case of the evaluation of an improper integral, (2.4) still holds if $f^{(n)}$ vanish as x goes to infinity for $n = 0 \ldots 2m-1$.

For 2π-periodic functions $f : \mathbb{R} \to \mathbb{R}$ with high degree of smoothness, the conditions of Theorem 2.1.2 are satisfied.

Corollary 2.1.1 [28, p.137] *Assume $f(x) \in C^{2m+1}(-\infty, \infty)$ and is 2π-periodic. If $|f^{2m+1}(x)| \leq M$, then*

$$\left| \int_0^{2\pi} f(x)dx - T_n(f) \right| \leq 4\pi M \zeta(2m+1)/n^{2m+1}.$$

2.2 Variable Transformation

Theorem 2.1.2 throws some light on a class of numerical integration techniques: if the integrand f satisfies certain conditions, a simple trapezoidal rule is often superior to other quadrature rules. It has been well known [28, 42, 44] that the application of the trapezoidal rule to analytic functions with equal step size can achieve very high accuracy. In 1970, Mori proved the optimality of the trapezoidal rule for the integrals of analytic functions [53].

In general, the integrand f does not have the desired property that every order derivative of f is equal at both end points. In order to best utilize the high degree of accuracy implied by (2.4), Iri, Moriguti and Takasawa proposed the *IMT rule* [27].

Consider the integration of analytic functions f over $(0,1)$

$$I = \int_0^1 f(x)dx,$$

where f may have integrable singularities at the end points 0 or 1. IMT employs a change of variable

$$x = \frac{1}{Q}\int_0^t \exp\left(-\frac{1}{s} - \frac{1}{1-s}\right) ds \tag{2.5}$$

where

$$Q = \int_0^1 \exp\left(-\frac{1}{s} - \frac{1}{1-s}\right) ds$$

The basic idea is to transform the original integrand f in such a way that all the derivatives of the new function g vanish at both end points of the interval. After applying a careful chosen mapping function, the new integrand can become a bell-shaped function which does not have any singularity or blow up at the endpoints. The error of this formula is $O\left(\exp\left(-C\sqrt{N}\right)\right)$ [27].

Note that (2.5) is a one-to-one transformation which maps the original (finite) interval of integration onto itself. The singular points will stay in the finite complex plane, which makes the error analysis very complicated. On the other hand, Schwartz [42] proposed a general method of variable transformation mapping a finite interval of integration to $(-\infty, \infty)$. In this case, the singularities are moved to infinity so that

$$I = \int_a^b f(x)dx = \int_{-\infty}^{\infty} f(\psi(t))\psi'(t)dt. \tag{2.6}$$

Applying the indefinite trapezoid rule to (2.6), we get

$$T_h = h\sum_{-\infty}^{\infty} w_j f(x_j), \tag{2.7}$$

where $x_j = \psi(hj)$ and $w_j = \psi'(hj)$. The function $\psi(x)$ is required to be an absolutely continuous monotonic increasing function mapping $(-\infty, \infty)$ onto (a, b), with the property that all its derivatives converge to zero quickly for large x. An approximation can then be obtained by truncating the infinite summation (2.7) at some number N:

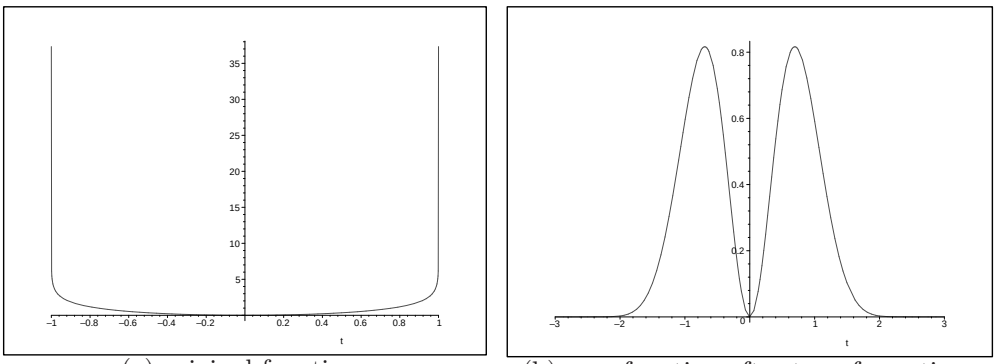
$$T_{h,N} = h\sum_{-N}^{N} w_j f(x_j). \tag{2.8}$$

In this case, the Theorem 2.1.2 guarantees a very fast convergence rate of order $O\left(h^{2m+2}\right)$.

As an example, Figure 2.1(a) shows the graph of $f(t) = -\ln\cos(\pi t/2)$, which has singular points at both end points $t = \pm 1$. This makes the integration very slow. Figure 2.1(b) shows the picture of the new function after applying the transformation

$$f\left(\psi(t)\right)\psi'(t) = -\ln\left(\cos\left(\frac{\pi\tanh\left(\pi/2\cdot\sinh(t)\right)}{2}\right)\right)\cdot\frac{\pi/2\cdot\sinh(t)}{\cosh^2\left(\pi/2\cdot\sinh(t)\right)}.$$

The new integrand becomes a smooth bell-shaped function without singular point.



(a) original function (b) new function after transformation

Figure 2.1: Comparison of the original function and transformed function

In the rest of this section, we will briefly survey the three important transformations: the tanh, erf and tanh-sinh rules.

2.2.1 Tanh quadrature rule

The tanh rule was first introduced by C. Schwartz in [42], in which he suggested using a change of variables and the trapezoid rule to accelerate the convergence of integral approximation over finite intervals. One particular transformation proposed was

$$\psi(t) = \tanh\left(\frac{t}{2}\right), \tag{2.9}$$

and the step size h he suggested was:

$$h = \pi\sqrt{\frac{2}{N}}. \tag{2.10}$$

Combining (2.9) and (2.10) together gives the *tanh rule*.

In [23], Haber studied the behavior of the tanh rule in the context of Hardy spaces and showed that its convergence rate is $O(e^{-C\sqrt{N}})$, where N is the number of evaluation points and C is a constant independent of N.

Theorem 2.2.1 [23, Thm 2] *Consider the application of the tanh rule over the Hardy space H^2. This space consists of the analytic functions f in the open disc whose Taylor coefficients satisfy*

$$\sum_{n=0}^{\infty} |a_n|^2 < \infty.$$

Let h and N denote the step size and the number of evaluation points used. If $h \to 0$ and $N \to \infty$ simultaneously in such a manner that $Nh \to \infty$, then the error norm can be estimated as

$$\|E_{h,N}\|^2 = 4\pi^2 e^{-2\pi^2/h} + 8\ln(2)e^{-Nh} + O\left(e^{-4\pi^2/h}\right) + O\left(he^{-2Nh}\right). \tag{2.11}$$

The error norm $\|E_{h,N}\|$ is minimized when h is chosen as $h = \pi\sqrt{2/N} + O(1/N)$.

In Algorithm 2, we give an implementation of the 1-D tanh rule. The **Initialize** subroutine is used to generate abscissa-weight pairs x_k and w_k. The **Quadrature** subroutine is used to evaluate the function $f(x_j)$ at each point and compute the summation $I(f)$.

The three major parts of the computational cost in this algorithm are the calculation of x_j and w_j , the function evaluation and the summation. It should be noted that the calculation of x_j and w_j is independent of the function f. For each test function, it is not necessary to recompute x_j and w_j each time. In order to reduce cost, x_j and w_j will be precalculated and stored at the very beginning of the program.

The parameter $h = 2^{-k}$ is the distance between two adjacent points, with k called the *level* of the quadrature. When the level increases 1, h will be reduced by half. This doubles the number of points N as well as the computational cost. In each implementation, the computation of the quadrature will normally start at a low level and keep increasing until the desired accuracy is achieved.

2.2.2 Error function quadrature

Error function quadrature uses the transformation

$$x = g(t) = \mathrm{erf}(t) = \frac{2}{\pi}\int_0^t e^{-x^2}dx.$$

The abscissas x_k and weights w_k are defined as

$$x_j = \mathrm{erf}(jh) \quad \text{and} \quad w_j = \frac{2}{\pi}e^{-(jh)^2}.$$

Algorithm 2 tanh quadrature rule [9]

Input: the level m, function f

Output: the approximation hS

Initialize:
Set $h := 2^{-m}$
for $k := 0$ to $20 \cdot 2^m$ **do**
  Set $t := kh$
  Set $x_k := \tanh(t/2)$
  Set $w_k := 1/2\cosh^2(t/2)$
  if $|x_k - 1| < \epsilon$ **then**
    exit **do**
  end if
end for
Set $n_t = k$ (the value of k at exit).

Quadrature:
Set $S := 0$ and $h := 1$
for $k := 1$ to m (or until the desired accuracy is obtained) **do**
  $h := h/2$.
  for $i := 0$ to n_t step 2^{m-k} **do**
    if mod $(i, 2^{m-k+1}) \neq 0$ or $k = 1$ **then**
      if $i = 0$ **then**
        $S := S + w_0 f(0)$
      else
        $S := S + w_i(f(-x_i) + f(x_i))$
      end if
    end if
  end for
end for
Result = hS

In the computation of x_j, we use the identity

$$\operatorname{erfc}(t) = 1 - \operatorname{erf}(t)$$

for the error function evaluation. Erfc is the *complementary error function*, which can be computed to very high precision using the formula [13]:

$$\operatorname{erfc}(t) = \frac{e^{-t^2}\alpha t}{\pi}\left(\frac{1}{t^2} + 2\sum_{k\geq 1}\frac{e^{-k^2\alpha^2}}{k^2\alpha^2 + t^2}\right) + \frac{2}{1 - e^{2\pi t/\alpha}} + E$$

where $t > 0$ and $|E| < 3e^{-\pi^2/\alpha^2}$. The parameter α is a small user-provided value used to control the error E.

Algorithm 3 Error function Evaluation [9]

Initialize:
Set $\alpha = \pi/\sqrt{n_p \log(10)}$
Set $n_t = n_p \log(10)/\pi$
Set $t_2 = e^{-\alpha^2}$, $t_3 = t_2^2$ and $t_4 = 1$
for $k = 1$ to n_t **do**
    Set $t_4 = t_2 \cdot t_4$, $E_k = t_4$, $t_2 = t_2 \cdot t_3$
end for
Evaluation of function at x
Set $t_1 = 0$, $t_2 = x^2$, $t_3 = e^{-t_2}$ and $t_4 = \epsilon/(1000 \cdot t_3)$
for $k = 1$ to n_t **do**
    Set $t_5 = E_k/(k^2\alpha^2 + t_2)$ and $t_1 = t_1 + t_5$
    if $|t_5| < t_4$ **then**
        exit **do**
    end if
end for
Set $\operatorname{erf}(x) = t_3\alpha x/\pi \cdot (1/t_2 + 2t_1) + 2/(1 - e^{2\pi x/\alpha})$

2.2.3 Tanh-Sinh quadrature rule

The tanh-sinh rule was first proposed in [55] by Mori and Takahasi. The particular transformation they suggested was

$$\psi(t) = \tanh\left(\frac{\pi}{2}\sinh(t)\right),$$

Algorithm 4 Error function quadrature [9]

Initialize:
Set $h = 2^{2-m}$
for $k = 0$ to $20 \cdot 2^m$ **do**
  Set $t = kh$, $x_k = 1 - \mathrm{erfc}(t)$
  Set $w_k = 2/\sqrt{\pi} \cdot e^{-t^2}$
  if $|x_k - 1| < \epsilon$ **then**
    exit **do**;
  end if
end for
Set $n_t = k$ (the value of k at exit)
Quadrature:
Set $S = 0$ and $h = 4$
for $k = 1$ to m (or until the desired accuracy is obtained) **do**
  $h = h/2$
  for $i = 0$ to n_t step 2^{m-k} **do**
    if mod $(i, 2^{m-k+1} \neq 0)$ or $k = 1$ **then**
      if $i = 0$ **then**
        $S = S + w_0 f(0)$
      else
        $S = S + w_i(f(-x_i) + f(x_i))$
      end if
    end if
  end for
end for
Result = hS

where

$$\sinh t = (e^t - e^{-t})/2$$

and

$$\cosh t = (e^t + e^{-t})/2.$$

By differentiating the $\psi(t)$, we can get abscissas x_k and weights w_k

$$\begin{aligned} x_j &= \tanh\left(\pi/2 \cdot \sinh(jh)\right) \\ w_j &= \frac{\pi/2 \cdot \cosh(jh)}{\cosh^2\left(\pi/2 \cdot \sinh(jh)\right)}. \end{aligned}$$

Due to the existence of the exponential components $\cosh^2\left(\pi/2 \cdot \sinh(jh)\right)$ in the denominator of w_j, the tanh-sinh rule is often called a *double exponential transformation* in the sense that the derivative of the tanh-sinh transformation exhibits a double exponential vanishing rate at infinity. For most functions, the new integrand $f\left(\psi(t)\right)\psi'(t)$ often decays double exponentially as well. This makes the tanh-sinh scheme very efficient in high precision integration even if $f(x)$ has singularities or vertical derivatives. An implementation of the tanh-sinh rule is given in Algorithm 5. For integrals over more general integration areas such as half-infinite and infinite intervals, the tanh-sinh rule can be slightly revised, while the derivatives of the new transformations still exhibit a double exponential decay rate [36], such as:

$$x = \exp\left(\frac{\pi}{2}\sinh t\right) \quad \text{for} \quad \int_0^\infty f(x)dx$$

and

$$x = \sinh\left(\frac{\pi}{2}\sinh t\right) \quad \text{for} \quad \int_{-\infty}^\infty f(x)dx.$$

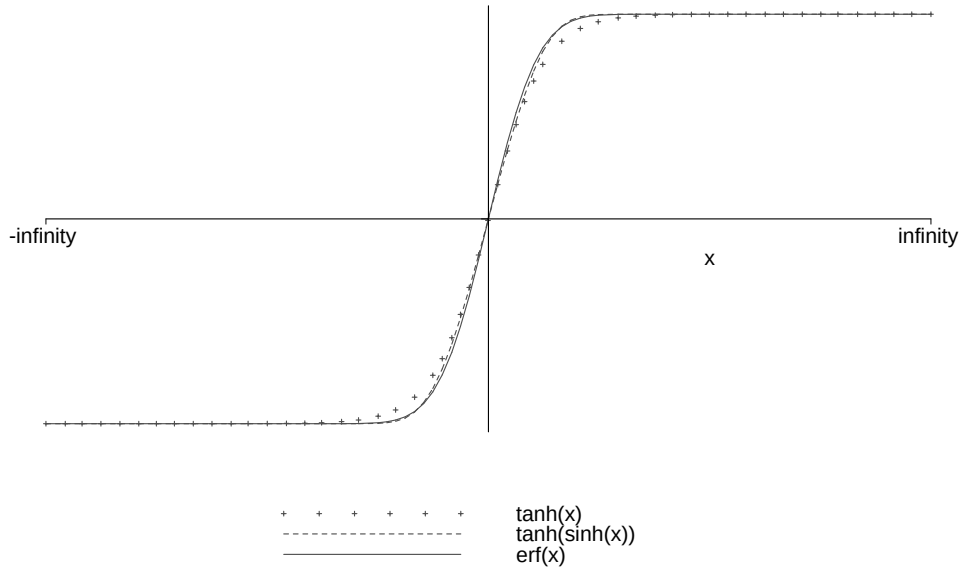
Algorithm 5 tanh-sinh quadrature rule [9]

Input: the level m, function f

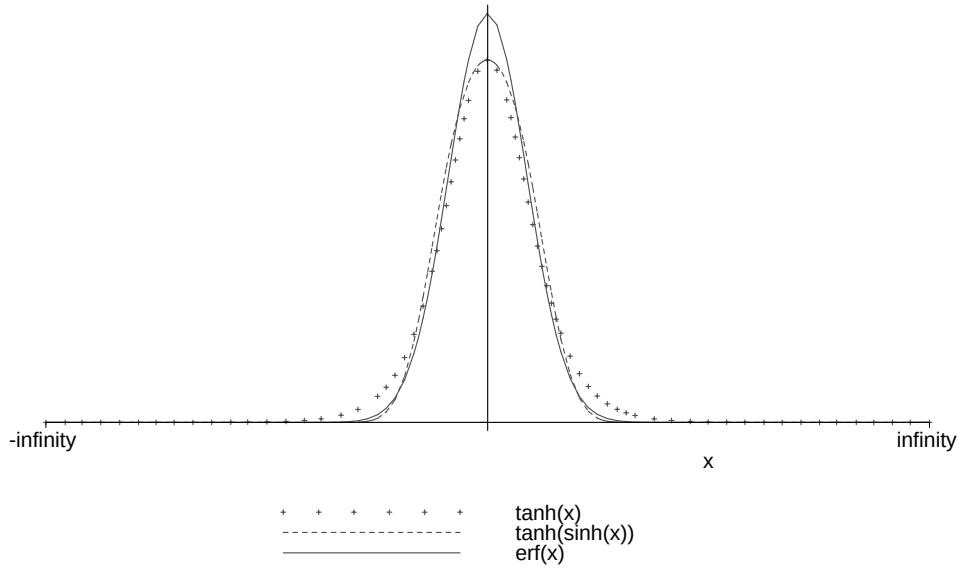
Output: the approximation hS

Initialize:
Set $h := 2^{-m}$
for $k := 0$ to $20 \cdot 2^m$ **do**
  Set $t := kh$
  Set $x_k := \tanh(\pi/2 \cdot \sinh t)$
  Set $w_k := \pi/2 \cdot \cosh t / \cosh^2(\pi/2 \cdot \sinh t)$
  if $|x_k - 1| < \epsilon$ **then**
    exit **do**
  end if
end for
Set $n_t = k$ (the value of k at exit).

Quadrature:
Set $S := 0$ and $h := 1$
for $k := 1$ to m (or until the desired accuracy is obtained) **do**
  $h := h/2$.
  for $i := 0$ to n_t step 2^{m-k} **do**
    if mod $(i, 2^{m-k+1}) \neq 0$ or $k = 1$ **then**
      if $i = 0$ **then**
        $S := S + w_0 f(0)$
      else
        $S := S + w_i(f(-x_i) + f(x_i))$
      end if
    end if
  end for
end for
Result = hS



(a) The pictures of $\tanh(x), \mathrm{erf(x)}, \tanh(\sinh(x))$



(b) The pictures of the derivatives of $\tanh(x), \mathrm{erf(x)}, \tanh(\sinh(x))$

Figure 2.2: $\tanh(x), \mathrm{erf(x)}, \tanh(\sinh(x))$ and their derivatives

2.3 Performance Comparison and Error Estimation

In [4] and [9] the authors describe the remarkable effectiveness of the doubly exponential tanh-sinh transformation for numerical integration—even for quite unruly integrands. Our experience is that the tanh-sinh is almost always as-or-more effective for high precision integration problems (see [3, 4]). In practice, the tanh-sinh scheme is almost invariably the best rule and is often the only effective rule when more that

50 or 100 digits are required. Actually, the tanh-sinh rule often exhibits quadratic convergent speed for a large class of integrands: when the level increases by one, the number of correct digits will often approximately double. Figure 2.2 shows the three schemes we have introduced: erf and tanh(sinh) are visually very close while tanh is the outlier.

As we have seen, the performance of different Euler-Maclaurin-based algorithms largely depends on the different choices of the transformations $\psi(t)$. These transformations can be characterized by the decay rate of their derivatives at infinity [49, 55]:

Decay Rate	**Transformation**
single exponential	tanh quadrature: $\tanh\left(\frac{t}{2}\right)$
super exponential	erf quadrature: $\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t \exp(-s^2) ds$
double exponential	tanh-sinh quadrature: $\tanh\left(\frac{\pi}{2}\sinh(t)\right)$
super double exponential	$\tanh\left(\frac{\pi}{2}\sinh(t^3)\right)$

In [49], Sugihara studied the convergence properties of various transformations by introducing a number of function spaces. Each one of the spaces consists of functions which are analytic in a strip containing the real line and have a unique decay rate. The intention here is to use the functions in those spaces to represent the new integrands after different transformations.

Definition 2.3.1 ([49], p.381) *For $d > 0$, define $\mathcal{D}_d$ to be the strip of length $2d$ containing the real axis:*

$$\mathcal{D}_d = \{z \in \mathbb{C} \,|\, |\mathrm{Im} z| < d\},$$

and let $B(\mathcal{D}_d)$ be the set of functions f analytic in $\mathcal{D}_d$ such that

$$\lim_{y \to d-0} \int_{-\infty}^{\infty} |f(x+iy)| + |f(x-iy)| \, dx < \infty$$

and

$$\int_{-d}^{d} |f(x+iy)| dy \to 0 \ \ as \ \ x \to \pm\infty.$$

Furthermore, let $\omega(z)$ be a non-zero function defined on the strip $\mathcal{D}_d$ that satisfies the conditions:

1. $\omega(z) \in B(\mathcal{D}_d)$.

2. $\omega(z)$ takes real values on the real line.

We then define the functional space H^∞ by:

$$H^\infty(\mathcal{D}_d,\omega)=\{f:\to\mathbb{C}\mid f(z)\ \text{ is analytic in } \mathcal{D}_d,\ \text{ and } \|f\|_{H^\infty}<\infty\},\qquad(2.12)$$

where the norm is defined as

$$\|f\|_{H^\infty}=\sup_{z\in\mathcal{D}_d}|f(z)/\omega(z)|.$$

By the definition of the norm, we have

$$|f(z)|\le\|f\||\omega(z)|.$$

This means $f(z)$ and $\omega(z)$ have the same decay rate. The function $\omega(z)$ can be used to uniquely identify the spaces with different decay rates.

The following theorem by Sugihara [49] gives both the upper bounds for the error norm of the trapezoidal rule and lower bounds for the optimal quadrature formula in $H^\infty(\mathcal{D}_d,\omega)$, where ω is either single exponentially or double exponentially convergent to zero at infinity.

Theorem 2.3.1 ([49], Thm 3.1 and 3.2) *Let the $\varepsilon^T_{N,h}\left(H^\infty(\mathcal{D}_d,\omega)\right)$ denote the error norm of the trapezoidal rule on the space $H^\infty(\mathcal{D}_d,\omega)$:*

$$\varepsilon^T_{N,h}\left(H^\infty(\mathcal{D}_d,\omega)\right)=\sup_{\|f\|\le1}\left|\int_{-\infty}^{\infty}f(x)dx-h\sum_{j=-n}^{n}f(jh)\right|.$$

Let $\varepsilon^{\min}_N\left(H^\infty(\mathcal{D}_d,\omega)\right)$ denote the optimal N-points quadrature formula:

$$\varepsilon^{\min}_N\left(H^\infty(\mathcal{D}_d,\omega)\right)=\inf_{1\le l\le N,a_j,c_{jk}}\left\{\sup_{\|f\|\le1}\left|\int_{-\infty}^{\infty}f(x)dx-h\sum_{j=1}^{l}\sum_{k=0}^{m_j-1}c_{jk}f^{(k)}(a_j)\right|\right\}.$$

where $a_j\in\mathcal{D}_d, c_{jk}\in\mathbb{C}$ and $N=m_1+m_2+\cdots+m_l$.

1. If the decay rate of $\omega(z)$ is specified by

$$\alpha_1\exp\left(-(\beta|x|)^\rho\right)\le|\omega(x)|\le\alpha_2\exp\left(-(\beta|x|)^\rho\right)\qquad(2.13)$$

where $\alpha_1,\alpha_2,\beta>0$ and $\rho\ge1$, then the error norm of the trapezoidal rule can be estimated as

$$\varepsilon^T_{N,h}\left(H^\infty(\mathcal{D}_d,\omega)\right)\le C_{d,\omega}\exp\left(-(\pi d\beta N)^{\frac{\rho}{\rho+1}}\right)\qquad(2.14)$$

where $C_{d,\omega}$ is a constant depending on d and ω, $N = 2n+1$, and the step size h is chosen as

$$h = (2\pi d)^{\frac{1}{\rho+1}} (\beta n)^{-\frac{\rho}{\rho+1}}$$

Also, we have

$$\varepsilon_N^{\min}\left(H^\infty(\mathcal{D}_d,\omega)\right) \geq C'_{d,\omega} N^{\frac{1}{\rho+1}} \exp\left(-\left(\left(\frac{2}{\rho+1}\right)^{\frac{1}{\rho}} 2\pi d\beta N\right)^{\frac{\rho}{\rho+1}}\right), \tag{2.15}$$

where $C'_{d,\omega}$ is another constant depending on d and w.

2. *If the decay rate of $\omega(z)$ is specified by*

$$\alpha_1 \exp\left(-\beta_1 \exp\left(\gamma|x|\right)\right) \leq |\omega(x)| \leq \alpha_2 \exp\left(-\beta_2 \exp\left(\gamma|x|\right)\right) \tag{2.16}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma > 0$, then the error norm of the trapezoidal rule can be estimated as:

$$\varepsilon_{N,h}^{T}\left(H^\infty(\mathcal{D}_d,\omega)\right) \leq C_{d,\omega} \exp\left(-\frac{\pi d\gamma N}{\ln\left(\pi d\gamma N/\beta_2\right)}\right) \tag{2.17}$$

where $C_{d,\omega}$ is a constant depending on d and ω, $N = 2n+1$, and the step size h is chosen as

$$h = \frac{\ln\left(2\pi d\gamma n/\beta_2\right)}{\gamma n}.$$

Also, we have

$$\varepsilon_N^{\min}\left(H^\infty(\mathcal{D}_d,\omega)\right) \geq C'_{d,\omega} \ln N \exp\left(-\frac{2\pi d\gamma N}{\ln\left(\pi d\gamma N/\beta_1\right)}\right), \tag{2.18}$$

where $C'_{d,\omega}$ is another constant depending on d and w.

If we compare the error bounds given above, it is clear that the upper bound of the trapezoidal rule is approximately equal to the lower bound of the optimal quadrature rule. The theorem actually implies that the trapezoidal rule is almost optimal among all possible quadrature rules over the space of $H^\infty(\mathcal{D}_d,\omega)$ for single and double exponential decay ω.

Now consider the tanh rule $\tanh(x/2)$, where

$$\omega = \frac{d}{dz}\tanh\left(\frac{z}{2}\right) = \frac{1}{2\cosh^2\left(z/2\right)}.$$

The conditions of Theorem 2.3.1 holds with $\beta = \rho = 1$ in (2.13). By (2.14), the error of the tanh rule can be estimated as

$$\varepsilon_{N,h}^{T}\left(H^{\infty}(\mathcal{D}_d, \omega)\right) = O\left(\exp\left(-\sqrt{\pi d N}\right)\right). \tag{2.19}$$

Furthermore setting $d = \pi/2$, (2.19) gives

$$\varepsilon_{N,h}^{T}\left(H^{\infty}(\mathcal{D}_{\pi/2}, \omega)\right) = O\left(\exp\left(-\pi\sqrt{N/2}\right)\right),$$

and the optimal step size should be chosen as $h = \pi\sqrt{2/N}$. This coincides with the result by Haber (see Theorem 2.2.1 and [23]).

For the error function

$$\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t \exp(-s^2) ds,$$

the conditions of Theorem 2.3.1 hold with $\beta = \rho = 2$. Then (2.14) yields

$$\varepsilon_{N,h}^{T}\left(H^{\infty}(\mathcal{D}_{\pi/2}, \omega)\right) = O\left(\exp\left(-\pi d N\right)^{2/3}\right)$$

where the step size $h = (2\pi d)^{1/2} n^{-2/3}$.

For the tanh-sinh rule, we have

$$\omega = \frac{d}{dz} \tanh\left(\frac{\pi}{2}\sinh(z)\right) = \frac{\pi/2 \cdot \cosh(z)}{\cosh^2\left(\pi/2 \cdot \sinh(z)\right)},$$

which satisfies the condition of Theorem 2.3.1 with $\beta_1 = \frac{\pi}{2}$ and $\beta_2 = \frac{\pi}{2} - \epsilon$, with ϵ a small positive number. The error can be estimated as

$$\varepsilon_{N,h}^{T}\left(H^{\infty}(\mathcal{D}_{\pi/2}, \omega)\right) = O\left(\exp\left(-\frac{\pi d N}{\ln\left(2\pi d N/(\pi - \epsilon)\right)}\right)\right),$$

where $h = \ln\left(4\pi d n/(\pi - \epsilon)\right)/n$.

Chapter 3

A New Approach to the Convergence Properties of the Tanh-Sinh Quadrature

In this chapter, we will provide an alternative proof of the convergence property of the tanh-sinh scheme. The analysis rests on the corresponding but somewhat easier analysis by Haber [23] of the less numerically effective 'tanh' quadrature.

As we have seen in Chapter 2, the tanh-sinh quadrature scheme is based on the frequent observation, rooted in the *Euler-Maclaurin summation formula* [3], that for certain bell-shaped integrands a simple block-function approximation to the integral is much more accurate than one would normally expect. In particular, the *tanh-sinh rule* uses the doubly-exponential transformation

$$\psi(x) = \tanh\left(\frac{\pi}{2}\sinh(x)\right) \tag{3.1}$$

with

$$\psi'(x) = \frac{\pi\cosh(x)}{2\cosh^2\left(\frac{\pi}{2}\sinh(x)\right)}.$$

Correspondingly, $\psi(x) = \tanh(x)$ gives rise to the scheme analyzed by Haber in [23].

A similar analysis may be undertaken for the erf rule, but not as explicitly since the zeros of the error function will be estimated numerically. In addition, various of the summations required appear to be more delicate.

3.1 Hardy Space

We will perform our analysis of the convergence of the tanh-sinh rule in the Hardy space H^2 (see [15]).

Definition 3.1.1 ([15], p.2) *For $0 < p < \infty$, the* Hardy space H^p *consists of the functions f, which are analytic on the unit disk and satisfies the growth condition*

$$\|f\|_{H^p} = \sup_{0\le r<1}\left\{\frac{1}{2\pi}\int_0^{2\pi}|f(re^{i\theta})|^p d\theta\right\}^{1/p} < \infty$$

where the quantity $\|f\|_{H^p}$ *is called the* Hardy norm.

Thus H^2 is the class of functions which are analytic on the unit disk $\{z : |z| < 1\}$ and whose Taylor coefficients satisfy

$$\sum_{n=0}^{\infty} |a_n|^2 < \infty.$$

A function f is said to have *nontangential limit* L at $e^{i\theta}$, if $f(z) \to L$ as $z \to e^{i\theta}$ inside the unit circle. Functions in H^2 have nontangential limits almost everywhere on the unit circle and belong to L^2 on the unit circle ([23], p.17 and p.21). The Hardy space H^2 becomes a Hilbert space on imposing the inner product

$$\langle f, g \rangle = \frac{1}{2\pi} \int_{|z|=1} f(z)\overline{g(z)}\, |dz|.$$

Definition 3.1.2 ([14, Def. 12.6.1]) *Let* S *denote a point set and* X *a complete inner product space of functions on* S*. A function of two variables* z *and* w *in* S*,* $K(z, w)$ *is called a* reproducing kernel *for the space* X *if:*

1. *For each fixed* $w \in S$*,* $K(z, w)$ *is in* X*;*

2. *For every function* $f(z) \in X$ *and for every point* $w \in S$*, the* reproducing property

$$f(w) = \langle f(z), K(z, w) \rangle_z. \tag{3.2}$$

holds, where the subscript z *indicates* w *is fixed and the inner product is taken on the variable* z*.*

With respect to the above inner product, H^2 has an orthonormal basis given by $\{1, z, z^2, \cdots, z^n, \cdots\}$ and a reproducing kernel

$$K(z, w) = \sum_{n=0}^{\infty} z^n \bar{w}^n = \frac{1}{1 - z\bar{w}}. \tag{3.3}$$

Due to the existence of the reproducing kernel [14, Thm.12.6.1], the *point functionals* P_z for $|z| < 1$, which is defined by

$$P_z(f) = f(z),$$

are bounded linear functionals on H^2 satisfying

$$|P_z f|^2 \leq K(z, z)\, \|f\|^2. \tag{3.4}$$

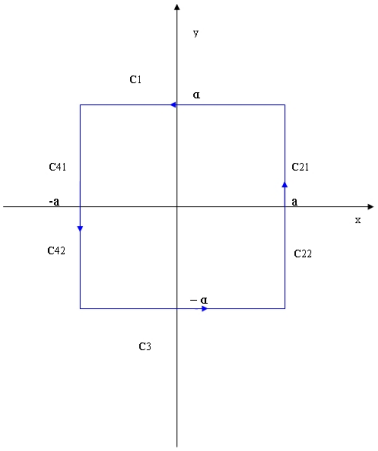


Figure 3.1: Contour for integration over the rectangle A

3.1.1 A residue theorem

In the sequel we shall use the following lemma, a version of which is given without proof in [23]. We set

$$\Psi(z) = \begin{cases} -\pi i, & \operatorname{Im} z > 0 \\ \pi i, & \operatorname{Im} z < 0. \end{cases} \tag{3.5}$$

Lemma 3.1.1 [23, Lemma 2]) *Suppose that α and h are positive real numbers and f is a function satisfying the following conditions:*

1. *f is analytic on the closure of the strip $S_\alpha = \{-\alpha < \operatorname{Im} z < \alpha\}$;*
2. *$\lim_{a\to\pm\infty} \int_{a-i\alpha}^{a+i\alpha} |f(z)|dz = 0$;*
3. *$\int_{-\infty}^{\infty} f(x)dx$ and $\sum_{n=-\infty}^{\infty} f(nh)$ exists.*

Then

$$\int_{-\infty}^{\infty} f(x)dx = \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\Psi(z)\,dz.$$

Moreover,

$$h\sum_{n=-\infty}^{\infty} f(nh) = \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\,\pi\cot\frac{\pi z}{h}\,dz.$$

Proof. First, we consider integrating f on the boundary of the rectangle $A = \{z : -\alpha < \operatorname{Im} z < \alpha \text{ and } -a < \operatorname{Re} z < a\}$ as shown in Figure 3.1. Since f is analytic on A, we have

$$\int_{C1} f + \int_{-a}^{a} f = -\int_{C21} f - \int_{C41} f$$

and

$$\int_{C3} f - \int_{-a}^{a} f = -\int_{C22} f - \int_{C42} f.$$

Combining these two equations gives

$$\int_{-a}^{a} f = \frac{1}{2}\left(-\int_{C1} f + \int_{C3} f - \int_{C21} f - \int_{C41} f + \int_{C22} f + \int_{C42} f\right).$$

Letting $a \to \infty$, by condition 2 of the hypotheses, $\int_{C21} f$, $\int_{C41} f$, $\int_{C22} f$ and $\int_{C42} f$ will each converge to zero. Thus, we get

$$\int_{-\infty}^{\infty} f(x)dx = \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\Psi(z)\,dz.$$

The function $f(z)\,\pi\cot(\pi z/h)$ has poles at nh for integer n with residues $hf(nh)$. According to the Residue Theorem, we have

$$\frac{1}{2\pi i}\int_{C1+C2+C3+C4} f = h\sum_{n=-N}^{N} f(nh),$$

where N is the largest integer such that $Nh < a$, $C2 = C21 + C22$ and $C4 = C41 + C42$. Letting $a \to \infty$, we obtain

$$h\sum_{n=-\infty}^{\infty} f(nh) = \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\pi\cot\frac{\pi z}{h}\,dz$$

as claimed. □

More generally, similar arguments will establish:

Lemma 3.1.2 *In Lemma 3.1.1, suppose that f has poles z_i with residues* $\mathrm{Res}(z_i)$ *inside S_α. Then*

$$\begin{aligned}\int_{-\infty}^{\infty} f(x)dx &= \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\Psi(z)\,dz + \pi i\sum_i \mathrm{Res}(z_i^+) - \pi i\sum_i \mathrm{Res}(z_i^-) \\ &= \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\Psi(z)\,dz - \sum_i \mathrm{Res}(z_i)\Psi(z_i) \qquad (3.6)\end{aligned}$$

and

$$h\sum_{n=-\infty}^{\infty} f(nh) = \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\pi\cot\left(\frac{\pi z}{h}\right)dz - \sum_i \mathrm{Res}(z_i)\pi\cot\left(\frac{\pi z_i}{h}\right). \qquad (3.7)$$

Here z_i^+ and z_i^- represent the poles above and below the real line.

3.1.2 Our working notation

We will work with the following quantities for $h > 0$:

1. The integral: $If = \int_{-1}^{1} f$,

2. The *tanh-sinh approximation*:

$$T_{h,N} f = h \sum_{n=-N}^{N} \frac{\pi \cosh(nh)}{2 \cosh^2\left(\frac{\pi}{2} \sinh(nh)\right)} f\left(\tanh\left(\frac{\pi}{2} \sinh(nh)\right)\right),$$

3. The N-th *approximation error*:

$$E_{h,N} f = (I - T_{h,N})\, f,$$

4. The approximation limit:

$$T_h f = \lim_{N \longrightarrow \infty} T_{h,N} f,$$

   which will be shown to exist in Lemma 3.2.1.

5. The limit error:

$$E_h = (I - T_h)\, f.$$

Then I, $T_{h,N}$ and $E_{h,N}$ are bounded linear functionals on H^2. So are T_h and E_h, once we show they exist.

3.2 The Associated Space G^2

Along with H^2, it is helpful to use the same change of variable as in (3.1) to define a corresponding space of functions [14]. Precisely, we let G^2 be the set of functions of the form

$$\psi'(w) f(\psi(w)) \qquad f \in H^2 \text{ and } \psi \text{ as in (3.1)}.$$

Assumptions. We assume ψ maps region A onto the unit disk, that functions in G^2 are analytic on A and are defined almost everywhere on ∂A.

Letting $\widehat{f}(w) = \psi'(w) f(\psi(w))$, we can induce an inner product in G^2 by

$$\begin{aligned} \langle \widehat{f}, \widehat{g} \rangle_{G^2} = \langle f, g \rangle_{H^2} &= \frac{1}{2\pi} \int_{|z|=1} f(z) \overline{g(z)} \, |dz| \\ &= \frac{1}{2\pi} \int_{\partial A} f(\psi(w)) \overline{g(\psi(w))} |\psi'(w) \, dw| \\ &= \frac{1}{2\pi} \int_{\partial A} \widehat{f}(w) \overline{\widehat{g}(w)} \, |dw/\psi'(w)|. \end{aligned} \tag{3.8}$$

Then G^2 is a Hilbert space and the mapping $f \mapsto \widehat{f}$ is an isomorphism of H^2 onto G^2. Also, G^2 has an orthonormal basis with elements

$$\phi_n(z) = \psi'(z) \left(\psi(z)\right)^n. \tag{3.9}$$

We then have the following result:

Lemma 3.2.1 *Assume the reproducing kernel of H^2 is $K(z, w)$; then the reproducing kernel of G^2 is*

$$\widehat{K}(z, w) = K\left(\psi(z), \psi(w)\right) \cdot \psi'(z) \cdot \overline{\psi'(w)} \tag{3.10}$$

given the transformation $\psi(z)$ from H^2 to G^2.

Proof. Given the reproducing property (3.2) and the definition of the inner product (3.8), if $\widehat{K}(z, w)$ denotes the reproducing kernel of G^2, then

$$\begin{aligned} \langle \widehat{f}(z), \widehat{K}(z, w) \rangle_z &= \frac{1}{2\pi} \int_{\partial A} \widehat{f}(z) \overline{\widehat{K}(z, w)} | \, dz/\psi'(z)| \\ &= \widehat{f}(w) \\ &= \psi'(w) \cdot \langle f(z), K(z, \psi(w)) \rangle_z \\ &= \psi'(w) \cdot \frac{1}{2\pi} \int_{\partial A} \widehat{f}(z) \overline{K(\psi(z), \psi(w)) \psi'(z)} | \, dz/\psi'(z)| \\ &= \frac{1}{2\pi} \int_{\partial A} \widehat{f}(z) \overline{K(\psi(z), \psi(w)) \psi'(z) \overline{\psi'(w)}} \, |dz/\psi'(z)|. \end{aligned}$$

Comparing the last integral with the first one, we get

$$\widehat{K}(z, w) = K\left(\psi(z), \psi(w)\right) \cdot \psi'(z) \cdot \overline{\psi'(w)}.$$

□

Therefore, for the tanh-sinh transformation, by (3.3), G^2 has a reproducing kernel

$$\begin{aligned}\widehat{K}(z,w) &= 1/\left(1-\psi(z)\overline{\psi(w)}\right)\cdot\psi'(z)\cdot\overline{\psi'(w)} \\ &= \frac{\pi^2}{4\cosh\left(\frac{\pi}{2}\sinh(z)-\frac{\pi}{2}\sinh(\overline{w})\right)} \\ &\times \frac{\cosh(z)\cosh(\overline{w})}{\cosh\left(\frac{\pi}{2}\sinh(z)\right)\cosh\left(\frac{\pi}{2}\sinh(\overline{w})\right)}. \end{aligned} \tag{3.11}$$

Consequently, the *point functionals* $\widehat{P}_z$ on G^2, defined for $z \in A$ by

$$\widehat{P}_z\widehat{f} = \widehat{f}(z)$$

are bounded linear functionals (see Thm. 12.6.1 of [14]) and satisfy

$$|\widehat{P}_z f|^2 \leq \widehat{K}(z,z)\,\|\widehat{f}\|^2. \tag{3.12}$$

3.2.1 Further working notation

We use similar notation in G^2 as in H^2:

$$\widehat{I}\widehat{f} = \int_{-1}^{1} f(x)dx = \int_{-\infty}^{\infty} \widehat{f}(u)du,$$

$$\widehat{T}_{h,N}\widehat{f} = T_{h,N}f = h\sum_{n=-N}^{N} \widehat{f}(nh),$$

$$\widehat{E}_{h,N}\widehat{f} = \left(\widehat{I}-\widehat{T}_{h,N}\right)\widehat{f},$$

$$\widehat{T}_h\widehat{f} = \lim_{N\longrightarrow\infty}\widehat{T}_{h,N}\widehat{f} \quad \text{and} \quad \widehat{E}_h\widehat{f} = \left(\widehat{I}-\widehat{T}_h\right)\widehat{f}.$$

Lemma 3.2.2 *For $h > 0$, the operators $\widehat{T}_h$ and $\widehat{E}_h$ are bounded linear functionals on G^2 (as are $\widehat{T}_{h,N}$ and $\widehat{E}_{h,N}$).*

Proof. Define $|\widehat{T}_{h,N}|\widehat{f} = h\sum_{n=-N}^{N}|\widehat{f}(nh)| \geq \widehat{T}_{h,N}\widehat{f}$. The inequality (3.12) shows that

$$\begin{aligned}
|\widehat{T}_{h,N}|\widehat{f} &\leq h\|\widehat{f}\|\sum_{n=-N}^{N}\widehat{K}(nh,nh)^{1/2} \\
&< h\|\widehat{f}\|\sum_{n=-\infty}^{\infty}\widehat{K}(nh,nh)^{1/2} \\
&= h\|\widehat{f}\|\sum_{n=-\infty}^{\infty}\frac{1}{\sqrt{1-\tanh^2(\pi/2\sinh(nh))}}\cdot\frac{\pi\cosh(nh)}{2\cosh^2(\pi/2\sinh(nh))} \\
&= h\|\widehat{f}\|\sum_{n=-\infty}^{\infty}\frac{\pi\cosh(nh)}{2\cosh(\pi/2\sinh(nh))} \\
&\leq h\pi\|\widehat{f}\|\sum_{n=-\infty}^{\infty}e^{nh-\pi/2|\sinh(nh)|} \\
&\leq h\pi\|\widehat{f}\|\sum_{n=-\infty}^{\infty}e^{-h|n|/2} = h\cdot\pi\cdot\coth(h/4)\,\|\widehat{f}\|, \qquad (3.13)
\end{aligned}$$

where

$$\begin{aligned}
\|\widehat{f}\|^2 = \|f\|^2 = \langle f,f\rangle &= \frac{1}{2\pi}\int_{|z|=1}f(z)\overline{f(z)}\,|dz| \\
&= \frac{1}{2\pi}\int_{|z|=1}|f(z)|^2\,|dz|. \qquad (3.14)
\end{aligned}$$

Since $f \in H^2$ belongs to L^2 on the unit circle, (3.14) is finite. Equation (3.13) shows that $\widehat{T}_{h,N}\widehat{f}$ is absolutely convergent. Also,

$$\widehat{T}_h\widehat{f} \leq h\cdot\pi\cdot\coth(h/4)\|\widehat{f}\|.$$

Thus, $\widehat{T}_h$ and also $\widehat{E}_h$ are bounded linear operators on G^2. □

If a space X has a reproducing kernel, then every bounded linear functional on X has a very simple expression. In the next result, $L_{(z)}$ denotes a functional L applied to K with respect to variable z while keeping w constant.

Theorem 3.2.1 [14, Thm 12.6.6 and Cor 12.6.7] *Let X have a reproducing kernel $K(z,w)$, and assume L is a bounded linear functional defined on X. Then*

$$\|L\|^2 = L_w\overline{L_zK(z,w)}.$$

When L is any of the previous functionals on G^2, then

$$\|L\|^2 = L_{(z)}L_{(w)}\widehat{K}(z,w), \tag{3.15}$$

since all functionals considered are real. Our goal is to estimate the approximation error

$$||E_{h,N}|| = ||\widehat{E}_{h,N}||$$

which will be computed by repeated use of Lemma 3.2.2 and (3.15).

3.3 Evaluation of the Error Norm $||\widehat{E}_{h,N}||$

Let $\widehat{\sigma}_{h,N} = \widehat{T}_h - \widehat{T}_{h,N}$. Then (3.15) implies that

$$\begin{aligned} ||\widehat{E}_{h,N}||^2 &= ||\widehat{E}_h + \widehat{\sigma}_{h,N}||^2 \\ &= \underbrace{\widehat{E}_{h,(z)}\widehat{E}_{h,(w)}\widehat{K}(z,w)}_{e_1} + \underbrace{\widehat{E}_{h,(z)}\widehat{\sigma}_{h,N,(w)}\widehat{K}(z,w)}_{e_2} \\ &\quad + \underbrace{\widehat{\sigma}_{h,N,(z)}\widehat{E}_{h,(w)}\widehat{K}(z,w)}_{e_3} + \underbrace{\widehat{\sigma}_{h,N,(z)}\widehat{\sigma}_{h,N,(w)}\widehat{K}(z,w)}_{e_4}. \end{aligned} \tag{3.16}$$

In (3.16), the error is divided into four parts denoted by e_1, e_2, e_3, and e_4. Over the rest of this section, we will evaluate these quantities term by term.

3.3.1 Evaluation of e_2 and e_3

The second and third terms of (3.16) are actually equal. Indeed:

$$\begin{aligned} e_3 = \widehat{\sigma}_{h,N,(z)}\widehat{E}_{h,(w)}\widehat{K}(z,w) &= \widehat{E}_{h,(z)}\widehat{\sigma}_{h,N,(w)}\widehat{K}(z,w) \\ &= \widehat{E}_{h,(z)}\left(h\sum_{|n|>N}\widehat{K}(z,nh)\right) \\ &= \widehat{E}_{h,(z)}\left(h\sum_{|n|>N}F_{nh}(z)\right) \\ &= h\sum_{|n|>N}\widehat{E}_h F_{nh} \end{aligned} \tag{3.17}$$

where

$$\begin{aligned} F_{nh} &= P_{nh,(w)}\widehat{K}(z,w) \\ &= \frac{\pi^2\cosh(z)\cosh(nh)}{4\cosh\left(\frac{\pi}{2}\sinh(z)-\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(z)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)}. \end{aligned} \tag{3.18}$$

To calculate (3.17), first we compute $\widehat{I}F_{nh}$:

$$\begin{aligned}\widehat{I}F_{nh} &= \int_{-\infty}^{\infty} P_{nh,(w)}\widehat{K}(z,w)dz \\ &= \int_{-\infty}^{\infty} \frac{\pi^2\cosh(z)\cosh(nh)}{4\cosh\left(\frac{\pi}{2}\sinh(z)-\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(z)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)}\,dz \\ &= \frac{\pi^2\cosh(nh)}{4\cosh\left(\frac{\pi}{2}\sinh(nh)\right)}\int_{-\infty}^{\infty}\frac{\cosh(z)}{\cosh\left(\frac{\pi}{2}\sinh(z)-\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(z)\right)}\,dz.\end{aligned}$$

Using the change of variable $u=\frac{\pi}{2}\sinh(x)$ and letting $a=\frac{\pi}{2}\sinh(nh)$, we get

$$\begin{aligned}\widehat{I}F_{nh} &= \frac{\pi\cosh(nh)}{2\cosh(a)}\int_{-\infty}^{\infty}\frac{1}{\cosh(u-a)\cosh(u)}du \\ &= \frac{\pi\cosh(nh)}{\cosh(a)}\int_{-\infty}^{\infty}\frac{1}{\cosh(a)+\cosh(2u-a)}\,du. \qquad (3.19)\end{aligned}$$

Applying the identity

$$\frac{\sinh(a)}{\cosh(a)+\cosh(b)} = \frac{1}{1+e^{b-a}} - \frac{1}{1+e^{b+a}}, \qquad (3.20)$$

we see that (3.19) equal to

$$\begin{aligned}&\frac{\pi\cosh(nh)}{\cosh(a)\sinh(a)}\int_{-\infty}^{\infty}\left(\frac{1}{1+e^{2u-2a}}-\frac{1}{1+e^{2u}}\right)du \\ =\ &\frac{\pi\cosh(nh)}{\cosh(a)\sinh(a)}\cdot a \\ =\ &\frac{\pi^2\sinh(2nh)}{2\sinh\left(\pi\sinh(nh)\right)} = \widehat{I}F_{nh}. \qquad (3.21)\end{aligned}$$

By a similar manipulation, we get $\widehat{T_n}F_{nh}$ equal to

$$\frac{\pi^2 h\cosh(nh)}{\sinh(\pi\sinh(nh))}\sum_{r=-\infty}^{\infty}\left(\frac{1}{1+e^{\pi\sinh(rh)-\pi\sinh(nh)}}-\frac{1}{1+e^{\pi\sinh(rh)}}\right)\cosh(rh). \qquad (3.22)$$

Therefore,

$$\begin{aligned}h\sum_{|n|>N}\widehat{E}_h F_{nh} &= \sum_{|n|>N}\frac{\pi^2\cosh(nh)}{\sinh\left(\pi\sinh(nh)\right)} \qquad (3.23)\\ \times\ &\left\{\sum_{r=-\infty}^{\infty}\left(\frac{1}{1+e^{\pi\sinh(rh)-\pi\sinh(nh)}}-\frac{1}{1+e^{\pi\sinh(rh)}}\right)h\cosh(rh)-\sinh(nh)\right\}.\end{aligned}$$

Consider the summation

$$S_n = \sum_{r=-\infty}^{\infty}\left(\frac{1}{1+e^{\pi\sinh(rh)-\pi\sinh(nh)}}-\frac{1}{1+e^{\pi\sinh(rh)}}\right)h\cosh(rh),$$

and note that for $h > 0$,

$$\begin{aligned} S_n &= \left|e^{\pi \sinh(nh)} - 1\right| \sum_{r=-\infty}^{\infty} \frac{e^{\pi/2 \sinh(rh)}}{e^{\pi \sinh(nh)} + e^{\pi \sinh(rh)}} \frac{h \cosh(rh)}{2 \cosh\left(\pi/2 \sinh(rh)\right)} \\ &= \sinh(\pi/2 \sinh(nh)) \sum_{r=-\infty}^{\infty} \frac{2\, e^{\pi/2 \sinh(nh)}\, e^{\pi/2 \sinh(rh)}}{e^{\pi \sinh(nh)} + e^{\pi \sinh(rh)}} \frac{h \cosh(rh)}{\cosh\left(\pi/2 \sinh(rh)\right)} \\ &\leq \sinh(\pi/2 \sinh(nh)) \sum_{r=-\infty}^{\infty} \frac{h \cosh(rh)}{\cosh(\pi/2 \sinh(rh))} \\ &= h \sinh(\pi/2 \sinh(nh)) \left\{ 1 + 2 \sum_{r=1}^{\infty} \frac{\cosh(rh)}{\cosh(\pi/2 \sinh(rh))} \right\}. \end{aligned}$$

If we denote

$$C(h) = 1 + 2 \sum_{r=1}^{\infty} \frac{\cosh(rh)}{\cosh(\pi/2 \sinh(rh))}, \tag{3.24}$$

then

$$S_n \leq h\, C(h) \sinh(\pi/2 \sinh(nh))$$

and thus by (3.23),

$$\begin{aligned} h \sum_{|n|>N} \widehat{E}_h F_{nh} &\leq \sum_{|n|>N} \frac{\pi^2 \cosh(nh)}{\sinh(\pi \sinh(nh))} \left\{ hC(h) \sinh\left(\pi/2 \sinh(nh)\right) + \sinh(nh) \right\} \\ &= \frac{\pi^2}{2} \left(\sum_{|n|>N} \frac{hC(h) \cosh(nh)}{\cosh(\pi \sinh(nh)/2)} + \sum_{|n|>N} \frac{\sinh(2nh)}{\sinh(\pi \sinh(nh))} \right). \end{aligned} \tag{3.25}$$

Since $\frac{\cosh(x)}{\cosh(\pi/2 \sinh(x))}$ is a decreasing function, we have

$$\begin{aligned} \sum_{n>N} \frac{\cosh(nh)}{\cosh(\pi/2 \sinh(nh))} &\leq \frac{1}{h} \int_{N-1}^{\infty} \frac{\cosh(x)}{\cosh(\pi/2 \sinh(x))} dx \\ &= \frac{2}{\pi h} \int_{\pi/2 \sinh(N-1)h}^{\infty} \operatorname{sech}(u) du \\ &= \frac{2}{\pi h} \left(\frac{\pi}{2} - \arctan\left(\sinh\left(\frac{\sinh(N-1)h}{\pi} \right) \right) \right). \end{aligned}$$

Also, since

$$\lim_{x \to \infty} x \left(\frac{\pi}{2} - \arctan(x) \right) = \frac{1}{2},$$

then when x is large enough,

$$O\left(\frac{\pi}{2} - \arctan\left(x\right) \right) = O\left(\frac{1}{x} \right).$$

Thus

$$\sum_{n>N} \frac{\cosh(nh)}{\cosh(\pi/2\sinh(nh))} = O\left(\frac{1}{h} e^{-\frac{e^{(N-1)h}}{\pi}}\right). \tag{3.26}$$

Also, the second summation in (3.25) is less than the first one , which gives

$$e2 = e3 = h \sum_{|n|>N} \widehat{E}_h F_{nh} = O\left(C(h) e^{-\frac{e^{(N-1)h}}{\pi}}\right). \tag{3.27}$$

3.3.2 Evaluation of e_1

In order to compute $e_1 = \widehat{E}_{h,(z)}\widehat{E}_{h,(w)}\widehat{K}(z,w)$, we have recourse to the results of Section 3.1.1. In this case, the reproducing kernel $\widehat{K}(z,\overline{w})$ has poles at

$$w = i \cdot \arcsin(2n+1) + i \cdot 2k\pi \qquad k = 0, \pm 1, \cdots$$

with residue $\pi\cosh(z)/\sinh(\pi\sinh(z))$, and poles at

$$w = \operatorname{arcsinh}(\sinh(z) + (2n+1)i) + i \cdot 2k\pi \qquad k = 0, \pm 1, \cdots$$

with residue $-\pi\cosh(z)/\sinh(\pi\sinh(z))$. Since we are dealing with real functions, $\widehat{K}(z,\overline{w}) = \widehat{K}(z,w)$, as a function of $\overline{w}$, satisfies the conditions of Lemmas 3.1.1 and 3.1.2, if we take $\alpha \leq i\arcsin(1)$.

Therefore,

$$\widehat{E}_{h,(w)}\widehat{K}(z,w) = \frac{1}{2\pi i}\int_{\partial S_\alpha} \widehat{K}(z,\overline{w})\Phi(w)\,dw$$

$$= \frac{1}{2\pi i}\int_{\partial S_\alpha} \frac{\pi^2\cosh(z)\cosh(w)\Phi(w)}{4\cosh\left(\frac{\pi}{2}\sinh(z) - \frac{\pi}{2}\sinh(w)\right)\cosh\left(\frac{\pi}{2}\sinh(z)\right)\cosh\left(\frac{\pi}{2}\sinh(w)\right)}\,dw,$$

where

$$\begin{aligned} \Phi(z) = \Phi(z;h) &= \Psi(z) - \pi\cot\frac{\pi z}{h} \\ &= \begin{cases} \frac{-2\pi i}{1-\exp(-2\pi i z/h)}, & \text{Im(z)} > 0 \\ \frac{2\pi i}{1-\exp(2\pi i z/h)}, & \text{Im(z)} < 0. \end{cases} \end{aligned} \tag{3.28}$$

Here $\Psi(x)$ is defined as in (3.5). If we let $\alpha \to \infty$ while keeping ∂S_α away from the

poles, (3.6) gives

$$\begin{aligned}
&\int_{-\infty}^{\infty} f(x)dx \\
= \;& \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\Psi(z)\,dz - \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Psi(i\arcsin(2n+1)+i2k\pi) \\
& + \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Psi\left(\operatorname{arcsinh}\left(\sinh(z)+(2n+1)i\right)+i2k\pi\right),
\end{aligned}$$

and (3.7) gives

$$\begin{aligned}
&h\sum_{n=-\infty}^{\infty} f(nh) \\
= \;& \frac{1}{2\pi i}\int_{\partial S_\alpha} f(z)\pi\cot\frac{\pi z}{h}dz - \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\pi\cot\left(\frac{\pi\cdot i\left(\arcsin(2n+1)+2k\pi\right)}{h}\right) \\
& + \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\pi\cot\left(\frac{\pi\cdot\left(\operatorname{arcsinh}(\sinh(z)+(2n+1)i)+i2k\pi\right)}{h}\right).
\end{aligned}$$

We thus have

$$\begin{aligned}
&\widehat{E}_{h,(w)}\widehat{K}(z,w) \\
= \;& \frac{1}{2\pi i}\int_{\partial S_\alpha}\widehat{K}(z,\overline{w})\Phi(w)\,dw - \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Phi(i\arcsin(2n+1)+i2k\pi) \\
& + \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Phi(\operatorname{arcsinh}(\sinh(z)+(2n+1)i)+i2k\pi).
\end{aligned}$$

Since $|\Phi(x+iy)| \sim 2\pi\exp(-2\pi|y|/h)$, the integral part will go to zero as $\alpha\to\infty$. Also, since $\Phi(-w) = -\Phi(w)$,

$$\begin{aligned}
&\frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Phi\left(i\arcsin(2n+1)+i2k\pi\right) \\
= \;& \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\lim_{N\to\infty}\lim_{K\to\infty}\sum_{n=0}^{N}\sum_{k=0}^{K}\{\Phi(i\arcsin(2n+1)+i2k\pi) \\
& + \Phi(-i\arcsin(2n+1)-i2k\pi) + \Phi(-i\arcsin(2n+1)+i2k\pi) \\
& + \Phi(i\arcsin(2n+1)-i2k\pi)\} \\
= \;& 0.
\end{aligned}$$

Therefore,

$$\widehat{E}_{h,(w)}\widehat{K}(z,w) = \frac{\pi\cosh(z)}{\sinh(\pi\sinh(z))}\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\Phi(\operatorname{arcsinh}(\sinh(z)+(2n+1)i)+i2k\pi). \quad (3.29)$$

Letting

$$H(z) = \widehat{E}_{h,(w)}\widehat{K}(z,w),$$

it is easy to see that $H(z) \in C^{\infty}$. Also, since we have shown that $\widehat{E}_h$ is a bounded linear functional and

$$\widehat{E}_h = \widehat{E}_{h,(z)}H(z) = \int_{-\infty}^{\infty} H(z)dz - h\sum_{n=-\infty}^{\infty} H(nh),$$

thus $\sum_{n=-\infty}^{\infty} H(nh)$ exists for all $h > 0$ and $H(z)$ is integrable because the integrand is independent of h. Taking $h \to 0$, the summation part will converge to zero, and so the integral is finite. Therefore, $H(z)$ satisfies the conditions of the Poisson summation formula (see the proof of [33, pages 243–244]). It follows that

$$\|\widehat{E}_h\|^2 = \widehat{E}_h\, H = -2\sum_{r=1}^{\infty}\int_{-\infty}^{\infty} H(x)\cos(2\pi rx/h)\,dx. \quad (3.30)$$

So if we set

$$\begin{aligned} H_{n,k}(x) &= \Phi\left(\operatorname{arcsinh}\left(\sinh(z)+(2n-1)i\right)+i2k\pi\right)+\Phi\left(\operatorname{arcsinh}\left(\sinh(z)-(2n-1)i\right)+i2k\pi\right) \\ &+ \Phi\left(\operatorname{arcsinh}\left(\sinh(z)+(2n-1)i\right)-i2k\pi\right)+\Phi\left(\operatorname{arcsinh}\left(\sinh(z)-(2n-1)i\right)-i2k\pi\right), \end{aligned}$$

and

$$H_{n,0}(x) = \Phi\left(\operatorname{arcsinh}\left(\sinh(z)+(2n-1)i\right)\pi\right)+\Phi\left(\operatorname{arcsinh}\left(\sinh(z)-(2n-1)i\right)\pi\right)$$

then

$$H(x) = \sum_{n=1}^{\infty}\sum_{k=0}^{\infty}\frac{\pi\cosh(x)}{\sinh(\pi\sinh(x))}H_{n,k}(x). \quad (3.31)$$

Letting $A(z,n)$ and $B(z,n)$ denote the real part and imaginary part of

$$\operatorname{arcsinh}(\sinh(z)+(2n-1)i),$$

we have

$$\begin{aligned} A(z,n) &= \operatorname{sgn}(z)\cdot\ln\left(\frac{1}{2}\sqrt{\sinh(z)^2+4n^2}+\frac{1}{2}\sqrt{\sinh(z)^2+(2n-2)^2}\right. \\ &+ \left.\sqrt{\left(\frac{1}{2}\sqrt{\sinh(z)^2+4n^2}+\frac{1}{2}\sqrt{\sinh(z)^2+(2n-2)^2}\right)^2-1}\right) \end{aligned}$$

and

$$B(z,n) = \arcsin\left(1/2\,\sqrt{\left(\sinh\left(z\right)\right)^2 + 4\,n^2} - 1/2\,\sqrt{\left(\sinh\left(z\right)\right)^2 + \left(2\,n-2\right)^2}\right)$$

where

$$\mathrm{sgn}(z) = \begin{cases} 1, & \mathrm{Re(z)}> 0 \\ -1, & \mathrm{Re(z)}< 0. \end{cases} \tag{3.32}$$

Therefore

$$\Phi(\mathrm{arcsinh}(\sinh(z) + (2n-1)i)) = \Phi(A(z,n) + B(z,n)i),$$

$$\Phi(\mathrm{arcsinh}(\sinh(z) - (2n-1)i)) = \Phi(A(z,n) - B(z,n)i).$$

Let $B_1 = B(z,n) + 2k\pi$ and $B_2 = B(z,n) - 2k\pi$. We have

$$\begin{aligned} & H_{n,k}(z) \\ = \;& \Phi\left(A + B_1 i\right) + \Phi\left(A - B_1 i\right) + \Phi\left(A + B_2 i\right) + \Phi\left(A - B_2 i\right) \\ = \;& 2\pi i\left(\frac{-1}{1-\exp\left(\frac{2\pi B_1 - i2\pi A}{h}\right)} + \frac{1}{1-\exp\left(\frac{2\pi B_1 + i2\pi A}{h}\right)}\right. \\ & \left. + \frac{-1}{1-\exp\left(\frac{-2\pi B_2 - i2\pi A}{h}\right)} + \frac{1}{1-\exp\left(\frac{-2\pi B_2 + i2\pi A}{h}\right)}\right) \\ = \;& \frac{-4\pi\sin\left(\frac{2\pi A}{h}\right)\exp\left(\frac{2\pi B_1}{h}\right)}{1-2\exp\left(\frac{2\pi B_1}{h}\right)\cos\left(\frac{2\pi A}{h}\right)+\exp\left(\frac{4\pi B_1}{h}\right)} \\ & + \frac{-4\pi\sin\left(\frac{2\pi A}{h}\right)\exp\left(\frac{-2\pi B_2}{h}\right)}{1-2\exp\left(\frac{-2\pi B_2}{h}\right)\cos\left(\frac{2\pi A}{h}\right)+\exp\left(\frac{-4\pi B_2}{h}\right)}. \end{aligned} \tag{3.33}$$

Then by (3.30) and (3.31),

$$\begin{aligned} \|\widehat{E}_h\|^2 &= -2\sum_{r=1}^{\infty}\int_{-\infty}^{\infty}\sum_{n=1}^{\infty}\sum_{k=0}^{\infty}\frac{\pi\cosh(x)}{\sinh(\pi\sinh(x))}H_{n,k}(x)\cos(2\pi rx/h)\,dx \\ &= -2\sum_{r=1}^{\infty}\sum_{n=1}^{\infty}\sum_{k=0}^{\infty}\int_{-\infty}^{\infty}\frac{\pi\cosh(x)}{\sinh(\pi\sinh(x))}H_{n,k}(x)\cos(2\pi rx/h)\,dx. \end{aligned} \tag{3.34}$$

Using a more elaborate (and computer algebra assisted) version of the argument given by [23] for tanh, we can estimate e_1 by

$$e_1 = \|\widehat{E}_h\|^2 = O\left(e^{-A/h}\right), \tag{3.35}$$

where A is a positive constant. This entails integrating the integral component of (3.34) by parts twice to obtain

$$\begin{aligned} & \int_{-\infty}^{\infty} H_{n,k}(x) \frac{\pi \cosh(x)}{\sinh(\pi \sinh(x))} \cos(2\pi r x/h) dx \\ = & -\left(\frac{h}{2\pi r}\right)^2 \int_{-\infty}^{\infty} \left(H_{n,k}(x) \frac{\pi \cosh(x)}{\sinh(\pi \sinh(x))} \right)'' \cos\left(2\pi r x/h\right)\, dx, \end{aligned} \tag{3.36}$$

which is

$$O\left(\frac{e^{-A/h} e^{-k\pi/h}}{r^2 n^2}\right).$$

Our numerical experiments indicate that A is some constant bounded below by $5/2$ and above by 10.

3.3.3 Evaluation of e_4

We have:

$$\begin{aligned} e_4 &= \widehat{\sigma}_{h,N,(z)} \widehat{\sigma}_{h,N,(w)} \widehat{K}(z,w) \\ &= \frac{h^2\pi^2}{4} \sum_{|r|,|n|>N} \frac{\cosh(nh)\cosh(rh)}{\cosh\left(\frac{\pi}{2}\sinh(nh) - \frac{\pi}{2}\sinh(rh)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(rh)\right)} \end{aligned}$$

which can be written as

$$\frac{h^2\pi^2}{2} \sum_{r,n>N} \frac{\cosh(nh)\cosh(rh)}{\cosh\left(\frac{\pi}{2}\sinh(nh) - \frac{\pi}{2}\sinh(rh)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(rh)\right)} \tag{3.37}$$

$$+\frac{h^2\pi^2}{2} \sum_{r,-n>N} \frac{\cosh(nh)\cosh(rh)}{\cosh\left(\frac{\pi}{2}\sinh(nh) - \frac{\pi}{2}\sinh(rh)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(rh)\right)}. \tag{3.38}$$

Then, because of (3.26), (3.38) is equal to

$$\begin{aligned} & \frac{h^2\pi^2}{2} \sum_{r,n>N} \frac{\cosh(nh)\cosh(rh)}{\cosh\left(\frac{\pi}{2}\sinh(nh) + \frac{\pi}{2}\sinh(rh)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(rh)\right)} \\ \leq & \frac{h^2\pi^2}{2} \left(\sum_{n>N} \frac{\cosh(nh)}{\cosh(\pi/2\sinh(nh))} \right)^2 = O\left(e^{-\frac{2e^{(N-1)h}}{\pi}}\right). \end{aligned} \tag{3.39}$$

Also, (3.37) can be written as

$$\frac{h^2\pi^2}{2}\sum_{n>N}\frac{\cosh^2(nh)}{\cosh^2(\pi/2\sinh(nh))}$$
$$+\quad 2\sum_{n>r>N}\frac{\cosh(nh)\cosh(rh)}{\cosh\left(\frac{\pi}{2}\sinh(nh)-\frac{\pi}{2}\sinh(rh)\right)\cosh\left(\frac{\pi}{2}\sinh(nh)\right)\cosh\left(\frac{\pi}{2}\sinh(rh)\right)}.$$

The first term above is less than

$$O\left(e^{-\frac{2e^{(N-1)h}}{\pi}}\right) \tag{3.40}$$

and the second term is less than the first. Then (3.39) and (3.40) together give

$$e_4=\widehat{\sigma}_{h,N,(z)}\widehat{\sigma}_{h,N,(w)}\widehat{K}(z,w)=O\left(e^{-\frac{2e^{(N-1)h}}{\pi}}\right). \tag{3.41}$$

3.4 The Main Results

Combining (3.27), (3.35) and (3.41) gives the following result:

Theorem 3.4.1 (Tanh-sinh convergence.) *For $f\in H^2$ and $\psi(x)=\tanh(\frac{\pi}{2}\sinh(x))$, the error bound can be evaluated as:*

$$||\widehat{E}_{h,N}||^2=||E_{h,N}||^2=O\left(e^{-A/h}\right)+O\left(C(h)e^{-\frac{e^{(N-1)h}}{\pi}}\right), \tag{3.42}$$

where the order constant is independent of both N and h. Here as before

$$C(h)=1+2\sum_{r=1}^{\infty}\frac{\cosh(rh)}{\cosh(\pi/2\sinh(rh))}\leq 1+\frac{4}{\pi h}. \tag{3.43}$$

Proof. We estimate

$$\begin{aligned}||\widehat{E}_{h,N}||^2=||E_{h,N}||^2 &= e_1+e_2+e_3+e_4\\ &= O\left(e^{-A/h}\right)+O\left(C(h)e^{-\frac{e^{(N-1)h}}{\pi}}\right)+O\left(e^{-\frac{2e^{(N-1)h}}{\pi}}\right)\\ &= O\left(e^{-A/h}\right)+O\left(C(h)e^{-\frac{e^{(N-1)h}}{\pi}}\right).\end{aligned}$$

To obtain the estimate for $C(h)$, observe that

$$hC(h)\leq h+2\int_0^{\infty}\frac{\cosh(t)}{\cosh(\pi/2\sinh(t))}\,dt=h+\frac{4}{\pi}\int_0^{\infty}\operatorname{sech}\left(\frac{\pi}{2}x\right)\,dx,$$

since the first integrand is decreasing. The latter integral evaluates to 1. Therefore, we have

$$C(h) \leq 1 + \frac{4}{\pi h}$$

as claimed. □

Our result is similar to the one given by Sugihara (compare the result in part two of Theorem 2.3.1 with (3.42)). In either case, the quadratic convergence of the double exponential method is shown. However, our analysis differs from the approach of Sugihara in two aspects:

1. In [49], Sugihara analyzed the performance of the double exponential quadrature formula over the space H^∞. This space contains the functions analytic on the strip

   $$\mathcal{D}_d = \{z \in \mathbb{C} \,|\, |\mathrm{Im\,z}| < d\}$$

   and has the same decay rate as the characterizing function $\omega(z)$. The space $H^\infty(\mathcal{D}_d, \omega)$ is a space of functions after transformation, consisting of the functions of the form $\psi'(w)f(\psi(w))$. Our analysis is carried out on the original function space H^2.

   Also, note the fact that the transformation $\tanh(z/2)$ maps the unit circle onto the strip $\mathcal{D}_{\pi/2}$. When studying the tanh rule with $\psi(w) = \tanh(z/2)$, the analysis on the H^∞ (the Hardy space with infinity norm) is equivalent to the analysis on the space $H^\infty(\mathcal{D}_{\pi/2}, \omega)$ with $\omega = 1/2\cosh(z/2)^2$. But in general, these two spaces are not equivalent.

2. The norm used in the two spaces are different. The H^2 space and new space G^2 are Hilbert spaces with the H^2 norm, but $H^\infty(\mathcal{D}_{\pi/2}, \omega)$ is not a complete space even in the case of the tanh rule.

3.5 Analysis for Error Function and Other Transformations

As shown in Figure 2.2, the tanh, tanh-sinh and error function have very similar properties: they are absolutely continuous monotonic increasing functions mapping $(-\infty, \infty)$ onto $(-1, 1)$ and are infinitely differentiable, with every order of their derivatives quickly vanishing at infinity. It is not surprising that a similar approach can be

applied to analyze the convergence properties of the error function and more generally to any transformation with the properties above.

For example, the error function quadrature uses the transformation:

$$\psi(x) = \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{3.44}$$

with

$$\psi'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}.$$

As in Section 3.2, one can introduce a corresponding space ${G_*}^2$, which contains the set of functions of the form

$$\psi'(w) f\left(\psi(w)\right),$$

where $f \in H^2$ and ψ as in (3.44). Then G_*^2 has a reproducing kernel

$$\begin{aligned} \widehat{K}(z,w) &= 1/\left(1 - \psi(z)\overline{\psi(w)}\right) \cdot \psi'(z) \cdot \overline{\psi'(w)} \\ &= \frac{4}{\pi\left(1 - \mathrm{erf}(z)\mathrm{erf}(w)\right)} \cdot e^{-z^2} e^{-w^2}. \end{aligned}$$

The approximation error $||\widehat{E}_{h,N}||$ can still be divided into 4 parts as in (3.16) with

$$e_2 = e_3 = h \sum_{|n|>N} \widehat{E}_h F_{nh}$$

and

$$e_4 = \frac{4}{\pi} \sum_{|r|,|n|>N} \frac{e^{-r^2h^2} e^{-n^2h^2}}{1 - \mathrm{erf}(rh)\mathrm{erf}(nh)},$$

where

$$\begin{aligned} F_{nh} &= P_{nh,(w)} \widehat{K}(z,w) \\ &= \frac{4}{\pi} \frac{e^{-z^2} e^{-n^2h^2}}{1 - \mathrm{erf}(z)\mathrm{erf}(nh)}. \end{aligned}$$

In order to evaluate e_1, one must determine the poles of

$$\frac{1}{1 - \mathrm{erf}(z)\mathrm{erf}(w)}.$$

Also, the evaluation of e_2 and e_3 involves the careful estimation of

$$\sum_{n=a}^{\infty} \sum_{r=b}^{\infty} \frac{e^{-r^2h^2}}{1 - \mathrm{erf}(rh)\mathrm{erf}(nh)}.$$

We will not give further analysis here. For any other quadrature rules with similar properties, a similar approach can be taken.

3.6 Numerical Convergence and Examples

We conclude with a few numerical illustrations and a brief discussion of the optimality of the coefficients of the infinite trapezoidal rule.

3.6.1 Numerical results

We exhibit results for the following set of test functions, earlier studied by Borwein and Bailey in [4]:

1. $\int_0^1 t\log(1+t)\,dt \;=\; 1/4$
2. $\int_0^1 t^2\arctan t\,dt \;=\; (\pi-2+2\log 2)/12$
3. $\int_0^{\pi/2} e^t\cos t\,dt \;=\; (e^{\pi/2}-1)/2$
4. $\int_0^1 \frac{\arctan(\sqrt{2+t^2})}{(1+t^2)\sqrt{2+t^2}}\,dt \;=\; 5\pi^2/96$
5. $\int_0^1 \sqrt{t}\log t\,dt \;=\; -4/9$
6. $\int_0^1 \sqrt{1-t^2}\,dt \;=\; \pi/4$
7. $\int_0^1 \frac{\sqrt{t}}{\sqrt{1-t^2}}dt \;=\; 2\sqrt{\pi}\Gamma(3/4)/\Gamma(1/4) \;=\; \beta(1/2,3/4)/2$
8. $\int_0^1 \log^2 t\,dt \;=\; 2$
9. $\int_0^{\pi/2} \log(\cos t)\,dt \;=\; -\pi\log(2)/2$
10. $\int_0^{\pi/2} \sqrt{\tan t}\,dt \;=\; \pi\sqrt{2}/2$
11. $\int_0^{\infty} \frac{1}{1+t^2}\,dt \;=\; \pi/2$
12. $\int_0^{\infty} \frac{e^{-t}}{\sqrt{t}}\,dt \;=\; \sqrt{\pi}$
13. $\int_0^{\infty} e^{-t^2/2}\,dt \;=\; \sqrt{\pi/2}$
14. $\int_0^{\infty} e^{-t}\cos t\,dt \;=\; 1/2$

Note that the first four of these integrals involve well-behaved continuous functions on finite intervals. The integrands in 5 and 6 have an infinite derivative at an endpoint. In 7 through 10, there is an integrable singularity at an endpoint. In 11 through 13,

Level	Prob. 1	Prob. 2	Prob. 3	Prob. 4	Prob. 5	Prob. 6	Prob. 7
1	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}	10^{-5}	10^{-6}
2	10^{-11}	10^{-11}	10^{-9}	10^{-9}	10^{-12}	10^{-12}	10^{-12}
3	10^{-24}	10^{-19}	10^{-21}	10^{-18}	10^{-28}	10^{-25}	10^{-26}
4	10^{-51}	10^{-38}	10^{-49}	10^{-36}	10^{-62}	10^{-50}	10^{-49}
5	10^{-98}	10^{-74}	10^{-106}	10^{-73}	10^{-129}	10^{-99}	10^{-98}
6	10^{-195}	10^{-147}	10^{-225}	10^{-145}	10^{-265}	10^{-196}	10^{-194}
7	10^{-390}	10^{-293}	10^{-471}	10^{-290}	10^{-539}	10^{-391}	10^{-388}
8	10^{-777}	10^{-584}	10^{-974}	10^{-582}		10^{-779}	10^{-777}
9							
10							
11							
Level	Prob. 8	Prob. 9	Prob. 10	Prob. 11	Prob. 12	Prob. 13	Prob. 14
1	10^{-5}	10^{-4}	10^{-6}	10^{-2}	10^{-2}	10^{-1}	10^{-1}
2	10^{-12}	10^{-11}	10^{-12}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
3	10^{-29}	10^{-24}	10^{-25}	10^{-11}	10^{-9}	10^{-6}	10^{-5}
4	10^{-62}	10^{-50}	10^{-48}	10^{-22}	10^{-15}	10^{-9}	10^{-8}
5	10^{-130}	10^{-97}	10^{-98}	10^{-45}	10^{-28}	10^{-19}	10^{-14}
6	10^{-266}	10^{-195}	10^{-194}	10^{-91}	10^{-50}	10^{-37}	10^{-26}
7	10^{-540}	10^{-389}	10^{-388}	10^{-182}	10^{-92}	10^{-66}	10^{-48}
8		10^{-777}	10^{-777}	10^{-365}	10^{-170}	10^{-126}	10^{-88}
9				10^{-731}	10^{-315}	10^{-240}	10^{-164}
10					10^{-584}	10^{-457}	10^{-304}
11						10^{-870}	10^{-564}

Table 3.1: Tanh-sinh errors at level m for the 14 test problems [4]

the interval of integration is infinite, requiring a transformation such as $s = 1/(t+1)$. Integrals 14 is an oscillatory function on an infinite interval.

Table 3.1 [4] gives the actual error resulting from using the summation (2.8) as an approximation to the specified integral, rounded to the nearest power of ten. The precision used in these calculations is 1000 digits, so no results are shown with errors less than 10^{-1000}. The "level" shown in the Table 3.1 is an index of the interval h, according to the rule $h = 2^{-m}$, where m is the level. In other words, at each successive level, the interval h is halved.

It can be seen from the results in the table that the tanh-sinh scheme achieves approximately exponential convergence, in the sense that the number of correct digits approximately doubles with each level. In some cases, such as Problems 3, 5 and 8, the convergence is slightly better than a doubling with each level; in other cases, such

as Problems 12, 13 and 14, the rate is slightly less. It is remarkable that Problems 1, 6, 7, 9 and 10 exhibit virtually identical convergence behavior.

3.6.2 Non-optimality of the limiting trapezoidal rule

Next, we examine the optimality of the coefficient of the infinite trapezoidal rule $\widehat{T}_h$ with $\psi(x)$ as defined in (3.1). Let $A = \{\ldots, a_{-1}, a_0, a_1 \ldots\}$ be the sequence of coefficients for $\widehat{T}_h$ such that

$$\widehat{T}_{h;A}\widehat{f} = \sum_{n=-\infty}^{\infty} a_n \widehat{f}(nh)$$

for every $\widehat{f} \in G^2$, and

$$\widehat{E}_{h;A} = \widehat{I} - \widehat{T}_{h;A}.$$

We may ask whether A_h with $a_n \equiv h$, is *optimal* in the sense that

$$\|\widehat{E}_{h;A_h}\| = \inf \|\widehat{E}_{h;A}\|$$

for all A's for which $\widehat{T}_A$ is a bounded linear functional. As shown in [23], A_h is optimal if and only if $\widehat{T}_h$ integrates F_{nh} exactly for every integer n. But from the analysis in Section 3.3.1, we know $\widehat{E}_h F_{nh} \neq 0$. Therefore, A_h is not optimal.

3.6.3 A nonexistence theorem

As we have seen in the previous chapter, the basic idea of the variable transformation technique is to transfer the endpoint singularities of a function f to infinity, so that the new integrand will have a faster decay rate at infinity. This point is demonstrated by (3.16) and (3.42), where the error norm can be divided into 2 parts: the limiting trapezoidal error

$$\|E_{h,N}\|^2 = O\left(e^{-A/h}\right)$$

and the truncation error

$$O\left(C(h)e^{-\frac{e^{(N-1)h}}{\pi}}\right).$$

Here Nh is the length of interval where the abscissas are sampled.

In practice, Nh will normally be kept constant throughout the computation. If $Nh = 20$ (as in [9]), the truncation error will roughly be 10^{-10^7} which is ignorable

compared to the limiting trapezoidal error. Then the overall error is approximately $O\left(e^{-A/h}\right)$. This explains the quadratic convergence of the tanh-sinh quadrature exhibited in Table 3.1.

Furthermore, by comparing the error estimation given by (2.11) and (3.42) for the tanh and tanh-sinh rules, one can observe that a transformation with faster decay rate (in this case tanh-sinh) converges faster than the one with lower decay rate (such as tanh and erf). Consequently, one may ask a question: is it possible to find a more efficient transformation $\psi(t)$, whose decay rate is faster than double exponential? The answer is no as shown by the following result.

Theorem 3.6.1 ([49, Thm 4.1]) *There exists no function $\omega(z)$ simultaneously satisfying the following conditions:*

1. *$\omega(z) \in B(\mathcal{D}_d)$;*
2. *$\omega(z)$ does not vanish within $\mathcal{D}_d$ and takes real values on the real line;*
3. *The decay rate of $\omega(z)$ is specified as*

$$\omega(x) = O\left(\exp\left(-\beta \exp\left(\gamma|x|\right)\right)\right) \quad \text{as } |x| \to \infty$$

where $\beta > 0$ and $\gamma > \pi/(2d)$.

Here, $\mathcal{D}_d$ and $B(\mathcal{D}_d)$ are defined as in Definition 2.3.1.

Chapter 4

Multidimensional Integration and Parallel Implementation

In Chapter 3, we carried out an analysis on the theoretical error bound of the tanh-sinh method. This chapter is mainly devoted to the discussion on the implementation side. We start by introducing the parallel 1-D tanh-sinh algorithm proposed by Borwein and Bailey [4]. We then show how the 1-D tanh-sinh method is generalized to higher dimensions. In particular, we focus on the discussion of the 2-D tanh-sinh rule and its parallel implementation in Section 4.2. We also propose a revised parallel 2-D tanh-sinh method based on the approach taken by Borwein and Bailey [4]. We finish our discussion by briefly introducing some applications of the double exponential transformation to various other research fields in Section 4.3.

4.1 Parallel Implementation

In Chapter 2, we introduced the tanh-sinh method as one of the best high precision integration algorithms. We also showed that the tanh-sinh method can often achieve quadratic convergence ratio even for quite unruly integrands. However, high precision numerical integration can be very expensive. For some hard problems, a sequential implementation might be too slow to get the desired accuracy. As an example, consider a conjecture proposed by Borwein and Broadhurst [4, 10]:

$$\frac{24}{7\sqrt{7}} \int_{\pi/3}^{\pi/2} \log \left| \frac{\tan t + \sqrt{7}}{\tan t - \sqrt{7}} \right| dt \tag{4.1}$$

$$\stackrel{?}{=} \sum_{n=0}^{\infty} \left[\frac{1}{(7n+1)^2} + \frac{1}{(7n+2)^2} - \frac{1}{(7n+3)^2} + \frac{1}{(7n+4)^2} - \frac{1}{(7n+5)^2} - \frac{1}{(7n+6)^2} \right].$$

In order to numerically verify this equation, they computed the left hand side integral to 20000 digits and compared it with the result given by summation at the right hand side of (4.1). A sequential implementation will take 2751357 seconds on an Apple G5 based computer [4], which is roughly one month running.

Furthermore, as we will see later in Section 4.2, the computational cost for high dimensional quadrature is much more expensive than in one dimension. For example, some of the recent researches on Ising integrals involves the evaluation of 3-D and 4-D integrals to high precision (ranging from 100 to 500 digits), which will take more than one year to finish in sequential [5]. Therefore, the application of parallel system to high precision integration problem is quite necessary.

Before we step into the discussion of the parallel tanh-sinh algorithm, we will give a very brief survey to some basic concepts and terminologies in parallel computing.

4.1.1 parallel computing

Simply speaking, *parallel computing* is the employment of multiple processors to work simultaneously on the same problem. There are many different types of parallel systems. A very popular taxonomy of parallel (and serial) computer systems was proposed by Flynn [17]. Based on the notions of instruction and data streams, it classifies computer systems into the following 4 categories:

- *SISD (single instruction and single data stream)* A SISD machine represents a typical serial computer, in which one instruction is operated on a single data stream by a single processing unit during one clock cycle.

- *MISD (multiple instruction and single data stream)* MISD defines the systems utilizing multiple processing units to execute different instructions with the same data input independently.

- *SIMD (single instruction and multiple data stream)* SIMD involves the simultaneous execution of the same instruction with different data streams by multiple processors. SIMD is best suited for the applications with high regularities such as image and video processing. Some of machines such as the GAPP, CM-2 and MasPar MP-1 belong to this category.

- *MIMD (multiple instruction and multiple data stream)* In the MIMD model, individual processors may execute different instructions or programs with different data streams. Most of the modern parallel computers fall into this category. Examples of MIMD systems include IBM SP2, the Cray T3D and Beowulf Cluster.

Depending on their memory architectures, parallel computers can also be classified as either *shared memory* or *distributed memory* systems [41]. In a *shared memory* parallel machine, a central main memory is shared by all processors as global address space. All processors have ability to access all memory and the change made by one processor is visible to all others. On the other hand, in a *distributed memory* parallel system, all processors have their own local memory. Each processor only has direct access to its local memory.

There are several parallel programming models such as threads, shared memory, message passing and hybrid models. Here we will only introduce the message passing model. For a complete discussion of various programming models and languages, one can read Chapter 3 of [21]. The message passing model assumes a distributed memory architecture so that all communication across processors is explicit, which takes place by sending and receiving messages among processors. Message passing implementations typically consists of a set of communication libraries which need to be called within the source code. It is the responsibility of programmers to manage the communication and determine all parallelism. There are several message passing implementations, in which *MPI* (*Message Passing Interface*) is the most popular one and has became the industry standard for message passing. A detailed description of MPI can be found at [18, 22].

In order to measure the effectiveness of a parallel algorithm, the concept of *speedup ratio* is widely used in the parallel literature. Let T_p denote the run time of a parallel algorithm with p processors and T_s denote the run time of the corresponding fastest sequential algorithm. The *speedup ratio* S_p is defined as [41]

$$S_p = \frac{T_s}{T_p}, \tag{4.2}$$

which measures the performance of the parallel program versus the number of processors. Instead of a sequential algorithm, if we use the execution time of the same parallel algorithm on one processor as T_s, (4.2) defines the *relative speedup ratio* [34]. It is not difficult to see that $1 \leq S_p \leq p$ and larger S_p's imply better parallelism. When $S_p = p$, a parallel algorithm is said to exhibit a *linear speedup ratio*. Sometimes it is possible to observe a speedup ratio larger than the number of processors used. In this case, the algorithm is said to achieve a *superlinear speedup*, which seems quite unusual since it exceeds the theoretical maximum. However, the superlinear

speedup is possible due to the cache effect that small portions of data normally fit better into the cache memories of the individual processors. As a result, the efficiency of each processor in a parallel system is higher than in the case when all data resides in a single processor system.

4.1.2 1-D parallel tanh-sinh rule

In general, numerical integration algorithms, especially automatic schemes, are very suitable for parallel implementation by their nature. Recall from (1.3)

$$I(f) \approx A(f) = \sum_{i=1}^{n} w_i f(x_i),$$

the approximation of $I(f)$ requires the computation of abscissa-weight pairs x_j,w_j and the function evaluations $f(x_j)$, both of which can be done based solely on the specific sample point (i.e no correlation between the abscissas).

Among all the numerical quadrature rules we have introduced, the variable transformation based algorithms, such as the tanh-sinh rule, are especially suited for parallel implementation. First, the abscissa-weight pairs are independent of each integrand. They can be calculated once at the very beginning and reused for the subsequent computations. Also, the abscissas are evenly distributed among the interval. During the next level of computation, the step size h will be halved. Therefore, the approximation result from the previous levels can be reused.

A straightforward way to implement the tanh-sinh quadrature scheme in parallel would be:

1. Distribute the computation of x_j and w_j among processors so that each of them calculates some of the abscissa-weight pairs;

2. For every integrand $f(x)$, each processor computes $w_j f(x_j)$ associated with its own pairs to the desired tolerance;

3. The root processor collects the partial sum from each processor and adds them up to get the final result.

The only communication required in this parallel implementation is in Step 3 when the root processor collects partial sums. In theory, the communication cost will be

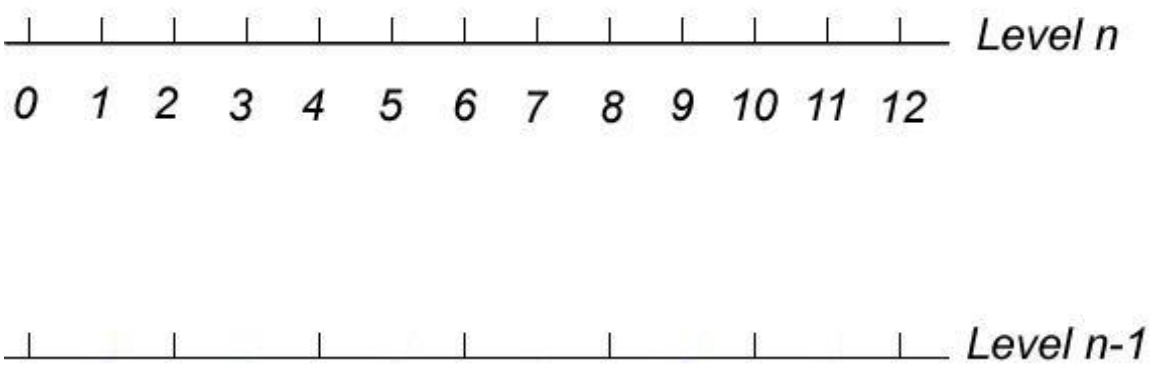


Figure 4.1: Potential problem for cyclic distribution

constant for different accuracies and test functions. Assuming each processor has the same computational capacity, one needs to find a good distribution scheme so that in Step 1 the abscissa-weight pairs can be evenly distributed among processors. It is also required that in Step 2, the desired accuracy can be achieved at (almost) the same level in different processors for every integrand, so that there is no imbalance in the computational cost.

It should be noted that different test functions may need different levels of computation to achieve the same accuracy. The computational load can be extremely unbalanced if we use the following straightforward cyclic distribution scheme:

$$\rho(j) = j \bmod m,$$

where m is the number of processors, j is the index of abscissas and $\rho(j)$ is the distribution function. Figure 4.1 shows an example. Assume the program initializes abscissa-weight pairs to level n and processor 0 is assigned all even number pairs and processor 1 is assigned all odd number pairs. However some problems may only need $n-1$ level of computation. In such cases, all odd number abscissas will be skipped in our algorithm. This results in processor 0 doing all the work while processor 1 is idle during the whole computation period. Since many parallel systems have a processor count that is power of 2, this situation can happen very frequently.

Algorithm 6 parallel 1-D tanh-sinh quadrature rule

Input: the maximal level m, functions f, tolerance ϵ

Output: the approximation hS

Initialize:
Set $s =$ mpi_comm_size(), $j =$ mpi_comm_rank() and $h := 2^{-m}$
for $k := 0$ to $20 \cdot 2^m$ **do**
  if $j =$ mod $(k + \lfloor k/16 \rfloor, s)$ **then**
    Set $t := kh$
    Set $x_k := \tanh(\pi/2 \cdot \sinh t)$ and $w_k := \pi/2 \cdot \cosh t / \cosh^2(\pi/2 \cdot \sinh t)$
    if $|x_k - 1| < \epsilon$ **then**
      exit **do**
    end if
  end if
end for
Set $n_t = \min(k_j, j = 0 \ldots s)$, where k_j is the value of k at exit for processor j.

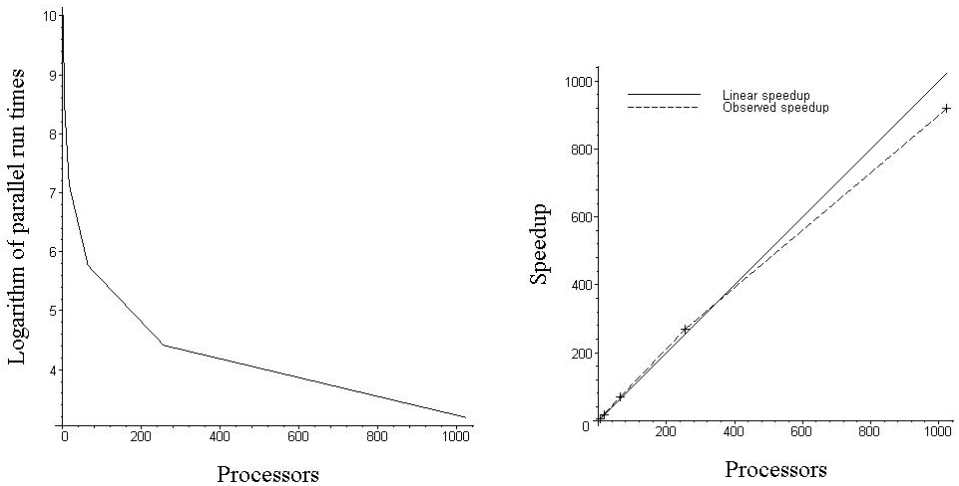
Quadrature:
Set $S := 0$ and $h := 1$.
for $k := 1$ to m (or until the desired accuracy is obtained) **do**
  $h := h/2$.
  for $i := 0$ to n_t step 2^{m-k} **do**
    if $j =$ mod $(k + \lfloor k/16 \rfloor, s)$ **then**
      if mod $(i, 2^{m-k+1}) \neq 0$ or $k = 1$ **then**
        if $i = 0$ **then**
          $S := S + w_0 f(0)$
        else
          $S := S + w_i(f(-x_i) + f(x_i))$
        end if
      end if
    end if
  end for
end for
Root processor collects partial sum S_j from each processor.
Result $= \sum_{j=0}^{s-1} hS_j$.

In [4], Borwein and Bailey compared five different distribution schemes and concluded that a *mixed cyclic, block-cyclic scheme* ($MCBC$)

$$\rho(j) = \mod\ (j + \lfloor j/16 \rfloor, m)$$

can provide the most balanced load across different number of processors and computational levels. A parallel implementation of the tanh-sinh rule is given in Algorithm 6.

In Table 4.1 [4], we show the test result of the parallel tanh-sinh algorithm on the function suite introduced in Section 3.6.1. The input tolerance for each test function is 2000 digits. The first line depicts the run time in the initialization procedure. The second column gives the level of computation to achieve the desired accuracy. It should be noted that the speedup ratios are absolute (i.e the single processor timing is given by an optimized sequential program). The experiment was made by Borwein and Bailey on "System X", an Apple Xserve G5 cluster at Virginia Technical University which uses the IBM XL Fortran compiler and IBM C compiler [4]. Figure 4.2(a) shows the logarithm of the parallel run time observed in tanh-sinh algorithm as a function of the number of processors and Figure 4.4(b) plots the corresponding speedup ratios. It is clear that the parallel 1-d tanh-sinh method can exhibit very good speedup performance: suplinear speedup for up to 256 processors and close-to-linear speedup for 1024 processors.



(a) 1-D parallel tanh-sinh method (b) corresponding speedup

Figure 4.2: Performance of 1-D parallel tanh-sinh method

Problem Number	Levels Required	Processors					
		1	4	16	64	256	1024
Init		4329.04	1085.34	271.87	68.88	17.73	5.02
1	10	480.07	101.63	25.55	6.45	1.65	0.53
2	10	1403.63	294.32	74.04	18.83	4.99	1.54
3	10	1421.99	317.01	79.69	20.42	5.24	1.83
4	10	1553.24	328.73	82.13	20.84	5.52	1.63
5	9	236.68	51.62	12.90	3.30	0.93	0.30
6	10	26.31	5.62	1.42	0.36	0.11	0.05
7	10	52.65	11.46	2.87	0.72	0.20	0.10
8	9	234.06	50.98	12.85	3.26	0.90	0.27
9	10	1552.38	333.24	83.60	21.34	5.44	1.84
10	10	1138.78	245.45	61.39	15.73	3.99	1.44
11	11	25.30	5.17	1.30	0.33	0.09	0.04
12	12	655.03	161.99	40.71	10.20	2.65	0.80
13	13	871.99	216.50	54.13	13.65	3.52	0.97
14	13	8291.43	1826.02	457.02	114.84	29.48	7.87
Total		22272.58	5035.08	1261.47	319.15	82.44	24.23
Speedup		1.00	4.42	17.66	69.79	270.17	919.22

Table 4.1: Run times (in seconds) and speedup ratios for 1-D problems [4]

4.2 Higher Dimensional Integration

As we have introduced in Chapter 1, the multidimensional integration is often very complicated. The quadrature rules developed for one dimension can't be directly applied to multiple integrals in general. However, the variable transformation based algorithm introduced in Chapter 4 can very easily be generalized to high dimensions. For example, consider integrating a 2-D function, for which (2.8) can be extended to

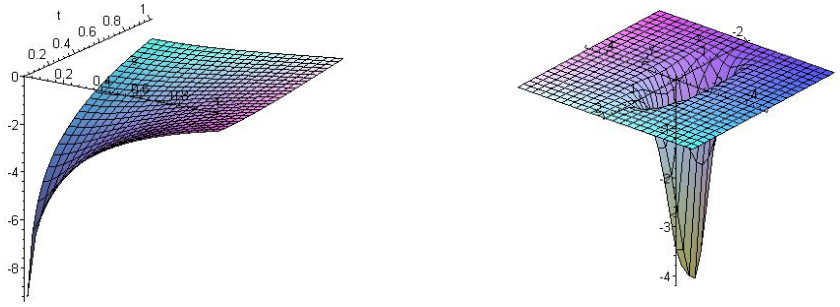
$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(g(s),g(t)\right)g'(s)g'(t)dsdt \\ &= h\sum_{k=-N}^{N}\sum_{j=-N}^{N} w_j w_k f(x_j,x_k) + E(h)\end{aligned}$$

For the integration of the more general form

$$\int_a^b \int_{c(y)}^{d(y)} f(x,y)dxdy,$$

one can use the transformation

$$x = \frac{d(y)-c(y)}{2}s + \frac{d(y)+c(y)}{2}$$



(a) $f(s,t) = \ln(2 - \cos(s) - \cos(t))$ (b) $f(x,y)$ after the DE transformation

Figure 4.3: Comparison of the original function and transformed function

$$y = \frac{b-a}{2}t + \frac{b+a}{2}.$$

The higher dimensional tanh-sinh rule can be derived similarly as in two dimensions.

Figure 4.3(a) shows the picture of the function $\ln(2 - \cos(s) - \cos(t))$, which has a singular point at $(0,0)$. As in one dimension, the 2-D tanh-sinh rule removes the singularity by converting the integrand to a new function with double exponential decay rate (See Figure 4.3(b)) at infinity.

When integrating well-behaved functions, the multidimensional tanh-sinh rule can often achieve quadratic convergence. However, high-order tanh-sinh schemes appear to be more sensitive to singularities on the boundary. It has been observed that the number of correct digits only increases roughly 1.4 times, when the level of computation increases by one [4]. Moreover, the computational cost is much greater in high dimension than in one dimension. In an n-dimension tanh-sinh rule, when one more level of computation is conducted, the cost will be increased 2^n times. This makes high precision integration extremely hard. Without the employment of a parallel computer, even in 2-D quadrature it is not possible to obtain high accuracy (say, more than 500 digits) in a reasonable time.

Here we compare the performance of the tanh-sinh rule with the performance of Quasi-Monte Carlo method from 2 to 7 dimensions. The test function package we use was originally proposed by Genz [19, 20]. All the integrands will be integrated on an n cube $[0,1]^n$. This package consists of six families of test integrands with different

attributes:

Function	**Property**
1) $f_1 = \cos\left(2\pi b_1 + \sum_{i=1}^{n} a_i x_i\right)$	Oscillatory
2) $f_2 = \prod_{i=1}^{n} \frac{1}{a_i^{-2} + (x_i - b_i)^2}$	Product Peak
3) $f_3 = \left(1 + \sum_{i=1}^{n} a_i x_i\right)^{-n-1}$	Corner Peak
4) $f_4 = \exp\left(-\sum_{i=1}^{n} a_i^2 \left(x_i - b_i\right)^2\right)$	Gaussian
5) $f_5 = \exp\left(-\sum_{i=1}^{n} a_i \left\|x_i - b_i\right\|\right)$	Piecewise Continue
6) $f_6 = \begin{cases} 0, & x_1 > b_1 \text{ or } x_2 > b_2 \\ \exp\left(\sum_{i=1}^{n} a_i x_i\right), & \text{Otherwise.} \end{cases}$	Discontinuous

The vector $\vec{a} = (a_1, a_2, \ldots a_n)$ acts as an *effective parameter* and $\vec{b}$ is a *shift parameter*. The level of difficulties of the integrals will normally increase as $||\vec{a}||$ increases. In order to get a comprehensive comparison result, multiple instances for each integrand family should be calculated with randomly chosen $\vec{a}$ and $\vec{b}$. For simplicity, we will only test one instance for each integrand family with fixed parameters in this thesis as specified below.

problem	1	2	3	4	5	6
a_i	10	2	1	2	1	2
b_i	0.1	0.5	1	0.5	0.5	0.5

The run was made on the D-Drive server, which is a dual-processor 3-GHz Xeon system with 2GB RAM. As we have shown in Subsection 1.5.1, it is extremely expensive for Monte Carlo integration to obtain high precision result. One additional correct digit will roughly increase the computational cost by a factor of 100. Therefore, the targeted accuracy for each integrand is limited to 5 digits in all dimensions. The test results are shown in Table 4.2. The third column of the table lists the maximal level of computation in the tanh-sinh algorithm in every dimension. Since computational costs in high dimensions is extremely expensive, we limit the maximal level to 4 as the dimension exceeds 3. The numbers enclosed with round brackets in Table 4.2 are the run times where the tanh-sinh method was computed to the maximal level but did not achieve the desired accuracy for the corresponding problems and dimensions.

The tanh-sinh method clearly does not work well for problem 5 and 6, where the integrands have nondifferentiable points inside the integration regions. For problem

Dimension	Maximal Level	Problem Number 1	2	3	4	5	6
		tanh-sinh method					
2	10	0.02	0.03	0.005	0.02	37	(157)
3	10	0.43	1.3	0.1	0.4	4038	(54513)
4	4	18	124	0.4	20	(1505)	(647)
5	3	346	5233	2	191	(2548)	(1110)
6	2	6979	(3323)	28	4456	(3417)	(1332)
7	1	1727	(1241)	345	(1196)	(937)	(341)
		Quasi-Monte Carlo					
2		0.1	0.02	0.15	0.002	0.002	0.08
3		0.01	0.02	0.19	0.017	0.018	0.085
4		1.3	0.3	0.2	0.002	0.021	5
5		1.38	3.5	0.2	0.002	0.023	54.7
6		7.13	7.6	2.12	0.002	0.025	117.43
7		7.6	22.25	2.33	0.002	0.028	255.45

Table 4.2: Performance(in seconds) of DE and qMC methods in dimensions 2-7

5, the tanh-sinh method can only achieve the desired accuracy in dimension 2 and 3. In both cases, the run time is much longer than the time needed in other problems. The situation for problem 6 is even worse: no satisfactory result was obtained even in two or three dimensions where ten levels of abscissas-weight pairs are used.

On the other hand, the Quasi-Monte Carlo method is not as sensitive to the nonsmoothness of the integrands as the tanh-sinh method. The required accuracy can be obtained for all six families of integrands within a reasonable time. It can also be shown from the results that the Quasi-Monte Carlo method degrades only slightly for high dimension. For problems 3, 4, and 5, there are no dramatic increases in the run times as the number of dimension increases. For the remaining test problems, the run times roughly increase by a factor of three as dimension increases by one. Compared with the tanh-sinh algorithm, the Quasi-Monte Carlo method is preferred for integrals in high dimension ($N > 10$) for low accuracy (normally 4 or 5 digits). The tanh-sinh algorithm is more suited for high precision integrations in low dimensions.

The parallel version of a high dimensional tanh-sinh scheme is very similar to the 1-D parallel algorithm. The core part of the algorithm is to find a good assignment scheme, so that the computation of the abscissa and weight pairs can be equally distributed among processors in the whole space. In [4], Borwein and Bailey propose

a double MCBC scheme for the 2-D tanh-sinh quadrature rule. This scheme first assigns 16 consecutively numbered processors to each column, where the processors p satisfying $\lfloor p/16 \rfloor \equiv j + \lfloor j/16 \rfloor \mod m$ are assigned to column j. Then within each column, the function evaluations are distributed among these 16 processors according to the 1-D MCBC scheme. By employing an MCBC scheme along each axis, a double MCBC scheme can yield a relatively balanced distribution of computational cost among the processors.

The implementation of the 2-D parallel tanh-sinh rule has been tested on the following eight functions [4]. Integrands 4 and 6 have blow-up singularities at their boundaries of integration. Integrand 1 has a non-differentiable point at $(0, 0)$.

$$1 \;:\; \int_0^1 \int_0^1 \sqrt{s^2+t^2}\, ds\, dt \;=\; \sqrt{2}/3 - \log(2)/6 + \log(2+\sqrt{2})/3$$

$$2 \;:\; \int_0^1 \int_0^1 \sqrt{1+(s-t)^2}\, ds\, dt \;=\; -\sqrt{2}/3 - \log(\sqrt{2}-1)/2 + \log(\sqrt{2}+1)/2 + 2/3$$

$$3 \;:\; \int_{-1}^1 \int_{-1}^1 (1+s^2+t^2)^{-1/2}\, ds\, dt \;=\; 4\log(2+\sqrt{3}) - 2\pi/3$$

$$4 \;:\; \int_0^\pi \int_0^\pi \log[2-\cos s - \cos t]\, ds\, dt \;=\; 4\pi G - \pi^2 \log 2$$

$$5 \;:\; \int_0^\infty \int_0^\infty \sqrt{s^2+st+t^2}\, e^{-s-t}\, ds\, dt \;=\; 1 + 3/4 \cdot \log 3$$

$$6 \;:\; \int_0^1 \int_0^1 (s+t)^{-1}[(1-s)(1-t)]^{-1/2}\, ds\, dt \;=\; 4G$$

$$7 \;:\; \int_0^1 \int_0^t (1+s^2+t^2)^{-1/2}\, ds\, dt \;=\; -\pi/12 - 1/2 \cdot \log 2 + \log(1+\sqrt{3})$$

$$8 \;:\; \int_0^\pi \int_0^t (\cos s \sin t) e^{-s-t}\, ds\, dt \;=\; 1/4 \cdot (1+e^{-\pi})$$

The test results are listed in Table 4.3 from [4], with a target accuracy of 100 digits in each case. The "level" column gives the level of computation to achieve the desired accuracy. Each run can yield the desired accuracy within 9 levels, except for integrand 4 and 6 in which one additional level will be needed. Figure 4.4 shows the speedup ratios observed in the 2-D tanh-sinh algorithm and the logarithm of the corresponding parallel run time. Again, the experiment was originally tested and reported by Borwein and Bailey in [4] on the same parallel system as described earlier in subsection 4.1.2.

It can be observed that the speedup of the 2-D parallel algorithm is not as good

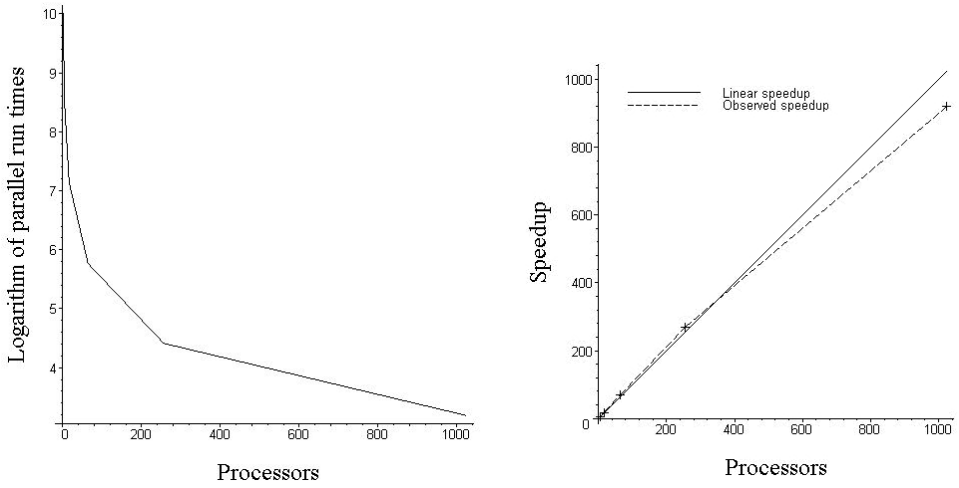
Problem Number	Levels Required	Processors 1	16	64	256	1024
1	9	1246.26	96.42	24.66	7.05	3.33
2	6	19.03	1.52	0.46	0.27	0.73
3	7	82.79	6.56	1.91	0.64	1.17
4	9	15310.44	1194.52	305.11	81.88	24.40
5	9	2209.86	170.84	44.38	12.23	4.62
6	9	1552.87	120.86	30.80	8.67	3.37
7	6	21.79	1.72	0.54	0.28	0.73
8	6	113.04	8.90	2.87	1.08	1.51
Total		20556.08	1601.34	410.73	112.10	39.86
Speedup		1.00	12.84	50.05	183.37	515.71

Table 4.3: Parallel run times (in seconds) and speedup ratios for 2-D problems [4]

as the 1-D algorithm. The work load is much harder to evenly distribute in the 2-D settings. Also, numerical error estimation is much more complicated in two dimensions.

A problem in the double MCBC rule is that once assigning a group of 16 processors to each column, the distribution of the processors within the same group along each row is fixed. As an example, assume 256 processors are used in the computation. By the double MCBC rule, they are divided into 16 groups, which are assigned to columns according to the 1-D MCBC rule. In this case, group 0 contains processor 0 to 15 and it is assigned to column $0, 31, 46, \ldots$. However, the distribution of processor 0 to 15 remains the same within each one of these columns. This means the double MCBC rule will accumulate the difference between the maximum and minimum number of indices for the same group of processors. Assuming in each column, processor 0 is assigned the maximum number of indices, processor 1 is assigned the minimum number of indices and the difference is 2. If we have 1600 columns, then the difference is exaggerated to 200.

In order to solve this problem, here we propose a revised double MCBC rule. Instead of employing 1-D MCBC rule along each column, the revised double MCBC rule applies 1-D MCBC rule separately to the sets of columns belonging to different groups. For example, consider applying the revised MCBC rule to the example above. For group 0, we will have a 1-D MCBC distribution starting with processor 0 along column 0. In column 31, the first abscissa is assigned to processor 1. By applying 1-D



(a) 2-D parallel tanh-sinh method (b) corresponding speedup

Figure 4.4: Performance of 2-D parallel tanh-sinh method

MCBC scheme separately for each group along columns, the revised double MCBC rule intends to provide a more uniform distribution to avoid difference accumulations. Table 4.2 compares the distribution results by the original double MCBC and revised double MCBC rule. In the test, a maximum 10 levels of pairs (approximately 600000000 indices) are assigned to various number of processors. The maximum and minimum number of pairs assigned to any processor, as well as their differences, are shown in table 4.2. It is obvious from the result that the revised MCBC rule is always better than the original MCBC rule.

4.3 Applications of the Tanh-Sinh Rule

In Chapter 2 and 4, we have shown the remarkable effectiveness of the tanh-sinh method for the integration of analytic functions with end-point singularities. It turns out that the double exponential transformation is also useful in the evaluation of other kinds of integrals, such as indefinite integrals and Cauchy principal value integrals. Recently, it is also found that the idea of double exponential transformation can be used in various Sinc methods. In fact, the double exponential transformation has been successfully applied in various areas, such as the boundary element method, molecular physics and fluid dynamics. In the rest of this section, we will briefly discuss the applications of the DE transformation in the fields of Fourier integrals,

		MCBC			Revised MCBC		
Proc.	Stride	Min	Max	Diff.	Min	Max	Diff.
16	1	37722625	37747200	24575	37745664	37745665	1
16	2	9424129	9436416	12287	9435648	9435649	1
16	4	2352769	2358912	6143	2358528	2358529	1
16	8	586561	589632	3071	589440	589441	1
16	16	145825	147360	1535	147264	147265	1
64	1	9429505	9437184	7679	9435264	9436800	1536
64	2	2355457	2359296	3839	2358336	2359104	768
64	4	587905	589824	1919	589344	589728	384
64	8	146497	147456	959	147216	147408	192
64	16	36385	36864	479	36744	36840	96
256	1	2356225	2359296	3071	2357664	2359200	1536
256	2	588289	589824	1535	589008	589776	768
256	4	146689	147456	767	147048	147432	384
256	8	36481	36864	383	36660	36852	192
256	16	9025	9216	191	9114	9210	96
1024	1	587905	589824	1919	588262	589802	1540
1024	2	146497	147456	959	146672	147448	776
1024	4	36385	36864	479	36472	36864	392
1024	8	8977	9216	239	9020	9216	196
1024	16	2185	2304	119	2206	2304	98

Table 4.4: Min/max processor counts for MCBC and revised MCBC

Sinc approximation and indefinite integration.

4.3.1 Integration of oscillatory functions

Even though the double exponential transformation is very effective in dealing with integrands with singularities, it does not work very well for oscillatory functions. The original double exponential transformation needs to be slightly revised to improve the performance. As an example, consider the numerical evaluation of the integral

$$\int_0^\infty \frac{\sin(x)}{x} = \frac{\pi}{2}. \tag{4.3}$$

It is clear from Figure 4.5(a) that the integrand $\sin(x)/x$ is an oscillatory function slowly vanishing at infinity. After applying the double exponential transformation

$$x = \exp\left(\frac{\pi}{2}\sinh(t)\right)$$

over the half infinite interval [36], the new integrand becomes

$$g = \frac{\pi\cosh(t)}{2}\sin\left(\exp\left(\frac{\pi}{2}\sinh(t)\right)\right).$$

The graph of the new integrand g is shown in Figure 4.5(b). Not only is g an oscillatory function, but also the dominating factor of g, $\cosh(t)$, diverges at infinity. This will make integration very slow.

In [39], Ooura and Mori suggested a special kind of variable transformation for more efficient evaluation of the following Fourier-type integrals:

$$I_s = \int_0^\infty f(x)\sin(wx)dx$$

and

$$I_c = \int_0^\infty f(x)\cos(wx)dx$$

where $f(x)$ is a function slowly vanishing at infinity.

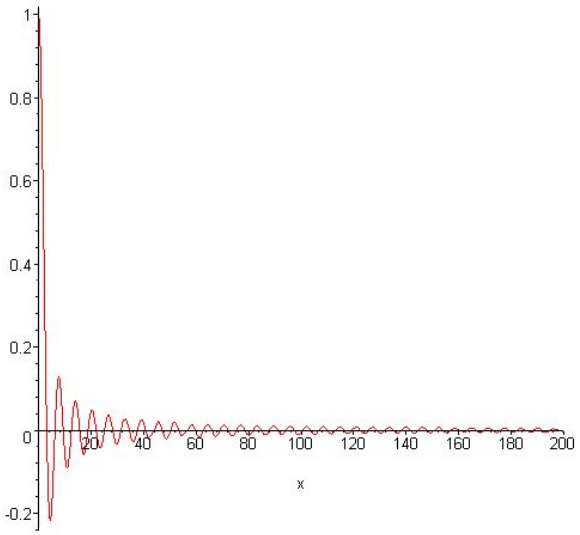
The transformations they proposed are

$$x = M\phi(t) \tag{4.4}$$

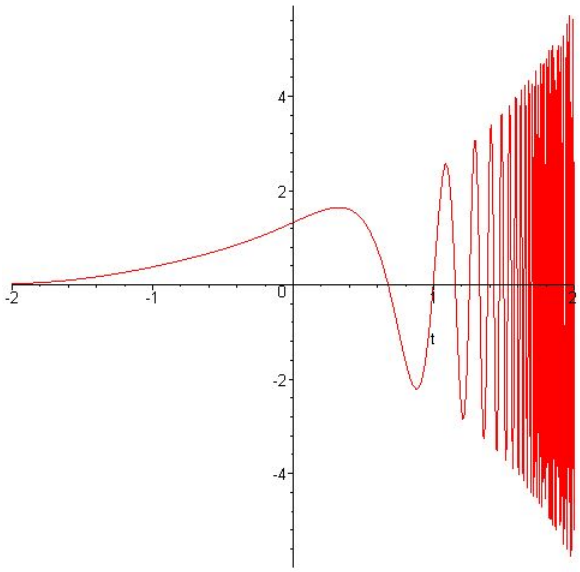
for I_s and

$$x = M\phi\left(t - \frac{\pi}{2Mw}\right) \tag{4.5}$$

for I_c. Here M is a constant satisfying $wMh = \pi$ and $\phi(t)$ is a function such that



(a) an oscillatory integral $\sin(x)/x$



(b) new integrand $\pi \cosh(t) \sin\left(\exp\left(\frac{\pi}{2}\sinh(t)\right)\right)/2$ after applying the DE transformation

Figure 4.5: An example in which the DE transformation does not work well

- $\phi(-\infty) = 0$ and $\phi(\infty) = \infty$;
- $\phi'(t) \to 0$ double exponentially as $t \to -\infty$ and $\phi(t) \to t$ double exponentially as $t \to \infty$.

Applying the trapezoidal rule after the transformation, we get:

$$I_s = Mh \sum_{k=-\infty}^{\infty} f\left(M\phi(kh)\right) \sin\left(wM\phi(kh)\right) \phi'(kh) \tag{4.6}$$

$$I_c = Mh \sum_{k=-\infty}^{\infty} f\left(M\phi\left(kh - \frac{\pi}{2M}\right)\right) \cos\left(wM\phi\left(kh - \frac{\pi}{2M}\right)\right) \phi'\left(kh - \frac{\pi}{2M}\right). \tag{4.7}$$

The idea here is that when kh is positive large enough, by the property 2 of function ϕ,

$$\sin\left(wM\phi(kh)\right) \approx \sin\left(wMkh\right) = \sin\left(k\pi\right) = 0,$$

and

$$\cos\left(wM\phi\left(kh - \frac{\pi}{2Mw}\right)\right) \approx \cos\left(wMkh - \frac{\pi}{2}\right) = \cos\left(k\pi - \frac{\pi}{2}\right) = 0.$$

Therefore, the abscissas converge double exponentially to the zeros of $\sin(wx)$ or $\cos(wx)$. On the other hand, when kh is negative, the integrand will be dominated by $\phi'(kh)$, which converges to zero double exponentially. Therefore, we can truncate the infinite summation at some constant N.

One particular function $\phi(t)$ suggested by Ooura [39] is

$$\phi(t) = \frac{t}{1 - \exp\left(-6\sinh t\right)}. \tag{4.8}$$

Later in [40], Ooura and Mori proposed an improved transformation:

$$\phi(t) = \frac{t}{1 - \exp\left(-2t - \alpha\left(1 - e^{-t}\right) - \beta\left(e^{t} - 1\right)\right)} \tag{4.9}$$

with $\beta = \frac{1}{4}$ and

$$\alpha = \frac{1}{4\sqrt{1 + M\ln(1+M)/(4\pi)}}.$$

As an example, consider applying transformation (4.4) to the integral (4.3) with $h = 0.1$, $M = 10\pi$ and $\phi(t)$ defined as in (4.8). Figure 4.6 shows the picture of the new integrand. The circles represent the location of sampling points. It is clear from the picture that as t increases, the abscissas move closer and closer to the zeros of the new integrand.

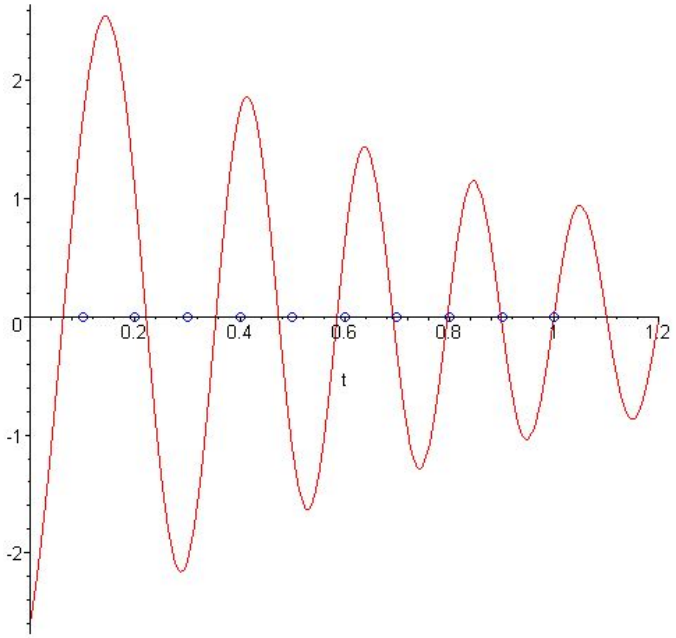


Figure 4.6: Graph of function $\sin(10\pi\phi(t))\phi'(t)/\phi(t)$ with $\phi(t)$ defined in (4.8)

4.3.2 Evaluation of Fourier transformation

One of the difficulties when applying (4.6) and (4.7) is that the formulae depend on the choice of w. If one wants to compute a set of test integrals with different frequencies (such as the evaluation of Fourier series), sampling points $\pi\phi(kh)/(wh)$ need to be recalculated. In order to overcome this drawback, Ooura [38] proposed a new method of evaluating Fourier integrals, which makes use of fixed abscissas.

Consider the computation of the Fourier transform

$$F(w) = \int_0^\infty f(x)\exp(iwx)dx. \tag{4.10}$$

After applying the transformation (4.4), one gets

$$F(w) = \int_{-\infty}^\infty f\left(M\phi(t)\right)\exp\left(iwM\phi(t)\right)M\phi'(t)dt. \tag{4.11}$$

Let $\hat{\phi}(t) = \phi(t) - t$ and w_0 be a positive constant and define

$$E(w) = \int_{-\infty}^\infty f\left(M\phi(t)\right)\exp\left(iwM\phi(t) - iw_0M\hat{\phi}(t)\right)M\phi'(t)dt \tag{4.12}$$

It can be shown that $E(w)$ vanishs exponentially for large M:

$$|E(w)| = O\left(\exp\left(-d'M\min(w, w_0)\right)\right),$$

where d' is a positive constant related to f. Therefore, instead of directly computing $F(w)$, we use

$$\begin{aligned}\tilde{F}(w) &= F(w) - E(w) \\ &= \int_{-\infty}^{\infty} f\left(M\phi(t)\right) \exp\left(iwM\phi(t) - \frac{i}{2}w_0 M\hat{\phi}(t)\right) \cdot 2iM \sin\left(\frac{1}{2}w_0 M\hat{\phi}(t)\right) \phi'(t)dt.\end{aligned}$$

as an approximation of $F(w)$. Furthermore, since $\hat{\phi}(t) \to 0$ double exponentially as $t \to \infty$ and $\phi'(t) \to 0$ double exponentially as $t \to -\infty$, the integrand of $\tilde{F}(w)$ will be a bell-shaped function to which the trapezoidal rule is applicable. The restriction $wMh = \pi$ is not necessary anymore, so that the sampling points can be fixed throughout the whole computation.

For the evaluation of integral with $w \in (0, 2w_0)$, Ooura suggested using $M = \pi/\left(w_0 h\right)$ [38]. This leads to the double exponential formula for the Fourier transform [38, Formula 2]:

$$\begin{aligned}\tilde{F}_h^N(w) &= \frac{2\pi i}{w_0} \sum_{-N}^{N} f\left(\frac{\pi}{w_0 h}\phi(nh)\right) \exp\left(\frac{\pi i w}{w_0 h}\phi(nh) - \frac{\pi i}{2h}\hat{\phi}(nh)\right) \\ &\quad \cdot \sin\left(\frac{\pi}{2h}\hat{\phi}(nh)\right) \phi'(nh) \qquad (4.13)\end{aligned}$$

Moreover, if $\phi(x)$ is defined as in (4.9), the error of (4.13) can be estimated as

$$O\left(\exp\left(-c_0/h\right)\right) + O\left(\exp\left(-c_1 w/h\right)\right) + O\left(\exp\left(-c_2(2w_0 - w)/h\right)\right)$$

where the c_i's are positive constants related to f.

4.3.3 Sinc method

The Sinc methods are a class of numerical methods based on the *Sinc approximation* which can be written as

$$f(x) \approx \hat{f}(x) = \sum_{k=-n}^{n} f(kh) S(k,h)(x). \qquad (4.14)$$

The function

$$S(k,h)(x) = \frac{\sin \pi(x - kh)/h}{\pi(x - kh)/h} \qquad (4.15)$$

is called the *Sinc function*. Sinc methods play a very important role in various fields such as function approximation, solutions of PDEs, definite and indefinite integration, approximation of Hilbert transforms and solutions of integral equations. It has been observed that Sinc method is particularly useful when solving problems with singularities, boundary layers and over infinite domain. A more detailed survey of Sinc methods can be found in [46, 47].

Let $H^1(S_\alpha)$ denote the Hardy space over the strip $S_\alpha = \{-\alpha < \mathrm{Im} z < \alpha\}$. The space $H^1(S_\alpha)$ consists of functions f regular in the interior of S_α and satisfying the boundary condition [51]

$$\lim_{\epsilon \to 0} \int_{\partial S_\alpha(\epsilon)} |f(z)||dz| < \infty,$$

where

$$S_\alpha(\epsilon) = \{z \in \mathbb{C} |\ |\mathrm{Re} z| < 1/\epsilon \text{ and } \ |\mathrm{Im}\, z| < d(1-\epsilon)\}.$$

We have the following theorem by Stenger [46].

Theorem 4.3.1 [46] *Let α, β and d be positive constants and assume f satisfies the conditions:*

1. $f \in H^1(S_d)$
2. $|f(x)| \leq \alpha \exp(-\beta|x|)$ *for all* $x \in \mathbb{R}$.

Then

$$\sup_{-\infty<x<\infty} \left| f(x) - \sum_{k=-N}^{N} f(kh) S(k,h)(x) \right| \leq C N^{1/2} \exp\left(-(\pi d \beta N)^{1/2}\right) \tag{4.16}$$

for some constant C, where the step size h is taken as

$$h = \left(\frac{\pi d}{\beta N}\right)^{1/2}.$$

Theorem 4.3.1 states that if f decays exponentially along the real line, then the convergence ratio of the infinite Sinc approximation (approximation over the interval $(-\infty, \infty)$) is $O\left(\exp(-\sqrt{N})\right)$. For definite Sinc approximation over an interval $[a, b]$, we can use the same approach as the variable transformation in numerical integration: transform the domain of integration to $(-\infty, \infty)$ and apply the infinite Sinc approximation.

Theorem 4.3.2 [46] *For a transformation $\psi(z)$ such that $\psi : (-\infty, \infty) \to (a, b)$ with $\lim_{z\to-\infty} \psi(z) = a$ and $\lim_{z\to\infty} \psi(z) = b$, if the transformed function $f(\psi(z))$ satisfies assumptions 1 and 2 in Theorem 4.3.1, then*

$$\sup_{-\infty<x<\infty} \left| f(x) - \sum_{k=-N}^{N} f(\psi(kh))S(k,h)(\psi^{-1}(x)) \right| \leq CN^{1/2} \exp\left(-(\pi d\beta N)^{1/2}\right) \quad (4.17)$$

for some constant C and the step size h is taken as

$$h = \left(\frac{\pi d}{\beta N}\right)^{1/2}.$$

In the previous theorems, the assumption on the decay rate of the function is required, but it's not necessary to be single exponential. Inspired by the analogy between Sinc approximation and numerical integration, Sugihara successfully applied the DE transformation to the Sinc approximation and proved the convergence rate to be $O\left(\exp\left(-C'n/\ln n\right)\right)$ [51].

Theorem 4.3.3 [51] *Let α, β, γ and d be some positive constants and assume f satisfies the conditions:*

1. *$f \in H^1(S_d)$*

2. *$|f(x)| \leq \alpha \exp(-\beta \exp(\gamma|x|))$ for all $x \in \mathbb{R}$.*

Then

$$\sup_{-\infty<x<\infty} \left| f(x) - \sum_{k=-N}^{N} f(kh)S(k,h)(x) \right| \leq C \exp\left(\frac{-\pi d\gamma N}{\ln(\pi d\gamma N/\beta)}\right) \quad (4.18)$$

for some constant C, where the step size h is taken as

$$h = \frac{\ln(\pi d\gamma N/\beta)}{\gamma N}.$$

Just as in numerical integration, Sugihara carried out a similar analysis of Sinc approximation over function spaces with different decay rates. The near optimality of the Sinc approximation is established when the double exponential transformation is used [50].

4.3.4 Indefinite integration

It can be shown that the Sinc approximation and the trapezoidal rule are connected by the following identity [35]:

$$\int_{-\infty}^{\infty} \hat{f}(x) = h \sum_{k=-n}^{n} f(kh). \tag{4.19}$$

Immediately, we get the following equation:

$$\int_{-\infty}^{\infty} \left(\hat{f}(x) - f(x) \right) dx = h \sum_{k=-n}^{n} f(kh) - \int_{-\infty}^{\infty} f(x)dx \tag{4.20}$$

This implies that functions whose integration can be approximated very well by the trapezoidal rule often have good Sinc approximations as well, and vice versa.

Stenger [45] first applied the Sinc method to the indefinite integration problem. Later in [24], Haber suggested two improved formulae for numerical integration based on Sinc approximation. However, their analysis is mainly based on the single exponential transformation. Following the same approach as Haber, Mori and Muhammad [37] applied double exponential transformation in their analysis and proved the optimality of the DE rule as in the definite integral.

Consider evaluating an indefinite integral

$$I(s) = \int_{-1}^{s} f(x)dx, \quad -1 < s < 1. \tag{4.21}$$

Applying the transformation

$$x = \phi(t) = \tanh\left(\frac{\pi}{2}\sinh t\right)$$

to (4.21), we get

$$\int_{-1}^{s} f(x)dx = \int_{-\infty}^{\tau} u(t)dt, \tag{4.22}$$

where $u(t) = f\left(\phi(t)\right)\phi'(t)$ and $\tau = \phi^{-1}(s) = \operatorname{arcsinh}\left(\frac{\pi}{2}\sinh s\right)$. Assume $u(t)$ is regular in the strip S_d for some $d > 0$ and the behavior of f at boundary points satisfies

$$f(x) = \begin{cases} O\left((1-x)^{-1+\alpha_+}\right) & \text{as x} \to 1 \\ O\left((1+x)^{-1+\alpha_-}\right) & \text{as x} \to \text{-1} \end{cases} \tag{4.23}$$

for some $\alpha_+, \alpha_- > 0$. If we substitute the Sinc expansion of

$$u(t) = \sum_{-\infty}^{\infty} u(kh)S(k,d)(t) + e(t,h,d)$$

into the integral (4.22) and exchange the order of summation and integration, we get

$$\int_{-\infty}^{\tau} u(t)dt = h\sum_{k=-\infty}^{\infty} u(kh)\left(\frac{1}{2}+\frac{1}{\pi}\mathrm{Si}\left(\pi\frac{\tau}{h}-\pi k\right)\right)+O\left(h\exp\left(-\frac{\pi d}{h}\right)\right) \quad (4.24)$$

where $\mathrm{Si}(x)$ is the *sine integral*

$$\mathrm{Si}(x) = \int_0^x \frac{\sin z}{z}dz. \quad (4.25)$$

In order to make the computation optimal, the infinite summation needs to be truncated at some constant N so that the truncation error equals the discretization error:

$$\begin{aligned} & h\sum_{k=N+1}^{\infty} u(kh)\left(\frac{1}{2}+\frac{1}{\pi}\mathrm{Si}\left(\pi\frac{\tau}{h}-\pi k\right)\right)+h\sum_{k=-N-1}^{-\infty} u(kh)\left(\frac{1}{2}+\frac{1}{\pi}\mathrm{Si}\left(\pi\frac{\tau}{h}-\pi k\right)\right) \\ = \;& O\left(h\exp\left(-\frac{\pi d}{h}\right)\right). \end{aligned}$$

A careful examination of this equation yields the optimal step size

$$h = \frac{1}{N}\ln\frac{2dN}{\alpha'},$$

where $\alpha' = \min(\alpha_-, \alpha_+)$.

This gives us the double exponential formula for the indefinite integral [37]:

$$\begin{aligned} \int_{-1}^{s} f(x)dx &= h\sum_{k=-N}^{N} f(\phi(kh))\,\phi'(kh)\left(\frac{1}{2}+\frac{1}{\pi}\mathrm{Si}\left(\pi\frac{\phi^{-1}(s)}{h}-\pi k\right)\right) \quad (4.26) \\ &+ O\left(\exp\left(-\frac{\pi dN}{\ln(2dN/\alpha')}\right)\right). \end{aligned}$$

It can also be shown that the new integrand

$$g(t) = f(\phi(t))\,\phi'(t)\left(\frac{1}{2}+\frac{1}{\pi}\mathrm{Si}\left(\frac{\pi}{h}\left(\phi^{-1}(s)-t\right)\right)\right)$$

is regular in the strip S_d, so that Theorem 3.6.1 applies. Therefore, just as in definite integration, the double exponential transformation is extremal and $O(\exp(-C'n/\ln n))$ is optimal in the indefinite integration.

Chapter 5

Conclusion and Remarks

This thesis discusses the double exponential formula for numerical integration and its application in relation to other fields. In particular, we provide a proof to the quadratic convergence property of the tanh-sinh method over the Hardy space. We also present a revised parallel implementation of the tanh-sinh rule in two dimensions based on the approach taken by Borwein and Bailey [4]. As discussed in previous chapter, the double exponential formula, when applied to various areas, can often achieve excellent performance and be proven to be (near) optimal under certain circumstances.

In the future, some of the following schemes can also be explored:

- *Analysis on more general spaces:* The analysis of the DE formula often rest on the space of analytic functions. In practice, the integrands can have nondifferentiable points inside the integration region and therefore do not belong to any spaces introduced in Section 2.3 and 3.2. The performance of the DE formula should be analyzed on a more generalized space.

- *Analysis of the multidimensional tanh-sinh rule:* In this thesis, we concentrate on one dimensional tanh-sinh quadrature. It has been shown that the 2-D tanh-sinh rule does not work as well as in one dimension. However, there is no theoretical approach to the high order double exponential formulae at present.

- *Applications to Sinc methods:* We have shown that the double exponential formula is closely related to the theory of Sinc approximation and indefinite integral. The previous applications of Sinc methods, such as PDEs and integral equation, mainly employ a single exponential transformation. It will be interesting to substitute with a DE formula and examine their performance.

Bibliography

[1] Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, 1989.

[2] David H. Bailey. High-precision arithmetic in scientific computation. *Computing in Science and Engineering*, pages 54–61, May 2005.

[3] David H. Bailey and Jonathan M. Borwein. Effective error bounds for euler-maclaurin-based quadrature schemes. *D-drive preprint 297*, 2005.

[4] David H. Bailey and Jonathan M. Borwein. Highly parallel, high-precision numerical integration. *D-drive preprint 294*, 2005.

[5] David H. Bailey, Jonathan M. Borwein, and Richard E. Crandall. Integrals of the ising class. *Journal of Physics A: Mathematical and General*, 2006.

[6] David H. Bailey, Yozo Hida, Xiaoye S. Li, and Brandon Thompson. *ARPREC: An Arbitrary Precision Computation Package*.

[7] David H. Bailey, Xiaoye S. Li, and Karthik Jeyabalan. A comparison of three high-precision quadrature schemes. *Experimental Mathematics*, 14(3):317–329, 2005.

[8] Jonathan M. Borwein and David H. Bailey. *Mathematics by Experiment, Plausible Reasoning in the 21st Century*. A K Peters, Ltd, 2004.

[9] Jonathan M. Borwein, David H. Bailey, and Roland Girgensohn. *Experimentation in Mathematics, Computational Paths to Discovery*. A K Peters, Ltd, 2004.

[10] Jonathan M. Borwein and D. Broadhurst. *Determination of Rational Dirichlet-zets Invariants of Hyperbolic Manifolds and Feynman Knots and Links*.

[11] Jonathan Mark Bull. *Parallel Algorithms For Globally Adaptive Quadrature*. PhD thesis, University of Manchester, 1997.

[12] Maui High Performance Computing Center. Introduction to parallel programming. Online tutorial available at: *http://www.mhpcc.edu/training/workshop/parallel_intro/MAIN.html*.

[13] Richard E. Crandall. *Topics in Advanced Scientific Computation*. Springer-Verlag, Heidelberg, 1996.

[14] Philip J. Davis. *Interpolation and Approximation*. Blaisdell, New York, 1963.

[15] P.L. Duren. *Theory of H^p spaces*. Academic Press, New York, 1970.

[16] H. Engels. *Numerical Quadrature and Cubature*. Academic Press, 1980.

[17] M.J. Flynn. Very high speed computing systems. *Proc. IEEE*, 54:1901–1909, 1966.

[18] Message Passing Interface Forum. Mpi: A message-passing interface standard. *International Journal of Supercomputer Applications and High Performance Computing*, 8(3):159–416, 1994.

[19] Alan Genz. Testing multidimensional integration routines. pages 81–94. Proceedings of international conference on tools, methods and languages for scientific and engineering computation, 1984.

[20] Alan Genz. A package for testing multiple integration subroutines. In P. Keast and G. Fairweather, editors, *Numerical Integration: Recent Developments, Software and Applications*, pages 337–340. Kluwer Academic, 1987.

[21] Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar. *Introduction to Parallel Computing, Second Edition*. Addison Wesley, 2003.

[22] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 1999.

[23] Seymour Haber. The tanh rule for numerical integration. *SIAM Journal on Numerical Analysis*, 14:668–685, 1977.

[24] Seymour Haber. Two formulas for numerical indefinte integration. *Math. Comp.*, 60:279–296, 1993.

[25] David H.Bailey and David Broadhurst. Parallel integer relation detection: Techniques and application. *Mathematics of Computation*, 70:1719–1736, 2000.

[26] S. Joe Ian H. Sloan. *Lattice Methods for Multiple Integration*. Oxford Science Publications, 1994.

[27] M. Iri, S. Moriguti, and Y. Takasawa. On a certain quadrature formula (in japanese). *Kokyuroku of Res. Inst. for Math. Sci., Kyoto University*, 91:82–118, 1970.

[28] Philip J.Davis and Philip Rabinowitz. *Methods of Numerical Integration, second Edition*. Academic Press, 1984.

[29] N.M Korobov. The approximate computation of multiply integrals. *Doklady Akademii Nauk SSSR*, 124:1207–1210, 1959.

[30] Rainer Kress. *Numerical analysis*. Springer, 1998.

[31] Arnold R. Krommer and Christoph W. Ueberhuber. *Numerical Integration on Advanced Computer Systems*. Springer-Verlag, 1994.

[32] Lawrence Livermore National Laboratory. Introduction to parallel computing. Online tutorial available at:
http://www.llnl.gov/computing/tutorials/parallel_comp/.

[33] Serge Lang. *Real and Functional analysis*. Springer-Verlag, 1993.

[34] Lee Margetts. *Parallel Finite Element Analysis*. PhD thesis, University of Manchester, 2002.

[35] M.Mori. Discovery of the double exponential transformation and its developments. *Publ. Res. Inst. Math. Sci.*, 41:897–935, 2005.

[36] M. Mori. The double exponential formula for numerical integration over the half infinite interval. *Numerical Mathematics(Singapore 1988), International Series of Numerical Mathematics*, 86:367–379, 1988.

[37] M. Muhammad and M. Mori. Double exponential formulas for numerical indefinte integration. *J. Comput. Appl. Math*, 161:431–448, 2003.

[38] T. Ooura. A double exponential formula for the fourier transforms. *Publ. Res. Inst. Math. Sci.*, 41:971–977, 2005.

[39] T. Ooura and M.Mori. The double exponential formula for oscillatory functions over the half infinite interval. *J. Comput. Appl. Math.*, 38:353–360, 1991.

[40] T. Ooura and M.Mori. A robust double exponential formula for fourier-type integrals. *J. Comput. Appl. Math.*, 112:229–241, 1999.

[41] Behrooz Parhami. *Introduction to Parallel Processing : Algorithms and Architectures*. Kluwer Academic Publishers, 1999.

[42] C. Schwartz. Numerical integration of analytic functions. *J. Computational Phys.*, 4:19–29, 1969.

[43] Ian H. Sloan and Philip J. Kachoyan. Lattice methods for multiply integration: Theory, error analysis and examples. *SIAM J. Numer. Anal.*, 24:116–128, 1987.

[44] F. Stenger. Integration formulae based on the trapezoidal formula. *J. Inst. Math. Appl.*, 12:103–114, 1973.

[45] F. Stenger. Numerical methods based on the whittaker cardinal, or sinc functions. *SIAM Rev.*, 23:165–224, 1981.

[46] F. Stenger. *Numerical Methods Based on Sinc and Analytic Functions*. Springer, Berlin, New York, 1993.

[47] F. Stenger. Summary of sinc numerical methods, numerical analysis in the 20th century, vol. i, approximation theory. *J. Comput. Appl. Math.*, pages 379–420, 2000.

[48] A.H. Stroud. *Approximate Calculation of Multiply Integral*. Prentice-Hall, Inc., 1971.

[49] M. Sugihara. Optimality of the double exponential formula -functional analysis approach. *Numer. Math.*, 75:379–395, 1997.

[50] M. Sugihara. Near optimality of the sinc approximation. *Math. Comp.*, 72:767–786, 2003.

[51] M. Sugihara and T. Matsuo. Recent developments of the sinc numerical methods. *J. Comput. Appl. Math.*, pages 673–689, 2004.

[52] H. Takahasi. Complex function theory and numerical analysis (in japanese). Kokyuroku RIMS, Kyoto Univ., 1975.

[53] H. Takahasi and M. Mori. *Error estimation in the numerical integration of analytic functions*, 1970.

[54] H. Takahasi and M. Mori. Quadrature formulas obtained by variable transformation. *Numer. Math.*, 12:206–219, 1973.

[55] H. Takahasi and M. Mori. Double exponential formulas for numerical integration. *Publications of RIMS*, 9:721–741, 1974.

[56] Michael T.Heath. *Scientific Computing, an introductory survey*. McGraw-Hill, 1997.

[57] Yajun Yang. *Multidimensional Numerical Integration and Applications to Boundary Integral Equations*. PhD thesis, University of Iowa, 1993.