TERMINATION CRITERIA OF THE WANG-LANDAU ALGORITHM FOR MULTIDIMENSIONAL NUMERICAL INTEGRATION

BY

MISS JIRARAT CHAIRAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE (MATHEMATICS)
DEPARTMENT OF MATHEMATICS AND STATISTICS
FACULTY OF SCIENCE AND TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2015

COPYRIGHT OF THAMMASAT UNIVERSITY

TERMINATION CRITERIA OF THE WANG-LANDAU ALGORITHM FOR MULTIDIMENSIONAL NUMERICAL INTEGRATION

BY

MISS JIRARAT CHAIRAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE (MATHEMATICS)
DEPARTMENT OF MATHEMATICS AND STATISTICS
FACULTY OF SCIENCE AND TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2015

COPYRIGHT OF THAMMASAT UNIVERS

THAMMASAT UNIVERSITY
FACULTY OF SCIENCE AND TECHNOLOGY

THESIS

BY

MISS JIRARAT CHAIRAT

ENTITLED

TERMINATION CRITERIA OF THE WANG-LANDAU ALGORITHM FOR MULTIDIMENSIONAL NUMERICAL INTEGRATION

was approved as partial fulfillment of the requirements for
the degree of Master of Science (Mathematics)

on July 21, 2016

Chairman
(Assistant Professor Noparit Jinuntuya, Ph.D.)

Member and Advisor
Wanyok Atisattapong
(Assistant Professor Wanyok Atisattapong, Ph.D.)

Member
S. Kongnuan
(Assistant Professor Supachara Kongnuan, Ph.D.)

Member
(Saifon Chaturantabut, Ph.D.)

Dean
P. Sermsuk
(Associate Professor Pakorn Sermsuk)

Dissertation Title	TERMINATION CRITERIA OF THE WANG-LANDAU ALGORITHM FOR MULTIDIMENSIONAL NUMERICAL INTEGRATION
Author	Miss Jirarat Chairat
Degree	Master of Science (Mathematics)
Major Field/Faculty/University	Mathematics Faculty of Science and Technology Thammasat University
Dissertation Advisor	Asst. Prof. Wanyok Atisattapong, Ph.D.
Academic Years	2015

ABSTRACT

In this thesis, we propose new termination criteria for Wang-Landau sampling in multidimensional numerical integration. Instead of completing the simulations as in conventional Wang-Landau sampling, the process can be terminated earlier using the new criteria. The appropriate termination criteria are derived by checking absolute errors and the behavior of estimated integrals obtained from the Wang-Landau sampling. An algorithm with the new termination criteria was applied to approximate multidimensional integrals of up to six dimensions.

In addition, the new termination criteria were applied to the estimation of corner-peak integrals in 5, 8, and 10 dimensions. The corner-peak integrand is one of six test integrand families designed by Genz [10] and can be used to compare the efficiency of algorithms. We investigated the number of correct digits instead of the estimated integrals.

The accuracy, the errors of the numerical estimates obtained, and the CPU time of the conventional termination criterion and the new criteria were investigated. The results showed a significant reduction of the CPU time while the accuracy remained within acceptable limits.

Keywords: Monte Carlo method, Numerical integration, the Wang-Landau algorithm, Termination criterion

มหาวิทยาลัยธรรมศาสตร์ THAMMASAT UNIVERSITY

ACKNOWLEDGEMENTS

I would like to thank my advisor, Assistant Professor Dr. Wanyok Atisattapong, for all the education, invaluable advice, and encouragement throughout the Master's degree study. This thesis could not have been completed without her. She also encouraged and guided me through the programming process. Her guidance helped me throughout research and writing of this thesis. Moreover, I am grateful to Mr. John Winward for comments, and suggestions on the manuscript.

I am also thankful to the thesis committee members, Assistant Professor Dr. Noparit Jinuntuya, Assistant Professor Dr. Supachara Kongnuan, and Dr. Saifon Chaturantabut, for their encouragement, constructive comments, and helpful advice.

Special thanks to all teachers on the Bachelor and Master programs and the staff in the Department of Mathematics and Statistics, the Faculty of Science and Technology, Thammasat University.

I gratefully acknowledge the financial support provided by the Faculty of Science and Technology of Thammasat University throughout the Master program and for giving me financial support for my research.

I am most grateful to my parents who are the most important thing in my life and who support me in everything.

I would also like to thank all of friends for their help in everything. Furthermore, I have benefited greatly from Ms. Nuchakorn Ngamsaowaros's guidance in C programming.

Miss Jirarat Chairat

TABLE OF CONTENTS

Page

ABSTRACT (1)

ACKNOWLEDGEMENTS (3)

LIST OF TABLES (6)

LIST OF FIGURES (8)

CHAPTER 1 INTRODUCTION 1
- 1.1 Introduction 1
- 1.2 Motivation and Objectives 3
- 1.3 Outline of the thesis 4

CHAPTER 2 BACKGROUND AND LITERATURE REVIEWS 5
- 2.1 Background 5
  - 2.1.1 Numerical methods for integration 5
  - 2.1.2 The Density of States 14
- 2.2 Literature reviews 18

CHAPTER 3 METHODS 34
- 3.1 The Wang-Landau algorithm for numerical integration 34
- 3.2 Termination criteria for numerical integration 37
  - 3.2.1 The first termination criterion 38
  - 3.2.2 The second termination criterion 40

CHAPTER 4 NUMERICAL RESULTS 43
- 4.1 Multidimensional integrals 43
  - 4.1.1 The first termination criterion 44
  - 4.1.2 The second termination criterion 46
- 4.2 The corner-peak integral 50

CHAPTER 5 CONCLUSIONS 59

REFERENCES 62

BIOGRAPHY 66

THAMMASAT UNIVERSITY

LIST OF TABLES

Tables Page

2.1 (Left column) All possible configurations of spins for the 2 × 2 square lattice Ising model. (Right column) The total energy H of each configuration. 15

2.2 The relative error of the simulated mean values with respect to the exact values 27

2.3 (Left table) Five independent runs using the checking parameter defined by $T_c(C)$. (Right table) Five independent runs using the checking parameter defined by Q. Each procedure is the result of 10 finite-size scaling extrapolations for T_c, γ, and β for outputs at the end of f_{13} and for decreasing ε, using the 80% flatness criterion. 32

3.1 The mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional termination criterion and the first new termination criterion for I_π. 39

3.2 The mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional termination criterion and the second new termination criterion for I_π. 41

4.1 The average CPU time per run (seconds) and mean final order of the modification factor of the conventional termination criterion and first new termination criterion for I_{1D-6D}. 46

4.2 Numerical estimates of integrals obtained using the conventional and the first termination criteria. 46

4.3 The average CPU time per run (seconds) and mean final order of the modification factor of the conventional termination criterion and the second termination criterion for I_{1D-6D}. 47

4.4 Numerical estimates of integrals obtained using the conventional and the second termination criteria. 49
4.5 The percentage of the change in relative error and the percentage of reduction in CPU time for I_{1D-6D} using the new criteria compared with the conventional criterion. 50
4.6 The mean final order of the modification factor, the average CPU time per run (seconds), the mean of the number of correct digits, and the standard deviation obtained using the conventional termination criterion and the two new termination criteria for the corner-peak integral. 57

LIST OF FIGURES

Figures Page

2.1 Three types of Riemann sum: (a) the left-hand Riemann sum, (b) the right-hand Riemann sum, and (c) the midpoint Riemann sum. 6

2.2 The trapezoidal rule 7

2.3 Simpson's 1/3 rule 8

2.4 Simpson's 3/8 rule 9

2.5 Simple sampling 12

2.6 2×2 square lattice Ising model 15

2.7 An example of the determination of the proportion of the integral domain 17

2.8 Behavior of $T_c(C)$ during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion and beginning at f_{17}. The dots show where the modification factors were updated and the straight line is the exact value from [36]. 23

2.9 Best-fit Gaussians for the histograms of $T_c(C)$ during Wang-Landau sampling up to $\ln f = 10^{-4}$, using the 80%-flatness criterion, each for $100,000$ independent runs with the density of states being updated every p spin-flip trial. The central line is the exact value from [36]. 24

2.10 Evolution of the microcanonical average of the magnetization for the 2D Ising model for $L = 32$ at $E = -1024$ and -536 during the simulations over $1,000$ independent runs for each flatness stage. 24

2.11 Behavior of $T_c(\chi)$ during the WLS for eight independent runs using a flatness criterion of 80% and beginning at f_9. The dots show where the modification factor was updated and the straight line is the result obtained using the exact data from Ref. [36]. The density of states was updated after (a) every spin-flip and (b) every L^2 trial move with the microcanonical average accumulated from $\ln f = \ln f_7$. 25

2.12 Behavior of $T_c(\chi)$ during the WLS, beginning from f_9, for eight independent runs using a flatness criterion of 80% and a common microcanonical average in 24 independent runs. The density of states was updated after every L^2 trial move. The dots show where the modification factor was updated and the straight line is the result obtained using the exact data from Ref. [36] with the microcanonical average accumulated from $\ln f = \ln f_7$. 26
2.13 Best-fit Gaussians for the histograms of the temperatures of the peak of the specific heat for the 2D Ising model during the WLS up to $f_{final} = f_{13}$, using the 80%- and 90%-flatness criteria, each for $100,000$ independent runs. The $1/t$ simulations were carried out within the same CPU time. The central line corresponds to the exact temperature obtained with data from Ref. [36]. 27
2.14 (Upper panel) Evolution of the temperature of the maximum of the specific heat during the WLS, beginning from f_5 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε during the same simulation. 28
2.15 (Upper panel) Evolution of the temperature of the heat transfer per unit Q during the WLS, beginning from f_5 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε calculated by using Q during the same simulation. 30
2.16 (Upper panel) Mean final order of the modification factor for three levels of demand for the checking parameter, using $T_c(C)$ as a reference. (Lower panel) The same using the quantity Q for calculating the checking parameter. 31
2.17 (Upper panel) Evolution of the temperature of $T_c(C)$ during the WLS of the three-dimensional Ising model, beginning from f_8 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε during the same simulation. 32

2.18 Mean order of the final modification factor for each simulated size of homopolymers using the 80% flatness criterion. 33

3.1 An example of the density of states $g(y)$ of the integral 34

3.2 The exact density of states $g(y)$ of the integral I_π 35

3.3 Behavior of the estimated integral I_π during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion, $dy = 0.05$, and beginning at f_{10}. The dots show where the modification factors were updated and the straight line is the exact value. 38

3.4 (Upper panel) Evolution of the estimated integral I_π during the WLS, beginning from f_{10} for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the checking parameter ε during the same simulation. 41

3.5 Best-fit Gaussians for the histograms of the estimated integral I_π obtained using the conventional and two new termination criteria, using the 80%-flatness criterion, for 10000 independent runs. The central line is the exact value. 42

4.1 Behavior of the estimated integral during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion, $dy = 0.05$ for I_{1D-5D} and $dy = 0.1$ for I_{6D}. The dots show where the modification factors were updated and the straight line is the exact value. 45

4.2 (Upper panel) Evolution of the estimated integrals I_{1D-6D} during WLS for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the checking parameter ε during the same simulation. 48

4.3 Plots of the corner-peak integrand in one dimension for a given c. 51

4.4 A plot of the corner-peak integrand in two dimensions for a given $\mathbf{c} = (c_1, c_2)$. 52

4.5 $\bar{d}$ of the corner-peak integral in 5 dimensions. 55

4.6 $\bar{d}$ of the corner-peak integral in 8 dimensions. 56

4.7 $\bar{d}$ of the corner-peak integral in 10 dimensions. 56

CHAPTER 1

INTRODUCTION

1.1 Introduction

Integration is an important mathematical technique that can be applied to solve problems in many fields such as physics, biophysics, economics, and engineering. Definite integrals arise in such problems as finding the total area under a curve, finding the volume and surface area of a solid, finding the length of a plane curve, calculating the work done by a force, finding the center of gravity of a planar region, and finding the pressure and force exerted by a fluid on a submerged object [1]. Since there are many situations in which the evaluation of a definite integral cannot use analytical methods directly, numerical integration formulas are required in such situations.

A general principle for developing numerical integration rules is to replace the function f with simpler functions coinciding with f at selected grid points, such as polynomial interpolants, and then define the numerical integral as the exact integral of the interpolant function [2]. Simple numerical methods such as Riemann sums and the trapezoidal rule use a first-order polynomial (a line) to interpolate the function f. In order to obtain more accurate values, Simpson's rules use a higher-order polynomial for interpolation.

Although these numerical methods are efficient for approximating integrals in low dimensions, they lack efficiency in high dimensions. Therefore, Monte Carlo methods such as simple sampling, importance sampling and the Metropolis algorithm have been proposed for use in multidimensional integrals. However, these methods have limitations. For example, simple sampling fails to estimate ill-behaved integrands. It also suffers from slow convergence requiring a large amount of sampling to reduce the statistical error, and convergence is not always assured. Importance sampling may even converge to incorrect values if a bad weighting function is chosen, and such errors are not readily detected [3].

In 2001, Wang and Landau [4] presented a new Monte Carlo method called Wang-Landau sampling, and applied it to a discrete system in statistical physics, i.e. the Ising model. This algorithm is used to estimate the density of states $g(E)$. Some thermodynamic quantities in the Ising model can be more easily derived when the density of states is known. In 2007, Li et al. [3] applied the Wang-Landau algorithm to a continuous system in mathematics, i.e. the numerical integration. The algorithm is efficient for approximating multidimensional integrals and ill-behaved integrands. However, Wang-Landau sampling has the same limitation for both discrete and continuous systems: the saturation of error.

In order to obviate the restriction of the Wang-Landau algorithm, a new procedure (called the $1/t$ algorithm) was proposed by Belardinelli and Pereyra, and used to eliminate the saturation of error in the Ising model [5]. The $1/t$ algorithm was applied to numerical integration in 2008 by Belardinelli, Manzi, and Pereyra [6]. They found that the error saturation of the $1/t$ algorithm arises from the effect of discretization in the y space (the bin width effect), and results are close to the exact value at only small bin widths. Atisattapong and Maruphanton [7] improved the $1/t$ algorithm for approximating multidimensional integrations by introducing a new approximation of integration for large bin widths. Instead of using a fixed value of y to evaluate the integrals, they used the average of y values, which varies as the number of Monte Carlo trials changes. They applied the proposed method to multidimensional integrals up to six dimensions and ill-behaved integrals up to three dimensions. The numerical results suggested that the estimated integrals obtained by the proposed method always converge to the exact value and do not display error saturation at any bin width.

As well as improving the accuracy of the methods, attempts have been made to reduce the CPU time. In 2012, Caparica and Cunha-Netto [8] proposed a new termination criterion for Wang-Landau sampling in the Ising model. They found that the behavior of the microcanonical and canonical averages of some thermodynamic quantities in the Ising model stabilizes at some value before the process completes. It is not therefore necessary to perform the simulation to completion. Instead, a new termination criterion (or final modification factor) is determined. In 2014, Caparica [9] introduced another termination criterion for Wang-Landau sampling in Ising models. Instead of determining the new termination criterion in advance, as in Caparica and Cunha-Netto, he derived the appropriate termination criterion by checking absolute errors. As well as reducing the CPU time, this new termination criterion had the additional advantage that it is no longer necessary to define a final modification factor in advance, as different runs can stop at different values of the final modification.

No research has investigated the use of these two criteria in numerical integration. Therefore, this thesis will study Wang-Landau sampling for numerical integration in multidimensions using these two termination criteria to reduce the CPU time.

1.2 Motivation and Objectives

In this thesis, we study the behavior of estimated integrals obtained from Wang-Landau sampling using the termination criteria from Caparica and Cunha-Netto [8] and Caparica [9]. Instead of completing the simulations as in conventional Wang-Landau sampling, the process can be terminated earlier using the new criteria. To compare the efficiency of these criteria and of the conventional method, we apply the new criteria to multidimensional integrals (as in the works of Li et al.[3] and Belardinelli et al.[5]) up to six dimensions. In addition, we use the approximation proposed by Atisattapong and Maruphanton [7] to calculate the integrals. This is applied to a corner-peak integral in 5, 8, and 10 dimensions [10, 11, 12, 13, 14, 15, 16, 17, 18] to investigate the efficiency of the new termination criteria.

The objectives of this thesis are as follows:

1. To study the Wang-Landau algorithm for numerical integration in multidimensions with two termination criteria.

2. To code and implement a program for approximating multidimen-

sional integrals up to six dimensions and a corner-peak integral in high dimensions.

3. To compare the accuracy, the errors of the numerical estimates obtained, and the CPU time of the conventional termination criterion and the new criteria.
4. To derive an appropriate termination criterion of the Wang-Landau algorithm for approximating the integral in each dimension.

Our expectation of this work is a significant reduction of the CPU time while keeping the accuracy within acceptable limits.

1.3 Outline of the thesis

This thesis consists of five chapters. The first chapter provides an overview of the work and introduces our motivation and objectives. The second chapter introduces the background to the thesis and reviews the previous literature. The method is presented in Chapter 3. Chapter 4 gives the numerical results and compares the accuracy of the algorithms. Finally, a summary is given in Chapter 5.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEWS

2.1 Background

In this section, we introduce some background information. The section introduces two topics: numerical methods for integration and a definition of the density of states.

2.1.1 Numerical methods for integration

Riemann sums

Riemann sums are the simplest approximation of definite integrals. The concept of the Riemann sum is to divide the region under a curve $f(x)$ into rectangles of area $f(x)\Delta x$. The area under the curve $f(x)$ is then approximated by the sum of all areas $\sum f(x)\Delta x$.

The procedures are as follows:

1. The interval $[a, b]$ is divided into n subintervals $[x_i, x_{i+1}]$ of equal length $\Delta x = \frac{b-a}{n}$.
2. The function $f(x_i)$ is evaluated.
3. The integral, or the area under the curve $f(x)$, is calculated by

$$I = \sum_{i=0}^{n-1} f(x_i)\Delta x. \tag{2.1}$$

For this approximation, the left endpoint x_i of each subinterval is used to evaluate the function $f(x)$. This method is called the left-hand Riemann sum (Figure 2.1(a)). In addition, there are two more Riemann sums: the right-hand Riemann sum and the midpoint Riemann sum.

The right-hand Riemann sum (Figure 2.1(b)) uses the right endpoint x_{i+1} of each subinterval to evaluate the function $f(x)$. Then the integral is approximated by

$$I = \sum_{i=0}^{n-1} f(x_{i+1})\Delta x. \tag{2.2}$$

The midpoint Riemann sum (Figure 2.1(c)) uses the midpoint $\frac{x_i+x_{i+1}}{2}$ of each subinterval to evaluate the function $f(x)$. Then, the integral is approximated by

$$I = \sum_{i=0}^{n-1} f(\tfrac{x_i+x_{i+1}}{2})\Delta x. \tag{2.3}$$

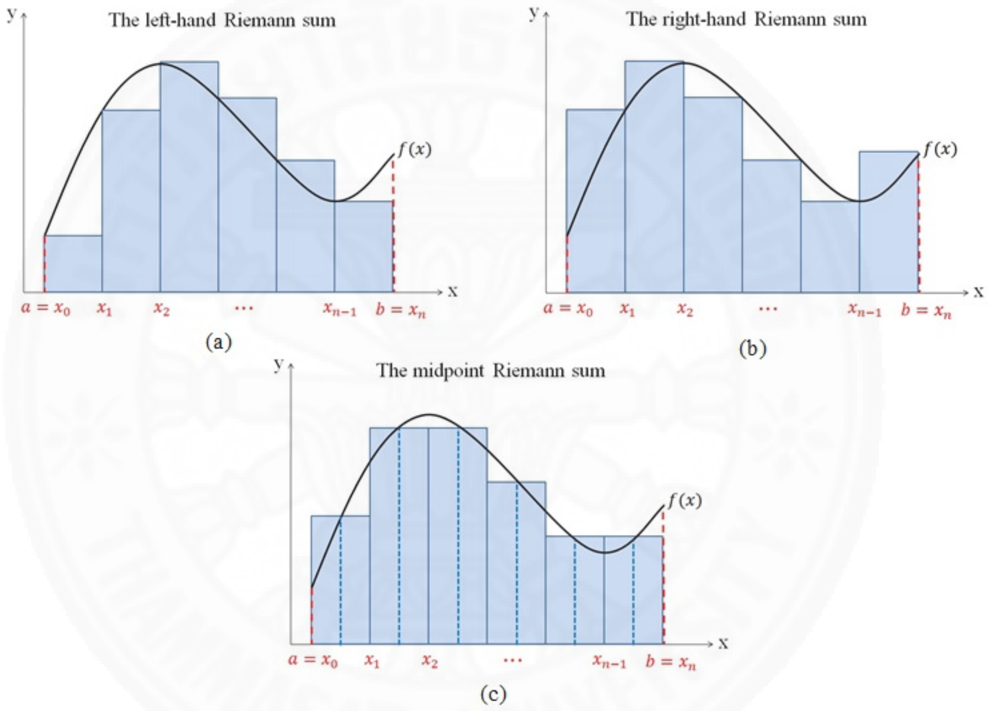


Figure 2.1: Three types of Riemann sum: (a) the left-hand Riemann sum, (b) the right-hand Riemann sum, and (c) the midpoint Riemann sum.

Next, we consider the error of numerical estimates for one-dimensional integrals using the Riemann sum. This depends on the number of subintervals and the degree of the interpolation polynomial. The error of the left-hand and right-hand approximations [19] with n subdivisions is in the order of $O(\Delta x) = \frac{1}{2}\frac{(b-a)^2}{n}f'(\xi)$, where f' is the first derivative of $f(x)$ and ξ lies somewhere in the interval $[a, b]$. The error of the midpoint Riemann sum approximation [19] with n subdivisions is in the order of $O((\Delta x)^2) = \frac{1}{24}\frac{(b-a)^3}{n^2}f''(\xi)$, where f'' is the second derivative of $f(x)$ at some $\xi \in [a, b]$.

Trapezoidal rule

The concept of the trapezoidal rule for approximating one-dimensional integrals is similar to that of the Riemann sum but the region under the curve $f(x)$ is divided into trapezoids instead of rectangles whose area is given by $\frac{1}{2}[f(x_{i+1})+f(x_i)]\Delta x$. The integral can then be approximated by summing the areas (Figure 2.2), as follows:

$$I=\sum_{i=0}^{n-1}\tfrac{1}{2}[f(x_{i+1})+f(x_i)]\Delta x=[\tfrac{1}{2}f(x_0)+\sum_{i=1}^{n-1}f(x_i)+\tfrac{1}{2}f(x_n)]\Delta x. \qquad (2.4)$$

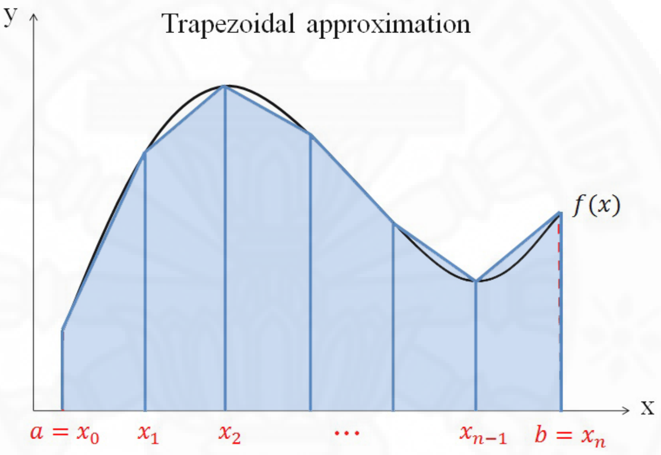


Figure 2.2: The trapezoidal rule

The error of the trapezoidal approximation [20] for one-dimensional integrals is in the order of $O((\Delta x)^2)=-\frac{1}{12}\frac{(b-a)^3}{n^2}f''(\xi)$.

Simpson's rule

Simpson's rule is another numerical method for approximating definite integrals. Both the Riemann sum and trapezoid rule approximate the curve $f(x)$ by a straight line while Simpson's rule approximates it using a higher order polynomial. Simpson's rule can be separated into two methods depending on the degree of interpolating polynomials: Simpson's 1/3 and Simpson's 3/8 methods.

Simpson's 1/3 rule

For Simpson's 1/3 rule, the interpolating polynomial is quadratic. Since this polynomial connecting three points (x_{i-1}, x_i and x_{i+1}) always covers two subintervals, it requires the number of subintervals (n) to be even. In particular, the region under the curve $f(x)$ is divided into the area $\frac{\Delta x}{3}[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})]$ for each pair of subintervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$. If the number of subintervals is odd, we cannot use Simpson's 1/3 rule to estimate the integral.

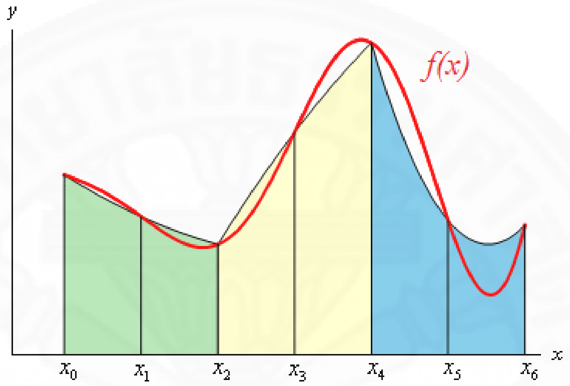


Figure 2.3: Simpson's 1/3 rule

For example, Figure 2.3 shows the region under the curve $f(x)$ divided by Simpson's 1/3 rule. The green, yellow, and blue areas are equal to $\frac{\Delta x}{3}[f(x_0) + 4f(x_1) + f(x_2)]$, $\frac{\Delta x}{3}[f(x_2) + 4f(x_3) + f(x_4)]$ and $\frac{\Delta x}{3}[f(x_4) + 4f(x_5) + f(x_6)]$, respectively. The integral can then be calculated by the sum of these areas:

$$I = \frac{\Delta x}{3}[f(x_0)+4f(x_1)+f(x_2)]+\frac{\Delta x}{3}[f(x_2)+4f(x_3)+f(x_4)]+\frac{\Delta x}{3}[f(x_4)+4f(x_5)+f(x_6)]. \tag{2.5}$$

In general, the composite Simpson's 1/3 rule for n subintervals is

$$\begin{aligned} I &= \tfrac{\Delta x}{3}[f(x_0) + 4f(x_1) + f(x_2)] + \tfrac{\Delta x}{3}[f(x_2) + 4f(x_3) + f(x_4)] \\ &\quad + \cdots + \tfrac{\Delta x}{3}[f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \\ &= \frac{\Delta x}{3}\sum_{i=1}^{n-1}[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})] \end{aligned} \tag{2.6}$$

The error of the composite Simpson's 1/3 rule [20] is in the order of $O((\Delta x)^4) = -\frac{1}{180}\frac{(b-a)^5}{n^4}f^{(4)}(\xi)$, where $f^{(4)}$ is the fourth derivative of $f(x)$ at some $\xi \in [a, b]$.

Simpson's 3/8 rule

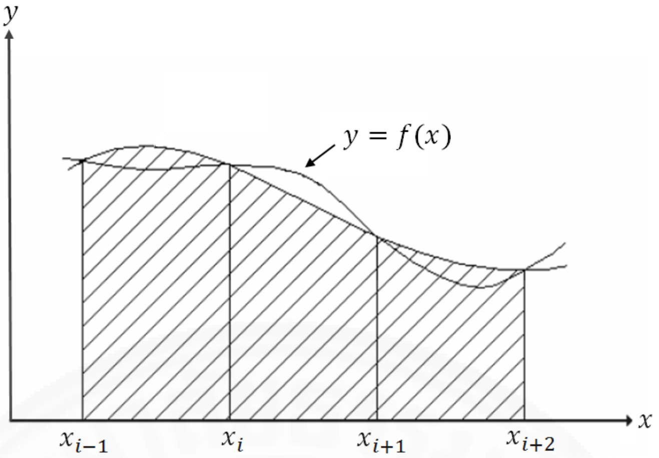


Figure 2.4: Simpson's 3/8 rule

For Simpson's 3/8 rule, the interpolating polynomial is cubic. Since this polynomial connecting four points (x_{i-1}, x_i, x_{i+1} and x_{i+2}) always covers three subintervals, it requires the number of subintervals (n) to be a multiple of three. In particular, the region under the curve $f(x)$ is divided into the area $\frac{3(\Delta x)}{8}[f(x_{i-1}) + 3f(x_i) + 3f(x_{i+1}) + f(x_{i+2})]$ for each of the three subintervals $[x_{i-1}, x_i]$, $[x_i, x_{i+1}]$, and $[x_{i+1}, x_{i+2}]$, as shown in Figure 2.4.

In general, the composite Simpson's 3/8 rule for n subintervals is

$$I = \tfrac{3(\Delta x)}{8} \sum_{i=1}^{n-2} [f(x_{i-1}) + 3f(x_i) + 3f(x_{i+1}) + f(x_{i+2})]. \tag{2.7}$$

The error of Simpson's 3/8 rule [20] with n subdivisions is in the order of $O((\Delta x)^4) = -\frac{1}{80}\frac{(b-a)^5}{n^4} f^{(4)}(\xi)$, where $f^{(4)}$ is the fourth derivative of $f(x)$ at some $\xi \in [a, b]$.

Numerical integration of multidimensional integrals

The methods mentioned above can approximate not only one-dimensional integrals, but also multidimensional integrals. For example, the two-dimensional integral is defined by

$$I = \int_c^d \int_a^b f(x, y) dx dy. \tag{2.8}$$

Since the order of the integration is not important, the integral can be computed as iterated integrals by

$$\begin{aligned} I &= \int_c^d \left(\int_a^b f(x,y)dx \right) dy \\ &= \int_a^b \left(\int_c^d f(x,y)dy \right) dx. \end{aligned} \tag{2.9}$$

First, we evaluate the integral in one dimension (only the integral in the bracket) using the methods mentioned above. Then, the result of this first integration becomes the integrand in the second dimension. For example, the integral can be estimated using the right-hand Riemann sum as follows:

$$\begin{aligned} I &= \int_c^d \left(\int_a^b f(x,y)dx \right) dy \\ &= \sum_{j=1}^{n_y} \left(\sum_{i=1}^{n_x} f(x_i, y_j)\Delta x \right) \Delta y, \end{aligned} \tag{2.10}$$

where n_x and n_y are the number of subintervals along the x-axis and y-axis. Integration in high dimensions is carried out in the same way.

Next, we consider the error of numerical estimates for multidimensional integrals. In one dimension, the errors of the Riemann sum, trapezoidal rule, and Simpson's rule are proportional to n^{-1}, n^{-2}, and n^{-4}. In two dimensions, these errors are proportional to $n^{-1/2}, n^{-1}$, and n^{-2}, respectively [20]. Generally, the errors in d dimensions are proportional to $n^{-a/d}$, where $a = 1, 2$, and 4 for the Riemann sum, trapezoidal rule, and Simpson's rule, respectively [20]. We can observe that the errors of these numerical integrations depend on the dimensionality, making it difficult to estimate the multidimensional integral using these methods. However, the Monte Carlo method is a numerical method for solving this problem which is efficient for higher-dimensional integrals and makes the error independent of the dimensionality of the integral.

Monte Carlo integration

Monte Carlo integration [21] is a simple numerical method for approximating integrals using random numbers, which is very useful for higher-dimensional and complex integrals. The Monte Carlo method uses randomly chosen points at which the integrand is evaluated. The main difference between Monte Carlo integration and the numerical methods discussed above is that the former uses probabilistic techniques, but the latter use deterministic processes. We will introduce three main Monte Carlo integration methods: simple sampling, importance sampling, and the Metropolis algorithm.

Simple sampling

Simple sampling [21] is a conventional Monte Carlo method which is the simplest and most effective approach to estimating definite integrals. In one dimension, the integral can be approximated by the average of the rectangular areas ($(b-a)f(x_i)$, for $i = 1 \dots N$) as follows:

$$I_N = \frac{(b-a)}{N} \sum_{i=1}^{N} f(x_i), \tag{2.11}$$

where x_i is randomly chosen with the uniform probability density distribution on the interval $[a, b]$, N is the number of trials (called the Monte Carlo trials or Monte Carlo sweeps), and I_N is the numerical integration at N MC trials.

For example, Figure 2.5 shows the numerical integration of the function $y = f(x)$ using three Monte Carlo trials. The integral can be estimated by the average of the area of rectangles $(b-a)f(x_1), (b-a)f(x_2)$, and $(b-a)f(x_3)$, where x_1, x_2, and x_3 are randomly chosen with the uniform probability density distribution on the interval $[a, b]$. Therefore, the value of the integral in Figure 2.5 is equal to $I_3 = \frac{(b-a)}{3}[f(x_1) + f(x_2) + f(x_3)]$.

Next, we consider the simple sampling Monte Carlo method for the d-dimensional integration defined by

$$I_D = \int\int \cdots \int_{\Omega} f(x_1, x_2, ..., x_d) dx_1 dx_2 ... dx_d, \tag{2.12}$$

where Ω is the integration domain. The simple sampling Monte Carlo method for the d-dimensional integration use the same concept as the one-dimensional integration. The n-tuple $(x_1, x_2, ..., x_d)$ is randomly chosen from the integration

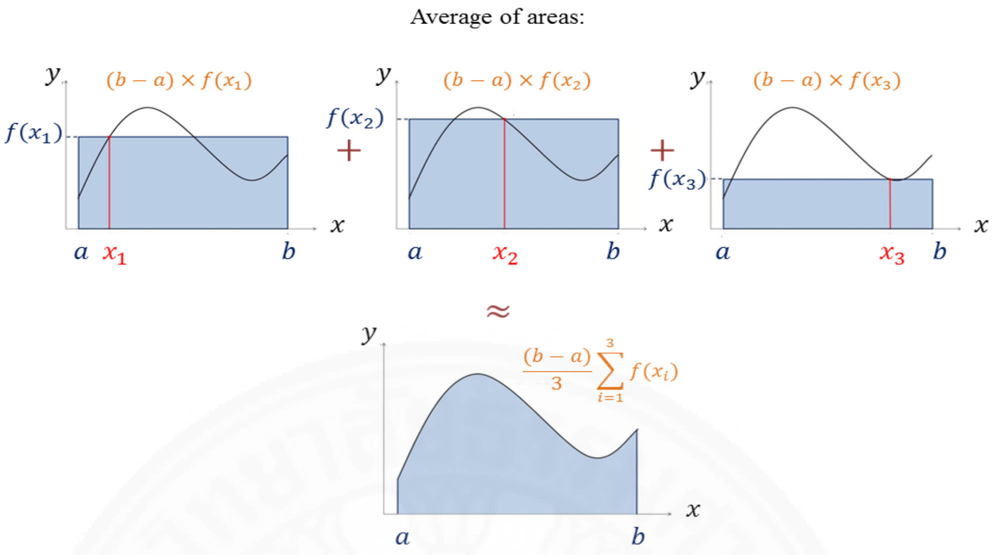


Figure 2.5: Simple sampling

domain Ω. Then, the d-dimensional integral can be approximated by

$$I_{D_N} = \frac{V}{N}\sum_{i=1}^{N} f(x_1, x_2, ..., x_d), \tag{2.13}$$

where V is the hypercube d-dimensional volume of the domain Ω.

The error of the simple sampling integration decreases as $1/\sqrt{N}$ [22] and is independent of the dimensionality of the integral. This demonstrates that the simple sampling integration is efficient in high dimensions. A variation on this method arises from the way of sampling the values of x which are randomly chosen with a uniform distribution. The advantages of this approach are ease of implementation and suitability for situations where little information is available. However, this method converges quite slowly for a function with very substantial variations over the range of interest. For example, it fails to apply on a sharply peaked integrand in very small regions (called the ill-behaved integrand). Ways of improving simple sampling are either to increase the number of trials or to reduce the variance of the integrand [23].

Importance sampling

Importance sampling [21, 23] is a method for improving simple sampling by reducing the variance of the integrand. In this method, instead of all points been randomly chosen with equal probability as in simple sampling, they are sampled according to a probability weighting function $p(x)$ where $p(x)$ is selected to be as close as possible to $f(x)$. Therefore, importance sampling increases the density of points in regions of interest. In this approach, the integral can be approximated by

$$I = \int_a^b \left[\frac{f(x)}{p(x)}\right] p(x)dx = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}, \tag{2.14}$$

where $p(x)$ is a positive function. The benefit of this method is that the error is in terms of the variance $Var(f(x)/p(x))$. This means that if $f(x)$ and $p(x)$ are close, the variance of $f(x)/p(x)$ is much smaller than the variance of $f(x)$. However, importance sampling can increase variance if a poorly weighting function $p(x)$ is selected. Moreover, there are limitations, i.e. $p(x)$ must be positive and normalized to unity in the integration domain [24].

The Metropolis algorithm

The Metropolis algorithm [25] is a subprocedure of importance sampling. This algorithm generates a random walk of points according to a desired probability distribution $p(x)$. The random walk is defined by determining a transition probability $T(x_i \to x_j)$ from point x_i to another point x_j. For the existence of the desired probability distribution $p(x)$, a sufficient (but not necessary) condition is the detailed balance:

$$p(x_i)T(x_i \to x_j) = p(x_j)T(x_j \to x_i). \tag{2.15}$$

The move is either accepted or rejected according to the Metropolis algorithm. The algorithm corresponds to choosing

$$T(x_i \to x_j) = \min\left[1, \frac{p(x_j)}{p(x_i)}\right]. \tag{2.16}$$

The procedures of the Metropolis algorithm applied to the integration are as follows:

1. x_i is chosen randomly from the interval $[a, b]$.
2. x_j is also chosen randomly from the interval $[a, b]$ and then $p(x_j)/p(x_i)$ is calculated.
3. If $T(x_i \to x_j) = 1$, the move is accepted. Then, let $x_{i+1} = x_j$.
4. If $T(x_i \to x_j) < 1$, a random number r is generated on $[0, 1]$.
    - If $r \leq T(x_i \to x_j)$, the move is accepted. Then, let $x_{i+1} = x_j$.
    - If $r > T(x_i \to x_j)$, the move is rejected. Then, let $x_{i+1} = x_i$.

However, the Metropolis algorithm cannot be applied to ill-behaved integrals. The random walk can be trapped on the peak because the transition probability at the peak is very low [24].

2.1.2 The Density of States

The density of states $g(E)$ is the number of states in each interval of energy of a system. For understanding the definition, we illustrate the density of states using a discrete system known as the Ising model. The Ising model, invented by the physicists Wilhelm Lenz and Ernst Ising [26], is a simple mathematical model in statistical physics that describes ferromagnetism. The model is based on the concept of interacting spins on an unchanging lattice L. For the two-dimensional Ising model, the arrangement of the spin i in each lattice site is either up or down. The values of the spin arrangements are denoted by δ_i. Let $\delta_i = 1$ if spin i is up and $\delta_i = -1$ if spin i is down. The Hamiltonian H (or the total energy of the system) is given by

$$H = -J \sum_{\langle ij \rangle} \delta_i \delta_j, \tag{2.17}$$

where J is called the exchange energy and $\langle ij \rangle$ is all nearest neighbor spins.

The density of states $g(E)$ in the Ising model represents the number of all possible states or configurations for an energy level E. For example, in the 2×2 square lattice Ising model (Figure 2.6), all possible configurations of spins

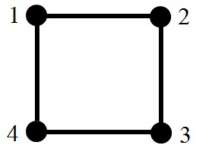


Figure 2.6: 2 × 2 square lattice Ising model

Table 2.1: (Left column) All possible configurations of spins for the 2 × 2 square lattice Ising model. (Right column) The total energy H of each configuration.

Configuration*	H
↑↑↑↑, ↓↓↓↓	$-4J$
↓↑↑↑, ↑↓↑↑, ↑↑↓↑, ↑↑↑↓, ↑↓↓↓, ↓↑↓↓, ↓↓↑↓, ↓↓↓↑, ↑↑↓↓, ↓↓↑↑, ↑↓↓↑, ↓↑↑↓	0
↓↑↓↑, ↑↓↑↓	$4J$

* ↑ denotes a spin up and ↓ denotes a spin down for 1st to 4th spin, respectively.

and the total energy H of each configuration are as in Table 2.1.

From Table 2.1, there are three possible energy levels: $E_1 = -4J, E_2 = 0$, and $E_3 = 4J$ and their densities of states are $g(E_1) = 2, g(E_2) = 12$, and $g(E_3) = 2$, respectively. The great advantage of the density of states is that the thermodynamic quantities can be calculated more easily and quickly.

Next, we introduce the partition function Z, which represents the statistical properties of a system in thermodynamic equilibrium, defined by

$$Z = \sum_{configurations} e^{-E/k_BT} = \sum_E g(E)e^{-E/k_BT}. \tag{2.18}$$

where T is the temperature and k_BT is the Boltzmann constant. If the density of states is not defined, the partition function is calculated by the first summation, which sums over all possible configurations. If the density of states is already known, the partition function can be calculated by the second summation, which sums only over all possible energy levels. In fact, the number of all possible energy levels is always less than the number of configurations.

Many important thermodynamic quantities [27] in statistical physics can be derived using the partition function. For example, the internal energy $U(T)$ (which is equivalent to the canonical average of the energy $\langle E\rangle_T$), heat capacity $C(T)$, free energy $F(T)$, and entropy $S(T)$ at temperature T are given by

$$U(T) = \frac{\sum_E Eg(E)e^{-\beta E}}{\sum_E g(E)e^{-\beta E}} = \frac{\sum_E Eg(E)e^{-\beta E}}{Z} \equiv \langle E\rangle_T, \tag{2.19}$$

$$C(T) = \frac{\partial U(T)}{\partial T} = \frac{\langle E^2\rangle_T - \langle E\rangle_T^2}{T^2}, \tag{2.20}$$

$$F(T) = -k_B T \log(Z), \tag{2.21}$$

$$S(T) = \frac{U(T) - F(T)}{T}, \tag{2.22}$$

where $\beta = 1/k_B T$.

For understanding the definition of the density of states in a continuous system, we take the numerical integration as an example. The one-dimensional definite integral can be calculated by

$$I = \int_a^b f(x)dx, \tag{2.23}$$

where $f(x)$ is a continuous function on the closed interval $[a, b]$. The geometrical interpretation of the definite integral is the area between the graph of the function $f(x)$ and the x-axis over a closed interval $[a, b]$.

Since the definite integral is hard to calculate directly, even for one dimension, numerical methods are needed. For calculating the area under the curve of the function $f(x)$ using the density of states $g(y)$, grid discretization is required. Therefore, it is necessary to define the upper bound y_{max} and lower bound y_{min} of the integral in advance. The interval $[y_{min}, y_{max}]$ is divided into n subintervals of equal length dy. The density of states of integrals is the proportion of the integral domain that lies within an interval $[y, y + dy]$, which is given by

$$g(y) \equiv \{x | x \in [a, b], y \leqslant f(x) \leqslant y + dy\}, \tag{2.24}$$

where dy is the bin width of y. Then, the integral can be approximated by

$$I = \int_a^b f(x)dx \approx \sum_{y_{min}}^{y_{max}} g(y)y^*, \tag{2.25}$$

where y^* is the midpoint of the interval $[y, y + dy]$ defined by $y^* = \frac{y+(y+dy)}{2}$.

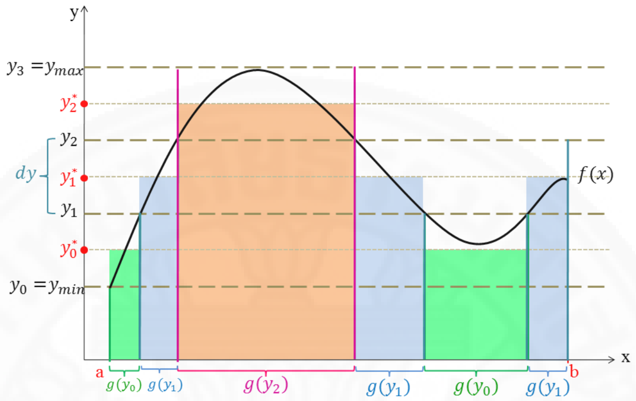


Figure 2.7: An example of the determination of the proportion of the integral domain

For example, we approximate the area under the curve $f(x)$, as shown in Figure 2.7. The upper bound y_{max} and lower bound y_{min} of the integral are defined in advance. The interval $[y_{min}, y_{max}]$ is divided into three subintervals of equal length dy.

Then, we can find the value of x by using the inverse function $x = f^{-1}(y)$. This is the reason that the integrand must be invertible. From the definition of the density of states of integrals as in Eq. (2.24), we can obtain the exact $g(y_0), g(y_1)$, and $g(y_2)$. Therefore, the area under the curve in Figure 2.7 can be approximated by

$$\int_a^b f(x)dx \approx \sum_{i=0}^{2} g(y_i)y_i^* = g(y_0)y_0^* + g(y_1)y_1^* + g(y_2)y_2^*, \tag{2.26}$$

where y_i^* is the midpoint of the interval $[y_i, y_i + dy]$ defined by $y_i^* = \frac{y_i+(y_i+dy)}{2}$ for $i = 0, 1, 2$. From Figure 2.7, we observe that the sum of all green areas is $g(y_0)y_0^*$, the sum of all blue areas is $g(y_1)y_1^*$, and the sum of all red areas is $g(y_2)y_2^*$.

The concept of the approximation of the definite d-dimensional integral given by Eq. (2.12) is the same as that in one dimension. The integration using the density of states requires the upper bound y_{max} and lower bound y_{min} of the integral in advance. The interval $[y_{min}, y_{max}]$ is divided into subintervals of equal length dy and then the value of x is evaluated using the inverse function. The density of states of the d-dimensional integrals is expressed as

$$g(y) \equiv \{(x_1, x_2, ..., x_d) | (x_1, x_2, ..., x_d) \in \Omega, y \leqslant f(x_1, x_2, ..., x_d) \leqslant y + dy\}. \quad (2.27)$$

This method is very useful for integration in high dimensions. We only divide the one-dimensional range of the integral, instead of dividing the multidimensional domain of the integral which is hard to calculate.

2.2 Literature reviews

Wang and Landau [4] introduced a new procedure for the Monte Carlo algorithm, called Wang-Landau sampling. The algorithm can estimate the density of states $g(E)$ for calculating some thermodynamic quantities in the Ising model. The great advantage of the algorithm is that the free energy and the entropy can be obtained at any temperature, instead of by direct approximation using conventional Monte Carlo methods at a given temperature. This new method is also efficient for finding the 1st order and 2nd order phase transitions, and is convenient for complicated systems with rough energy landscapes.

The Wang-Landau algorithm is as follows:

1. The density of states is *a priori* unknown. We set $g(E) = 1$ and $H(E) = 0$ for all energy levels E, where $g(E)$ is the density of states and $H(E)$ is the histogram at energy level E.
2. A random walk starts at any energy level E defined by E_{old}. After that, random walks are performed in energy space by choosing a random spin-flip. After flipping the spin, the new energy level defined by E_{new} is calculated.

3. The transition probability from the old energy level (E_{old}) to the new energy level (E_{new}) is
$$p(E_{old} \to E_{new}) = \min\left(\frac{g(E_{old})}{g(E_{new})}, 1\right). \qquad (2.28)$$

4. If the random walk is accepted, the density of states and the histogram at the new energy level (E_{new}) are updated by $g(E_{new}) \to g(E_{new}) \times f$ and $H(E_{new}) \to H(E_{new}) + 1$, where f is a modification factor. The initial modification factor is $f_0 = e_1 \backsimeq 2.71828\ldots$
   Otherwise, the density of states and the histogram at the old energy level (E_{old}) are updated by $g(E_{old}) \to g(E_{old}) \times f$ and $H(E_{old}) \to H(E_{old}) + 1$.

5. The random walks are performed repeatedly until the accumulated histogram $H(E)$ is flat. This means that for a given flatness criterion p, $H(E) > p \times \langle H(E) \rangle$ where $\langle H(E) \rangle$ is the average histogram of all energy levels. After that, the modification factor is refined by $f \to \sqrt{f}$ and the histogram is reset to $H(E) = 0$ for all energy levels.

6. The simulation continues until f reaches the final modification factor $f_{final} = \exp(10^{-8}) \backsimeq 1.00000001$.

Moreover, Wang and Landau suggested that the algorithm can be used to study complex systems such as spin glass models [28, 29, 30] and protein folding problems [31] that present challenges when using other optimization algorithms, because the energy landscapes are very rough. The Wang-Landau algorithm has been successfully applied in many fields including statistical physics [32, 33], biophysics [34], and mathematics [35].

Tröster and Dellago [35] presented Wang-Landau sampling with a self-adaptive range. This method is useful for estimating the density of states in systems in which the boundary of energy is unknown in advance. They applied the method to the two-dimensional Ising model to provide upper and lower bounds of energy. This allows the number of energy levels and values at each energy level during simulation to be calculated. Moreover, it is efficient for calculating the integrals of sharply peaked functions in high dimensions. However, in their approach, the integrand $f(x)$ was expressed in terms of a Boltzmann factor $e^{-\phi(x)}$ with $\phi(x) = -\ln[f(x)]$. The method has the restriction that the integrand $f(x)$ must be positive.

Li et al. [3] proposed a new application of Wang-Landau sampling to numerical integration. They illustrated the numerical results for the two following integrals:

$$I_{1D} = \int_{-2}^{2} (x^5 - 4x^3 + x^2 - x) \sin(4x) dx, \tag{2.29}$$

$$I_{2D} = \int_{-1}^{1} \int_{-1}^{1} (x_1^6 - x_1 x_2^3 + x_1^2 x_2 + 2x_1) \sin(4x_1 + 1) \cos(4x_2) dx_1 dx_2. \tag{2.30}$$

These functions are exactly integrable and feature multiple maxima and minima in one- and two-dimensions. Li et al. compared the estimated values of integrals with those derived by the simple sampling method. The results indicated that both methods provide the same convergence behavior in low dimensions. The self-adaptive method cannot estimate negative integrands, but Wang-Landau sampling does not have this restriction.

Furthermore, the method can be applied to estimate the ground state energy in the lattice Anderson model, which can be decomposed into a non-interacting value E_0 plus correction term $(E_2, E_4, \ldots)$. Each correction term is in terms of multidimensional integrals of the summation of the products of the Green's function. The results show that the estimated value converges to the exact value for smaller bin widths dy and larger flatness criteria p. Therefore, the accuracy of the numerical estimate depends on the two adjustable parameters: the flatness criterion p and the bin width dy.

Belardinelli and Pereyra [5] proposed a new procedure, called the $1/t$ algorithm, to avoid the restrictions of the Wang-Landau algorithm. They found that the main problem of Wang-Landau sampling arises from the saturation of the error. This is caused by the way of refining the modification factor and the condition for the histogram flatness. The relations $S(E) = \ln[g(E)]$ and $F = \ln(f)$ are used for fitting all values of $g(E)$ into double precision numbers.

The $1/t$ algorithm is as follows:

1. Initially, set $S(E) = \ln[g(E)] = 0$ and set $H(E) = 0$ for all energy levels E.
2. Set $F_0 = \ln(f_0) = 1$ and $F_{final} = \ln(f_{final}) = \ln[\exp(10^{-8})] = 10^{-8}$.
3. A random walk starts at any energy level E defined by E_{old}. Random walks are performed in energy space by randomly choosing a spin-flip. A new energy value defined by E_{new} is then calculated.
4. The transition probability from the old energy level E_{old} to the new energy level E_{new} is

   $$p(E_{old} \to E_{new}) = \min\{1, \exp[S(E_{old}) - S(E_{new})]\}.$$

5. If the random walk is accepted, the density of states and the histogram at the new energy level (E_{new}) are updated by $S(E_{new}) \to S(E_{new}) + F_k$ and $H(E_{new}) \to H(E_{new}) + 1$. Otherwise, the density of states and the histogram at the old energy level (E_{old}) are updated by $S(E_{old}) \to S(E_{old}) + F_k$ and $H(E_{old}) \to H(E_{old}) + 1$.
6. The histogram is checked after some Monte Carlo sweeps (e.g. 1000) with the condition $H(E) \neq 0$ for all energy levels E.
7. When all the energy levels E have been visited by the random walker at least once, the modification factor is refined using the following condition:

   $$F_{k+1} = \begin{cases} F_k/2, & F_k > 1/t; \\ 1/t, & F_k \leqslant 1/t \end{cases} \tag{2.31}$$

   and the histogram is reset to $H(E) = 0$ for all energy levels E.

8. The process is repeated up to $F_{k+1} = F_{final}$.

In this new procedure, a flatness criterion is not necessary for the simulation. Refining the modification factor is now based on the number of Monte Carlo trials, not on the number of iterations (k). Belardinelli and Pereyra used this method to calculate the density of states of the Ising model in a two-dimensional square lattice. The results converge to the exact value without the saturation of error, and are more accurate than the Wang-Landau algorithm.

Belardinelli, Manzi, and Pereyra [6] reported the application of the $1/t$ and Wang-Landau algorithms to multidimensional integrations. Both algorithms were compared with the simple sampling algorithm. They found that the error of the simple sampling algorithm decreases as $N^{-1/2}$, where N is the number of Monte Carlo trials, while the error of the Wang-Landau algorithm saturates. Although the error of the $1/t$ algorithm also saturates and does not converge to I_{exact}, it converges to $I(dy)_{ex} = \int_a^b f(x)dx \approx \sum\limits_{y_{min}}^{y_{max}} g_{ex}(y)y^*$, where $g_{ex}(y) \equiv \{x|x \in [a,b], y \leqslant f(x) \leqslant y+dy\}$ for a given bin width dy, and y^* denotes the midpoint of the interval $[y, y+dy]$. The results show that the $1/t$ algorithm is more accurate than the Wang-Landau algorithm in all dimensions. The accuracy of the $1/t$ algorithm is worse than the simple sampling algorithm in some low dimensions, but is significantly more accurate in high dimensions.

Caparica and Cunha-Netto [8] studied the behavior of the microcanonical and canonical averages of some thermodynamic quantities of the two-dimensional Ising model using Wang-Landau simulation. They found that the averages converge to their exact values before the process completes ($f_k \leq f_{final}$). This makes it unnecessary to perform the simulation to the end. They then determined a criterion for terminating the simulation f_{final} and a criterion for ceasing accumulation of the microcanonical averages f_{micro} from the behavior of the canonical and microcanonical averages during the simulation.

As an example, they studied the behavior of the temperature of the extremum of the specific heat $T_c(C)$. They found that the evolution of $T_c(C)$ in eight independent runs stabilized at some value before the end of the simulation (see Figure 2.8) [8].

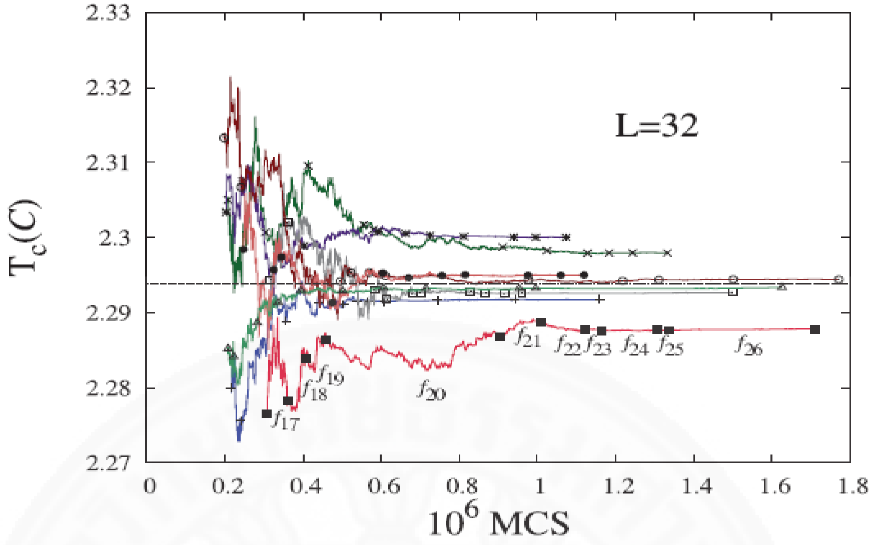


Figure 2.8: Behavior of $T_c(C)$ during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion and beginning at f_{17}. The dots show where the modification factors were updated and the straight line is the exact value from [36].

In Figure 2.8, $T_c(C)$ converged to the exact value at f_{23} and the process should be stopped. To improve the accuracy of Wang-Landau sampling, they also proposed a new way of updating the density of states. Instead of updating after every spin-flip trial, updating is done after some number (p) of Monte Carlo sweeps. Using $\ln f_{final} = 10^{-4}$ and updating the density of states only after L^2 trial moves produced more accurate results than other p trial moves, as shown in Figure 2.9 [8]. Note that $p = 1$ is equivalent to the conventional Wang-Landau algorithm.

Caparica and Cunha-Netto also studied the evolution of the microcanonical average of the magnetization $\langle m \rangle_E$ at $E = -1024$ and $E = -536$. Figure 2.10 [8] shows that $\langle m \rangle_E$ converged to a constant at f_7 for both energy levels making it unnecessary to accumulate $\langle m \rangle_E$ through the end of simulation, and allowing the CPU time to be reduced. After $\langle m \rangle_E$ is obtained, the temperature of the extremum of the susceptibility ($T_c(\chi)$) can be calculated by

$$\chi = L^2 \langle (m - \langle |m| \rangle)^2 \rangle / T. \tag{2.32}$$

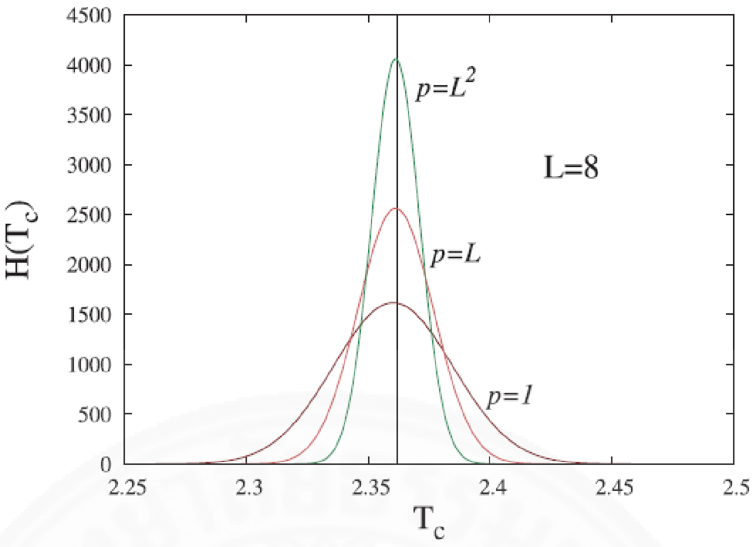


Figure 2.9: Best-fit Gaussians for the histograms of $T_c(C)$ during Wang-Landau sampling up to $\ln f = 10^{-4}$, using the 80%-flatness criterion, each for $100,000$ independent runs with the density of states being updated every p spin-flip trial. The central line is the exact value from [36].

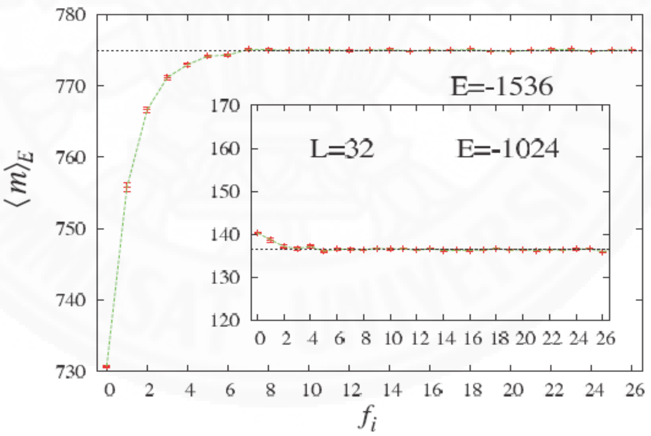
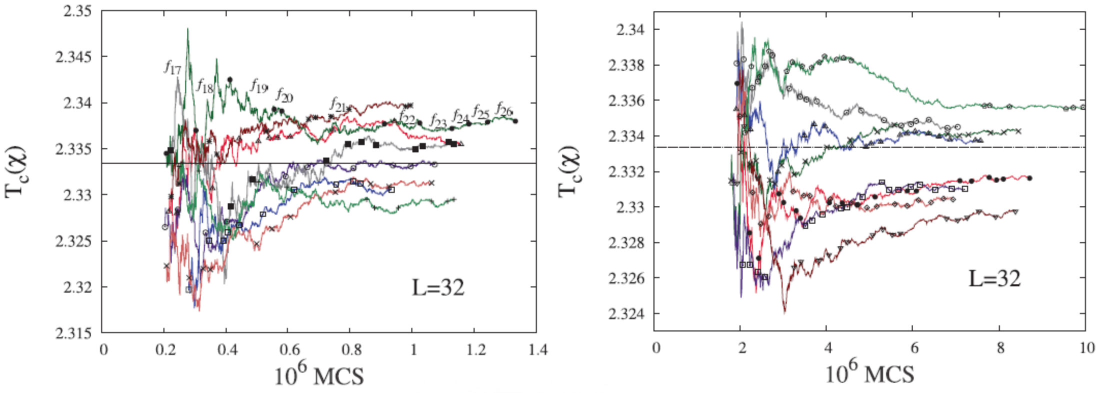


Figure 2.10: Evolution of the microcanonical average of the magnetization for the 2D Ising model for $L = 32$ at $E = -1024$ and -536 during the simulations over $1,000$ independent runs for each flatness stage.

Figs. 2.11(a) and 2.11(b) show that neither $T_c(\chi)$ obtained by the conventional Wang-Landau algorithm nor $T_c(\chi)$ obtained by the improved method (using $p = L^2$ and $f_{micro} = f_7$) flowed to steady values. However, $T_c(\chi)$ cal-



(a) (b)

Figure 2.11: Behavior of $T_c(\chi)$ during the WLS for eight independent runs using a flatness criterion of 80% and beginning at f_9. The dots show where the modification factor was updated and the straight line is the result obtained using the exact data from Ref. [36]. The density of states was updated after (a) every spin-flip and (b) every L^2 trial move with the microcanonical average accumulated from $\ln f = \ln f_7$.

culated using the mean values of $\langle m \rangle_E$ in 24 independent runs flowed to steady values, as shown in Figure 2.12 [8]. For calculating $T_c(\chi)$, the simulation should be terminated at $f_{final} = f_{13}$.

In brief, they proposed a new procedure for the Wang-Landau simulation which improved accuracy and reduced the CPU time. The new procedure involves the following steps:

1. Updating the density of states only after each Monte Carlo sweep instead of updating it after every spin-flip.

2. Performing the simulation until $f_k = f_{final}$, where f_{final} is determined by the behavior of the canonical averages during the simulation.

3. Accumulating the microcanonical average until $f_k = f_{micro}$, where f_{micro} is determined by the behavior of the microcanonical averages during the simulation.

Caparica and Cunha-Netto plotted the histogram of T_c $(H(T_c))$ in order

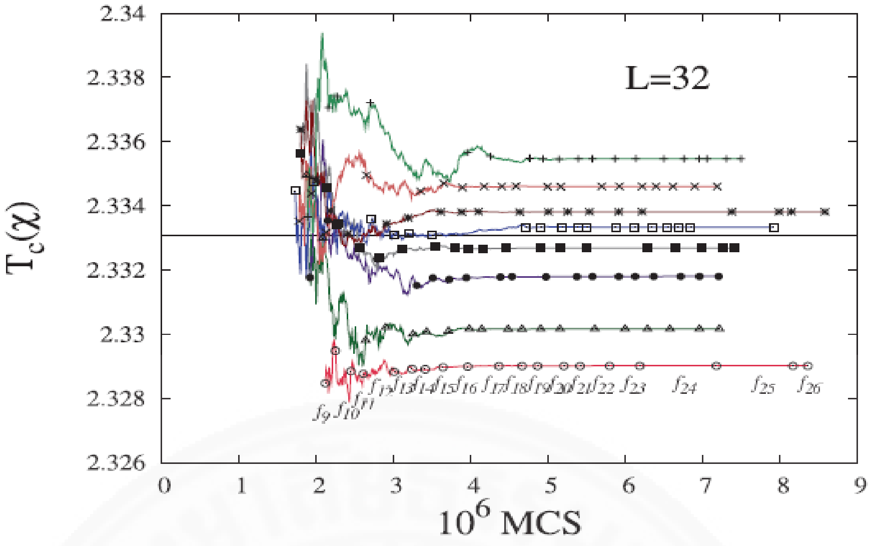


Figure 2.12: Behavior of $T_c(\chi)$ during the WLS, beginning from f_9, for eight independent runs using a flatness criterion of 80% and a common microcanonical average in 24 independent runs. The density of states was updated after every L^2 trial move. The dots show where the modification factor was updated and the straight line is the result obtained using the exact data from Ref. [36] with the microcanonical average accumulated from $\ln f = \ln f_7$.

to observe the peak of the histogram. Figure 2.13 shows that within the same CPU time, the peaks of $H(T_c)$ of the improved Wang-Landau method ($WL.f_{13}$) using 80%- and 90%-flatness criteria were lower than those of the $1/t$ algorithm using the same flatness criteria. Nevertheless, the centers of peaks of the $1/t$ algorithm were farther apart from the exact value than those of the $WL.f_{13}$ algorithm. This reveals a biased approximation effect in the $1/t$ algorithm.

Caparica and Cunha-Netto also calculated the relative errors of the simulated mean values with respect to the result using Ref. [36], as shown in Table 2.2. It is obvious that the relative errors of the improved method are less than those of the $1/t$ algorithm. The improved Wang-Landau method is therefore more efficient than the $1/t$ algorithm.

They then applied this idea to a self-avoiding homopolymer to calculate the energy and the mean-square end-to-end distance at some temperature. They

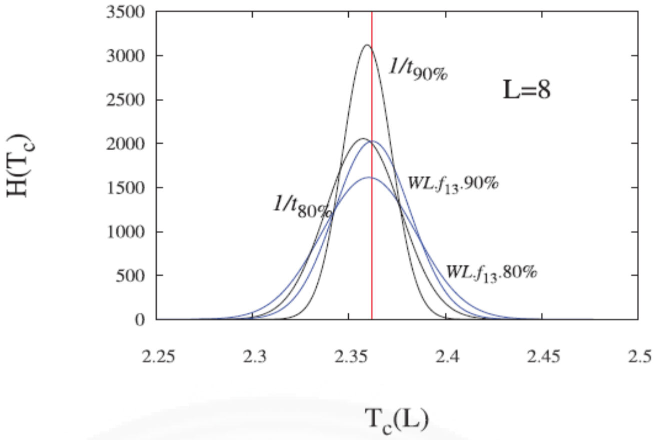


Figure 2.13: Best-fit Gaussians for the histograms of the temperatures of the peak of the specific heat for the 2D Ising model during the WLS up to $f_{final} = f_{13}$, using the 80%- and 90%-flatness criteria, each for $100,000$ independent runs. The $1/t$ simulations were carried out within the same CPU time. The central line corresponds to the exact temperature obtained with data from Ref. [36].

Table 2.2: The relative error of the simulated mean values with respect to the exact values

Simulation	Relative error
$WL.\ln.f_{13}.80\%$	0.00041
$WL.\ln.f_{13}.90\%$	0.00036
$1/t80\%$	0.00170
$1/t90\%$	0.00081

defined the proper f_{micro} and f_{final} of this model in the same way as the 2D Ising model.

Caparica [9] proposed a new halting criterion for Wang-Landau sampling in the two- and three-dimensional Ising models. The behavior of the critical temperature of the specific heat $T_c(C)$ and the susceptibility $T_c(\chi)$ were investigated during the Wang-Landau simulations. He suggested that the simulations should be terminated when $T_c(C)$ and $T_c(\chi)$ vary below a given threshold limit.

He also defined the checking parameter which is calculated by

$$\varepsilon = |T_c(t) - T_c(0)|, \tag{2.33}$$

where $T_c(0)$ is the last value obtained in the previous modification factor and $T_c(t)$ is the value obtained in the current Monte Carlo step. If ε during the interval $[f_k, f_{k+1}]$ is less than a predefined threshold limit value, then the simulation can be terminated at f_k. The proper threshold limits (e.g. $limit = 10^{-4}, 10^{-5}$, and 10^{-6}) are derived for ceasing the simulation in each model.

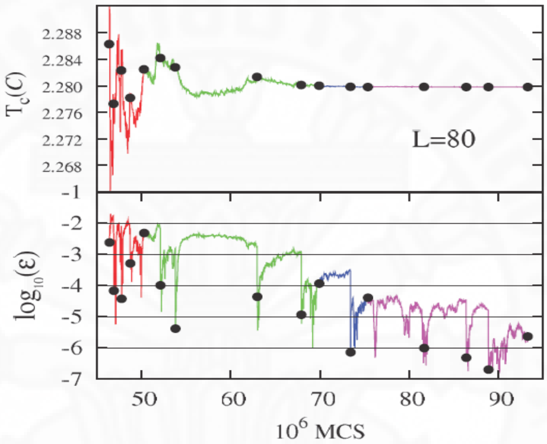


Figure 2.14: (Upper panel) Evolution of the temperature of the maximum of the specific heat during the WLS, beginning from f_5 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε during the same simulation.

For example, the upper panel in Figure 2.14 shows the behavior of $T_c(C)$ in a single run for $L = 80$ and the lower panel shows its evolution in terms of $\log_{10}(\varepsilon)$. Since the value of ε during the interval $[f_8, f_9]$ (the red line) is less than $limit = 10^{-2}$, the simulation can be terminated at f_8. Similarly, the simulation can be terminated at f_{13} (the green line), f_{15} (the blue line), and f_{19} (the purple line) for $limit = 10^{-3}, 10^{-4}$, and 10^{-5}, respectively.

In order to obtain a suitable *limit* for the Ising model, he performed manifold finite-size scaling simulations. According to finite-size scaling theory [37, 38, 39], the zero field scaling expressions for the magnetization and the susceptibility are given by

$$m \approx L^{-\beta/\nu} M(tL^{1/\nu}), \tag{2.34}$$

$$\chi \approx L^{\gamma/\nu} \chi(tL^{1/\nu}). \tag{2.35}$$

The locations of the maxima of these functions scale asymptotically as

$$T_c(L) \approx T_c + a_q L^{-1/\nu}, \tag{2.36}$$

where a_q is a quantity-dependent constant, allowing then the determination of T_c.

Using these scaling functions and assuming $\nu = 1$, the critical temperature T_c and the critical exponents β and γ can be calculated. He found that two independent finite-size scaling simulations provided very different results for the critical temperature. Then, he performed 10 independent Wang-Landau simulations for $L = 32, 36, 40, 44, 48, 52, 56, 64, 72$, and 80 with $N = 24, 24, 20, 20, 20, 16, 16, 16, 12$, and 12 independent runs for each size.

He compared the averages of 10 finite-size results for T_c, γ, and β by halting the simulations at $f_{final} = f_{13}$ (from Ref. [9]) and by halting the simulations at the given $limit = 10^{-3}, 10^{-4}$, and 10^{-5}. He found that the results were close to the exact value even for $limit = 10^{-3}$.

In the case of systems that exhibit more than one peak or behave unexpectedly, an alternative quantity, the heat transfer per unit Q, could be used. This value is defined by

$$Q = \frac{1}{N} \int_{T_i}^{T_f} C(T) dT, \tag{2.37}$$

where $C(T)$ is the specific heat and N is the number of units in the system (the number of spins, monomers, etc.).

Therefore, instead of checking the value of ε by Eq. (2.33), another checking parameter is used, defined by

$$\varepsilon = |Q(t) - Q(0)|, \tag{2.38}$$

where $Q(0)$ is the last value obtained in the previous modification factor and $Q(t)$ is the value obtained in the current Monte Carlo step.

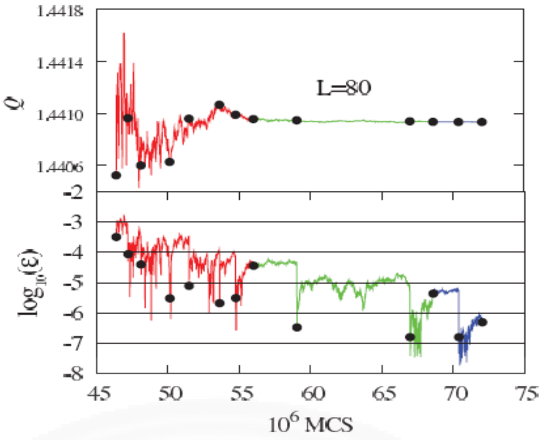


Figure 2.15: (Upper panel) Evolution of the temperature of the heat transfer per unit Q during the WLS, beginning from f_5 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε calculated by using Q during the same simulation.

For example, the upper panel in Figure 2.15 shows the behavior of Q in a single run for $L = 80$ and the lower panel shows its evolution in terms of $\log_{10}(\varepsilon)$. After f_{12}, the value of Q flows to a steady value and the value of error is also less than $limit = 10^{-4}, 10^{-5}$, and 10^{-6}. Thus, the halting criteria (Eq. (2.38)): $\varepsilon < 10^{-4}, \varepsilon < 10^{-5}$, and $\varepsilon < 10^{-6}$ are surveyed.

Next, Caparica considered the mean final orders of the modification factor (i_{final}) using the checking parameter defined by $T_c(C)$ (upper panel in Figure 2.16) and Q (lower panel in Figure 2.16). He found that i_{final} using Eq. (2.33) for $limit = 10^{-3}, 10^{-4}$, and 10^{-5} were roughly size independent while i_{final} using Eq. (2.38) for $limit = 10^{-4}, 10^{-5}$, and 10^{-6} decreased slightly as the lattice size increased.

Caparica then performed 50 finite-size scaling simulations to calculate T_c, γ, and β in order to find the proper limit for the two-dimensional Ising model. Table 2.3 shows the mean of 10 finite-size scaling extrapolations. The values in the left side use the checking parameter defined by $T_c(C)$ for $f_{final} = f_{13}$ and $limit = 10^{-3}, 10^{-4}$, and 10^{-5}. The results stabilized at $limit = 10^{-4}$ and the proper threshold limit was $limit = 10^{-4}$. The values in the right side in Table 2.3

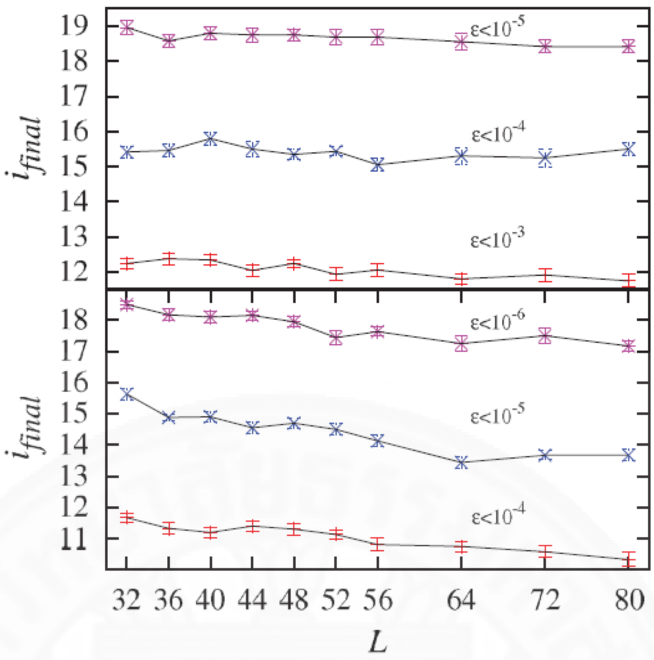


Figure 2.16: (Upper panel) Mean final order of the modification factor for three levels of demand for the checking parameter, using $T_c(C)$ as a reference. (Lower panel) The same using the quantity Q for calculating the checking parameter.

use the checking parameter defined by Q for $f_{final} = f_{13}$ and $limit = 10^{-4}, 10^{-5}$, and 10^{-6}. The results stabilized at $limit = 10^{-5}$ and the appropriate threshold limit was $limit = 10^{-5}$.

The advantages of the new halting criterion are that the CPU time is reduced, different runs can stop at different values of the final modification (f_{final}), and it is not necessary to define f_{final} in advance. The appropriate limit can be determined for all simulations and all sample sizes for the two-dimensional Ising model.

Caparica also applied new criterion for halting the simulations to the three-dimensional Ising model. The upper panel in Figure 2.17 shows the behavior of $T_c(C)$ beginning from f_8 in a single run for $L = 20$ and the lower panel shows its evolution in terms of $\log_{10}(\varepsilon)$. Caparica suggested that the simulation could be halted at f_{10}, f_{13}, and f_{18} for $limit = 10^{-3}, 10^{-4}$, and 10^{-5}, respectively.

He applied this criterion for ceasing the Wang-Landau simulation to

Table 2.3: (Left table) Five independent runs using the checking parameter defined by $T_c(C)$. (Right table) Five independent runs using the checking parameter defined by Q. Each procedure is the result of 10 finite-size scaling extrapolations for T_c, γ, and β for outputs at the end of f_{13} and for decreasing ε, using the 80% flatness criterion.

$T_c(C)$				Q			
f_{13}	$\varepsilon < 10^{-3}$	$\varepsilon < 10^{-4}$	$\varepsilon < 10^{-5}$	f_{13}	$\varepsilon < 10^{-4}$	$\varepsilon < 10^{-5}$	$\varepsilon < 10^{-6}$
$T_c = 2.2691853...$				$T_c = 2.2691853...$			
a. 2.26917(15)	2.26925(18)	2.26916(10)	2.26910(10)	a. 2.26917(15)	2.26942(25)	2.26917(12)	2.26911(10)
b. 2.26914(18)	2.26929(19)	2.26915(14)	2.26911(14)	b. 2.26914(18)	2.26944(27)	2.26913(17)	2.26911(14)
c. 2.26915(11)	2.26909(16)	2.26913(10)	2.26912(10)	c. 2.26915(11)	2.26921(16)	2.26918(11)	2.26914(11)
d. 2.26930(14)	2.26921(17)	2.26928(12)	2.26927(10)	d. 2.26930(14)	2.26936(21)	2.26928(13)	2.26927(11)
e. 2.26925(15)	2.26916(16)	2.26928(15)	2.26926(12)	e. 2.26925(15)	2.26959(20)	2.26926(15)	2.26925(13)
$\gamma = 1.75$				$\gamma = 1.75$			
a. 1.7609(12)	1.7600(11)	1.7605(10)	1.7605(10)	a. 1.7609(12)	1.7631(12)	1.7610(10)	1.7605(10)
b. 1.7586(15)	1.7579(19)	1.7587(11)	1.7589(10)	b. 1.7586(15)	1.7590(21)	1.7582(12)	1.7593(10)
c. 1.7597(15)	1.7601(18)	1.7584(11)	1.7580(10)	c. 1.7597(15)	1.7615(14)	1.7590(11)	1.7587(11)
d. 1.7568(16)	1.7576(18)	1.7571(13)	1.7573(10)	d. 1.7568(16)	1.7568(16)	1.7568(16)	1.7571(10)
e. 1.7590(11)	1.7590(15)	1.7577(10)	1.7576(10)	e. 1.7590(11)	1.7588(11)	1.7589(11)	1.7577(10)
$\beta = 0.125$				$\beta = 0.125$			
a. 0.12520(63)	0.12541(72)	0.12514(44)	0.12491(32)	a. 0.12520(63)	0.12570(12)	0.12517(49)	0.12496(36)
b. 0.12526(72)	0.12540(79)	0.12529(65)	0.12523(60)	b. 0.12526(72)	0.12560(97)	0.12525(68)	0.12520(60)
c. 0.12560(56)	0.12561(56)	0.12548(48)	0.12516(45)	c. 0.12559(56)	0.12588(79)	0.12550(52)	0.12522(45)
d. 0.12572(74)	0.12528(84)	0.12554(65)	0.12547(52)	d. 0.12572(74)	0.12619(85)	0.12565(67)	0.12543(57)
e. 0.12559(52)	0.12559(56)	0.12554(46)	0.12554(34)	e. 0.12559(52)	0.12655(71)	0.12557(50)	0.12549(40)

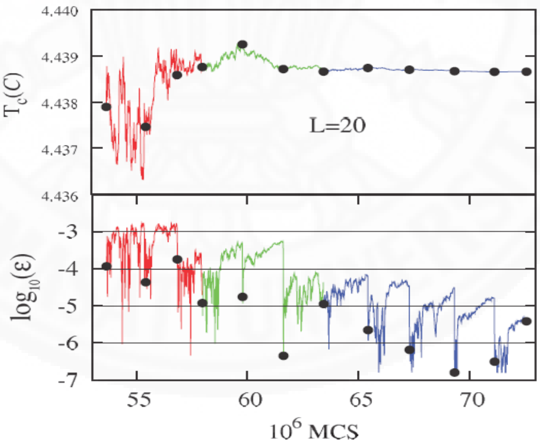


Figure 2.17: (Upper panel) Evolution of the temperature of $T_c(C)$ during the WLS of the three-dimensional Ising model, beginning from f_8 for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the logarithm of the checking parameter ε during the same simulation.

the self-avoiding homopolymer for $L = 90$. He estimated $T_c(C)$ by performing 10 independent runs for each size $N = 50, 70, \ldots, 150$ and using the finishing

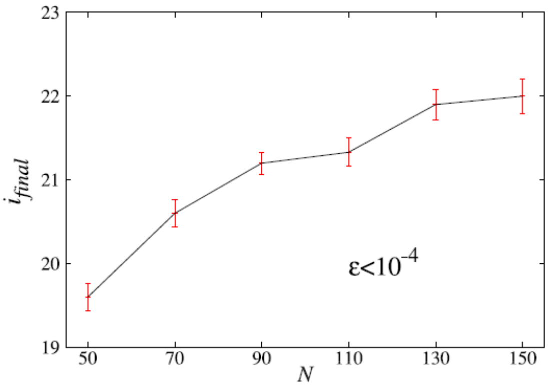


Figure 2.18: Mean order of the final modification factor for each simulated size of homopolymers using the 80% flatness criterion.

condition $|T_c(t) - T_c(0)| < 10^{-4}$. Figure 2.18 shows that i_{final} increased when the polymer size increased.

Atisattapong and Maruphanton [7] improved the $1/t$ algorithm for calculation of multidimensional integrations. The problem of the $1/t$ algorithm is the effect of discretization in the y space (the bin width effect). The result of the conventional $1/t$ algorithm is close to the exact value only for small bin widths. For large bin widths, the numerical estimates do not converge to the exact value and the errors saturate as a function of dy. To improve the method for large bin widths, they introduced a new approximation to estimate integrals using averages of y values in the subinterval $[y, y+dy]$ instead of using the midpoint $y^* = \frac{y+(y+dy)}{2}$. Then, Eq. (**??**) becomes

$$I = \int_a^b f(x)dx \approx \sum_{y_{min}}^{y_{max}} g(y)\bar{y}, \tag{2.39}$$

where $\bar{y} = \sum_{y^* \in [y,y+dy]} \frac{y^*}{n_y}$, and n_y is the number of sampling points y^* in an interval $[y, y+dy]$ for which $\sum_y n_y = N$, where N is the number of Monte Carlo trials. They proved the convergence of the new method and the non-convergence of the conventional method by theoretical analysis. This new approximation provides more accurate results. The bin width effect and the saturation of error for large bin widths are eliminated.

CHAPTER 3

METHODS

This chapter consists of two parts: the Wang-Landau algorithm for numerical integration and the termination criteria for numerical integration.

3.1 The Wang-Landau algorithm for numerical integration

The Wang-Landau algorithm is used in numerical integration to calculate the density of states $g(y)$. The density of states $g(y)$ is a direct analogy to the density of states $g(E)$ in a physical system. To understand the definition of $g(y)$, we take the numerical integration in one dimension as an example. The one-dimensional definite integral can be calculated by

$$I = \int_a^b f(x)dx, \tag{3.1}$$

where $f(x)$ is a continuous function on the closed interval $[a, b]$.

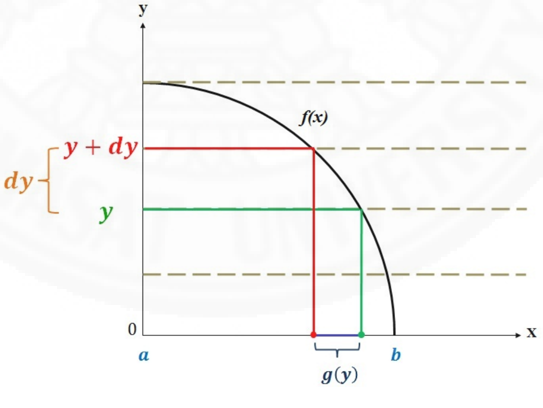


Figure 3.1: An example of the density of states $g(y)$ of the integral

Since the definite integral is hard to calculate directly, even for one dimension, numerical methods are needed. For calculating the area under the curve of the function $f(x)$ using the density of states $g(y)$, grid discretization is required. Therefore, it is necessary to define the upper bound y_{max} and lower bound y_{min} of the integral in advance. The interval $[y_{min}, y_{max}]$ is divided into

n subintervals of equal length dy. The density of states of integrals [3] is the proportion of the integral domain that lies within an interval $[y, y + dy]$ (Figure 3.1), which is given by

$$g(y) \equiv \{x | x \in [a, b], y \leqslant f(x) \leqslant y + dy\}, \tag{3.2}$$

where dy is the bin width of y. Then, the integral can be approximated by

$$I = \int_a^b f(x)dx \approx \sum_{y_{min}}^{y_{max}} g(y)\overline{y} = \sum_{i=0}^{n-1} g(y_i)\overline{y_i}, \tag{3.3}$$

where $\overline{y_i} = \sum_{y^* \in [y_i, y_i + dy]} \frac{y^*}{n_y}$ and n_y is the number of sampling points y^* in an interval $[y_i, y_i + dy]$ for which $\sum_y n_y = N$ where N is the number of Monte Carlo trials [7].

To better understand the definition of the density of states of integrals $g(y)$, we illustrate the approximation of the simple integral (I_π) by

$$I_\pi = 4\int_0^1 \sqrt{1 - x^2} dx. \tag{3.4}$$

where the integrand is the function of one-quarter of a circle, as shown in Figure 3.1. Thus, the integral multiplied by four is equal to the area of a unit circle. Therefore, the exact value of the integral I_π is $\pi = 3.14159265358979...$

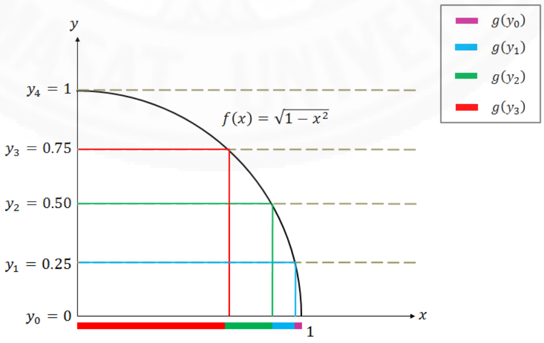


Figure 3.2: The exact density of states $g(y)$ of the integral I_π

Given the integration domain $x \in [0, 1]$, the integrand $f(x)$ in Eq. (3.4) ranges from $y_{min} = 0$ to $y_{max} = 1$. Suppose that we divide the interval $[y_{min}, y_{max}] = [0, 1]$ into four subintervals of equal length $dy = 0.25$. Since $y =$

$f(x) = \sqrt{1-x^2}$ is bijective, $f(x)$ is invertible. Then, we can find the inverse function as follows:

$$\begin{aligned} y = f(x) &= \sqrt{1-x^2} \\ y^2 &= 1-x^2 \\ x^2 &= 1-y^2 \\ x &= \sqrt{1-y^2} \end{aligned}$$

From the definition of the density of states of integrals defined in Eq. (3.2), we can obtain the exact $g(y_i)$ given by $g_{ex}(y_i) = \sqrt{1-y_i^2} - \sqrt{1-(y_i+dy)^2}$, which represents the width of the interval of x, as shown in Figure 3.2. The exact $g_{ex}(y_0), g_{ex}(y_1), g_{ex}(y_2)$, and $g_{ex}(y_3)$ are equal to 0.031754 (purple), 0.102220 (blue), 0.204588 (green), and 0.661438 (red). Then, the integral can be approximated by

$$\begin{aligned} \int_0^1 \sqrt{1-x^2}dx &\approx \sum_{i=0}^{3} g_{ex}(y_i)\overline{y_i} \\ &= g_{ex}(y_0)\overline{y_0} + g_{ex}(y_1)\overline{y_1} + g_{ex}(y_2)\overline{y_2} + g_{ex}(y_3)\overline{y_3} \\ &= 0.031754\overline{y_0} + 0.102220\overline{y_1} + 0.204588\overline{y_2} + 0.661438\overline{y_3}, \end{aligned}$$

where $\overline{y_i} = \sum_{y^* \in [y_i, y_{i+1}]} \frac{y^*}{n_y}$ and n_y is the number of sampling points y^* in an interval $[y_i, y_{i+1}]$ for which $\sum_y n_y = N$ where $y_0 = 0, y_1 = 0.25, y_2 = 0.5$, and $y_3 = 1$.

To generate the distribution $g(y)$ using the Wang-Landau algorithm, the interval $[y_{min}, y_{max}]$ is divided into n segments. The Wang-Landau algorithm is then used to calculate the numerical integration as follows.

1. At the beginning of the simulation, $g(y)$ is *a priori* unknown. We set $g(y) = 1$ and histogram $H(y) = 0$ for all values of y.
   Note that, since the density of states produces huge numbers, the relation $S(y) = \ln[g(y)]$ is used for fitting all values of $g(y)$ into double precision numbers; that is $S(y) = 0$.

2. A random walk starts at any point x in the integration domain defined by x_{old}, then $y_{old} = f(x_{old})$ is calculated.

3. After that, random walks are performed in y-space by randomly choosing another x and setting it to x_{new}. $y_{new} = f(x_{new})$ is then calculated.

4. The transition probability from y_{old} to y_{new} is accepted when

   $$p(y_{old} \to y_{new}) = \min\{1, \exp[S(y_{old}) - S(y_{new})]\}.$$

   If the random walk is accepted, the density of states and the histogram at y_{new} are updated by $S(y_{new}) \to S(y_{new}) + F$ and $H(y_{new}) \to H(y_{new}) + 1$, where $F = \ln(f)$ is a modification factor. The initial modification factor is $f_0 = e_1 \backsimeq 2.71828\ldots$.
   Otherwise, the density of states and the histogram at y_{old} are updated by $S(y_{old}) \to S(y_{old}) + F$ and $H(y_{old}) \to H(y_{old}) + 1$.

5. The random walks are performed repeatedly until the accumulated histogram $H(y)$ is flat. This means that for a given flatness criterion p, $H(y) > p \times \langle H(y) \rangle$, where $\langle H(y) \rangle$ is the average histogram of all values of y.

6. The modification factor is refined by $F \to F/2$ and the histogram is reset to $H(y) = 0$ for all values of y.

7. The distribution function must be normalized by

   $$g_{norm}(y) = \frac{(b-a)g(y)}{\sum_{y_{min}}^{y_{max}} g(y)}. \tag{3.5}$$

8. The simulation continues until f reaches the final modification factor $f_{final} = \exp(10^{-8}) \backsimeq 1.00000001$. Therefore, f_{final} of the conventional Wang-Landau simulation is f_{27}.

3.2 Termination criteria for numerical integration

In this section, we explain how two termination criteria can be applied to the Wang-Landau algorithm for numerical integration.

3.2.1 The first termination criterion

We adapted the termination criterion from Ref. [8] by investigating the behavior of integrals. We selected a new termination criterion (or f_{final}) at which the estimated integrals stabilized. To demonstrate how the criterion works, we consider the termination criterion of Wang-Landau sampling for the integral I_π.

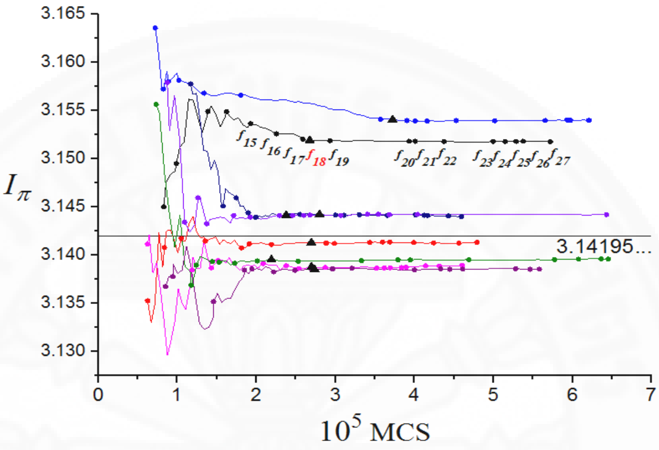


Figure 3.3: Behavior of the estimated integral I_π during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion, $dy = 0.05$, and beginning at f_{10}. The dots show where the modification factors were updated and the straight line is the exact value.

Figure 3.3 shows the evolution of the estimated integral I_π for eight independent runs using the 80%-flatness criterion and $dy = 0.05$. We can observe that around f_{18} (triangle points), all the curves stabilized at some value. This makes it unnecessary to perform the simulation to the end. Therefore, the appropriate termination criterion for I_π is $f_{final} = f_{18}$, which reduces the CPU time.

Since a single independent run of WLS is completed very quickly, it is difficult to demonstrate the efficiency of this termination criterion. We therefore performed 10000 independent runs using the same $f_{final} = f_{18}$.

In order to compare the accuracy of the integral, we computed the relative error in each run by

$$a_f = \left| \frac{I_{MC}(N) - I_{exact}}{I_{exact}} \right|, \tag{3.6}$$

where $I_{MC}(N)$ denotes the numerical estimate from the Monte Carlo method in the Nth Monte Carlo trial and I_{exact} denotes the exact value of the integral. After completing all independent runs, we compared the accuracy by considering the mean value $\bar{a}_f$, defined by

$$\bar{a}_f = \frac{1}{n} \sum_{i=1}^{n} a_{f_i}, \tag{3.7}$$

where a_{f_i} is the relative error of the integral from the ith run and n is the number of all independent runs. The standard deviation of the numerical estimates is given by

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (I_{MC_i}(N) - \bar{I})^2}, \tag{3.8}$$

where $\bar{I}$ is the average of all numerical estimates. The standard error of mean is defined by

$$\sigma_{error} = \frac{\sigma}{\sqrt{n}}. \tag{3.9}$$

Table 3.1: The mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional termination criterion and the first new termination criterion for I_π.

	Conventional criterion	**First new criterion**
f_{final}	27	18
CPU time	0.177(±0.223)	0.088(±0.057)
Estimates	$3.14206(\pm 5.672 \times 10^{-5})$	$3.14209(\pm 5.805 \times 10^{-5})$
Relative error $\bar{a}_f$	1.49877×10^{-4}	1.57813×10^{-4}

Table 3.1 compares the mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional termination criterion and the first new termination criterion for I_π. Note that the error of the average CPU

time per run (values in brackets) is the standard deviation, and the error of the numerical estimates is the standard error of mean calculated by Eq. (3.9).

As a result, the average CPU time per run of the conventional termination criterion was 0.177 seconds while that of the new criterion was only 0.088 seconds, a reduction of 50.28 percent. The relative error of the simulated mean values of the conventional criterion with respect to the exact value was 1.49877×10^{-4} while that of the new termination criterion was 1.57813×10^{-4}, an increase of only 0.00079 percent. This indicates that the new criterion can save significant CPU time while keeping the accuracy of the estimated value within acceptable limits.

3.2.2 The second termination criterion

We adapted the termination criterion from Ref. [9] by checking the relative errors given by

$$\varepsilon = \left| \frac{I(t) - I(0)}{I(0)} \right|, \tag{3.10}$$

where $I(0)$ is the value obtained from the previous modification factor and $I(t)$ is the value obtained in the current Monte Carlo step. The value of the estimated integral $I(0)$ can be preserved when the histogram is flat. If ε during the interval $[f_k, f_{k+1}]$ is less than a predefined threshold limit value (*limit*) then the simulation can be terminated at f_k, where k is the final order of the modification factor.

The upper panel in Figure 3.4 shows only the worst case (the uppermost line) of the evolution of the estimated integral I_π from Figure 3.3, beginning from f_{10} using the 80%-flatness criterion and $dy = 0.05$. The lower panel in Figure 3.4 shows the evolution of the checking relative error ε during the same simulation. We can observe that, after the estimated integral stabilizes (at f_{18}), all values of ε are less than 10^{-4}. Therefore, $limit = 10^{-4}$ can be selected as the appropriate threshold limit for halting the simulation of I_π. In this case, $f_{final} = f_{18}$, though different runs will terminate at different f_{final} values.

We then performed 10000 independent runs under the same conditions using the second termination criterion. Table 3.2 compares the mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional

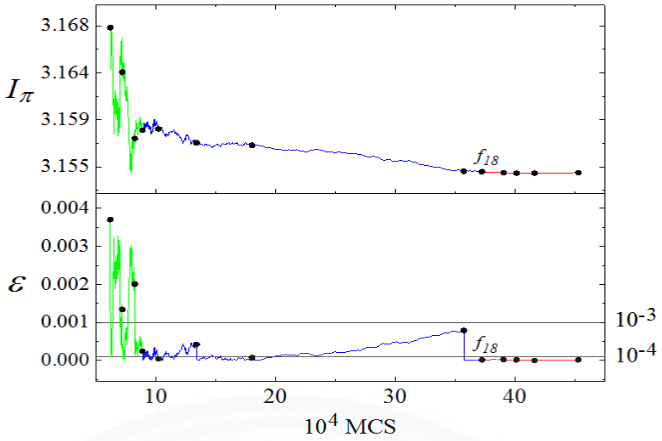


Figure 3.4: (Upper panel) Evolution of the estimated integral I_π during the WLS, beginning from f_{10} for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the checking parameter ε during the same simulation.

termination criterion and the second new termination criterion for I_π.

As shown in Table 3.2, the mean final order of the modification factor (k) for I_π dropped slightly to 17.0723. The average CPU time per run of the second termination criterion was 0.079 seconds, shorter than the first termination criterion by 10.23 percent. The relative error of the simulated mean values of the second termination criterion with respect to the exact value was 1.61863×10^{-4}, greater than the first termination criterion by 0.00041 percent. In brief, the

Table 3.2: The mean final order of the modification factor, the average CPU time per run (seconds), the average of the numerical estimates, and the relative error obtained using the conventional termination criterion and the second new termination criterion for I_π.

	Conventional criterion	**Second new criterion**
f_{final}	27	17.0723
CPU time	0.177(±0.223)	0.079(±0.047)
Estimates	$3.14206(\pm 5.672 \times 10^{-5})$	$3.14210(\pm 5.901 \times 10^{-5})$
Relative error $\bar{a}_f$	1.49877×10^{-4}	1.61863×10^{-4}

second termination criterion eliminated unnecessary iterations, while maintaining the accuracy of the estimated value within acceptable limits. The additional advantage of this termination criterion is that it does not require f_{final} to be specified in advance because it is determined in the course of the simulation.

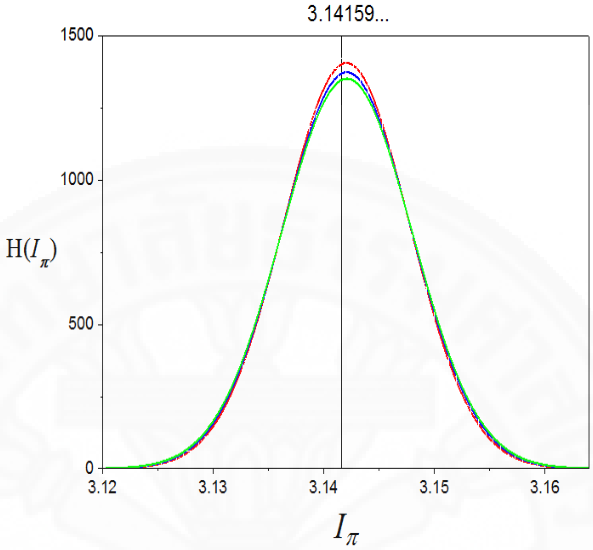


Figure 3.5: Best-fit Gaussians for the histograms of the estimated integral I_π obtained using the conventional and two new termination criteria, using the 80%-flatness criterion, for 10000 independent runs. The central line is the exact value.

To investigate how the termination criteria affect the final results, we observed the Gaussian best fits for the histograms of the estimated integrals, as shown in Figure 3.5. The vertical line indicates the exact value I_π. The colors (blue, green, and red) represent the results using the first, second, and conventional criteria, respectively. We can observe that there was no significantly difference between the heights of the histograms and that peaks were centered at the exact value. This demonstrates that the new termination criteria do not decrease the accuracy of the numerical results.

In the next chapter, we show the behavior of estimated integrals up to six dimensions and a corner-peak integral in high dimensions. We then derive the two new criteria for each dimension, give the numerical results, and compare the accuracy and the CPU time of the algorithms.

CHAPTER 4

NUMERICAL RESULTS

In this chapter, we compare the accuracy, the errors of the numerical estimates obtained, and the CPU time of the conventional termination criterion and the two new criteria. We discuss two applications: the multidimensional integrals and a corner-peak integral.

4.1 Multidimensional integrals

To compare the efficiency of the conventional termination criterion and the two new criteria, we used integrals of up to six dimensions from Refs. [3, 6, 7].

The one- and two-dimensional integrals were defined by

$$I_{1D} = \int_{-2}^{2} (x^5 - 4x^3 + x^2 - x)\sin(4x)dx, \tag{4.1}$$

$$I_{2D} = \int_{-1}^{1}\int_{-1}^{1} (x_1^6 - x_1x_2^3 + x_1^2x_2 + 2x_1)\sin(4x_1+1)\cos(4x_2)dx_1dx_2. \tag{4.2}$$

Unlike that in I_π, the integrands in I_{1D} and I_{2D} are not bijective functions, so it is difficult to determine $g_{ex}(y)$ precisely. The exact values of the integrals obtained by the analytical method in Ref. [3] are $I_{1D} = 1.63564436296\ldots$ and $I_{2D} = -0.017979927\ldots$.

The multidimensional integrals (for $n = 3, 4, 5,$ and 6) were defined by

$$I_{nD} = \int_0^1\int_0^1\cdots\int_0^1 \prod_{i=1}^{n}\cos(ix_i)dx_1dx_2\cdots dx_n \tag{4.3}$$

and their exact values were given by

$$I_{nD} = \prod_{i=1}^{n}\left(\tfrac{\sin(ix_i)}{i}\right). \tag{4.4}$$

That is, $I_{3D} = 0.017996268\ldots, I_{4D} = -0.003404905\ldots, I_{5D} = 0.000653009\ldots,$ and $I_{6D} = -0.000030410\ldots$. These integrals have no specific mathematical significance but are useful for comparing the accuracy of the three termination criteria used in this work. Note that the characteristics of the integrand function of I_{3D-6D} were the same, whereas those of I_{1D} and I_{2D} were different. Thus, these integrals could not be directly compared.

Next, we determined the first and second termination criteria for $I_{nD}, n = 1, 2, \ldots, 6$, by investigating the behavior of the integrals, as shown in Figure 4.1.

4.1.1 The first termination criterion

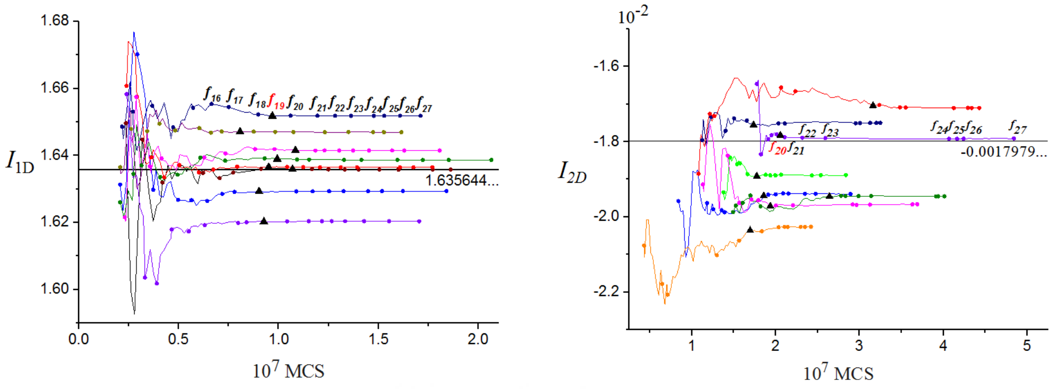
We determined the new f_{final} by investigating the behavior of the integrals, as shown in Figure 4.1. It can be observed that the behavior of integrals I_{1D-6D} stabilized at some value before completion of the simulation, making it unnecessary to perform the simulation to the end.

In one dimension (Figure 4.1a), all the curves stabilized at f_{19} (triangle points). Thus, the new suitable termination criterion for I_{1D} was $f_{final} = f_{19}$. Similarly, in two dimensions (Figure 4.1b), all the curves stabilized at f_{20}. Therefore, the new f_{final} for I_{2D} was f_{20}. By investigating the behavior of the integrals for I_{3D-6D}, as shown in Figure 4.1, the new f_{final} (triangle points) of the integrals I_{nD} for $n = 3, \ldots, 6$ were determined as f_{17}, f_{18}, f_{20}, and f_{22}, respectively.

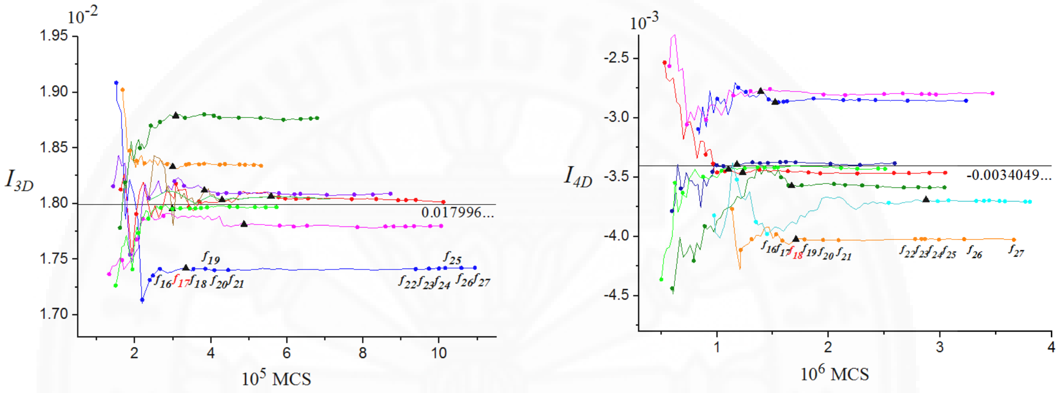
Next, we compare the accuracy, the average CPU time per run, and the mean final order of the modification factor (f_{final}) of the conventional termination criterion and the first new termination criterion. We performed 10000 independent WLS runs with $dy = 0.05$ for I_{1D-5D}, and 1000 independent WLS runs with $dy = 0.1$ for I_{6D}. In both cases, an 80%-flatness criterion was used.

Table 4.1 shows the average CPU time per run, the standard deviation of CPU time, and the mean final order of the modification factor (f_{final}) obtained from all independent runs. It was demonstrated that the first new termination criterion required a significantly lower mean final order of the modification factor and a shorter CPU time for all dimensions.

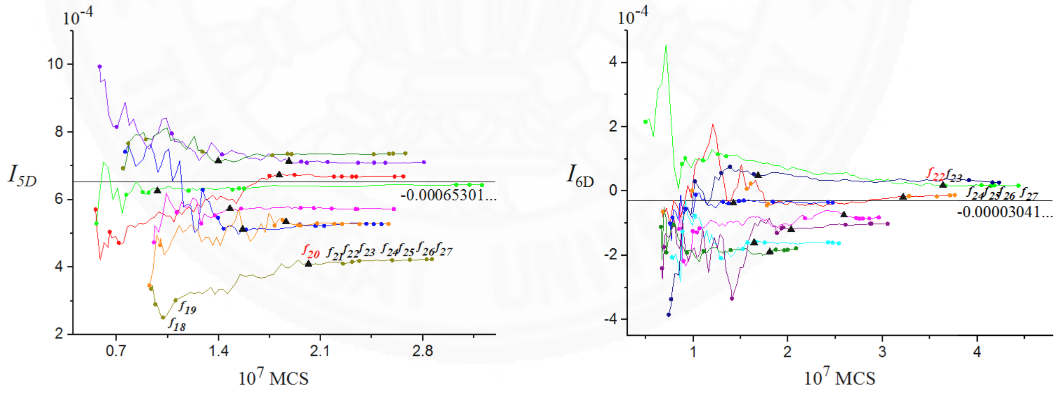
Table 4.2 shows the average of the numerical estimates, the standard error of mean, and the relative error with respect to the exact value of all independent runs obtained using the conventional and the first termination criteria. The standard error of mean σ_{error} is defined by Eq. (3.9). We can observe that,



(a) I_{1D} (b) I_{2D}



(c) I_{3D} (d) I_{4D}



(e) I_{5D} (f) I_{6D}

Figure 4.1: Behavior of the estimated integral during Wang-Landau sampling for eight independent runs using the 80%-flatness criterion, $dy = 0.05$ for I_{1D-5D} and $dy = 0.1$ for I_{6D}. The dots show where the modification factors were updated and the straight line is the exact value.

although the conventional criterion performed the simulation through to the end ($f_{final} = f_{27}$), the relative errors of the integrals did not guarantee the lowest error,

Table 4.1: The average CPU time per run (seconds) and mean final order of the modification factor of the conventional termination criterion and first new termination criterion for I_{1D-6D}.

Integral	Conventional criterion		First new criterion	
	CPU time	f_{final}	CPU time	f_{final}
I_{1D}	7.789(±0.712)	27	4.219(±0.410)	19
I_{2D}	19.547(±6.308)		12.378(±5.033)	20
I_{3D}	0.361(±0.163)	27	0.160(±0.069)	17
I_{4D}	1.917(±0.934)		0.820(±0.345)	18
I_{5D}	15.167(±4.905)		7.847(±2.748)	20
I_{6D}	17.287(±5.148)		11.415(±3.894)	22

Table 4.2: Numerical estimates of integrals obtained using the conventional and the first termination criteria.

Integral	Exact	Conventional criterion		First new criterion	
		Estimate	Relative error	Estimate	Relative error
I_{1D}	16.35646	16.35563($\pm 1.045 \times 10^{-4}$)	0.000498	16.35558($\pm 1.058 \times 10^{-4}$)	0.000529
I_{2D}	-17.97993	-18.00034($\pm 1.750 \times 10^{-5}$)	1.13520	-17.99900($\pm 1.803 \times 10^{-5}$)	1.06098
I_{3D}	17.99627	17.99795($\pm 1.016 \times 10^{-5}$)	0.09335	17.99793($\pm 1.188 \times 10^{-5}$)	0.09227
I_{4D}	-3.40491	-3.39329($\pm 4.837 \times 10^{-6}$)	3.41018	-3.39319($\pm 5.066 \times 10^{-6}$)	3.44088
I_{5D}	0.65301	0.65193($\pm 1.316 \times 10^{-6}$)	1.65760	0.65167($\pm 1.433 \times 10^{-6}$)	2.05553
I_{6D}	-0.000304	-0.000341($\pm 2.979 \times 10^{-6}$)	1.22248	-0.000338($\pm 3.113 \times 10^{-6}$)	1.12510

Values as follows: I_{2D-5D} multiplied by 10^{-3}, and I_{1D} and I_{6D} multiplied by 10^{-1}.

as better precision could arise from the first criterion. This is because the estimated integrals fluctuated slightly in high-digit precision after stabilizing. Thus, the increment of the order of the modification factor is unnecessary. We can see that the relative errors of I_{2D}, I_{3D}, and I_{6D} obtained from the first termination criterion were lower than those from the conventional criterion while the results in other dimensions were higher.

4.1.2 The second termination criterion

We determined the appropriate *limits* for $I_{nD}, n = 1, 2, \ldots, 6$, by considering the behavior of the estimated integrals after stabilizing.

For I_{1D}, Figure 4.2 shows that after the new f_{final} was reached, the variance in the relative error ε was less than 10^{-4}. Therefore, the appropriate *limit* of the integral I_{1D} was 10^{-4}. Similarly, for I_{2D-4D}, we can observe that after the new f_{final} was reached, the variance in the relative error ε was less than 10^{-3}. Therefore, the appropriate *limit* of the integrals I_{2D-4D} was 10^{-3}. For I_{5D} and I_{6D}, after the new f_{final} was reached the variance in the relative error ε was less than 10^{-2} and 10^{-1}, respectively. These values were therefore shown to be the suitable *limits* of the integrals I_{5D} and I_{6D}. Note that the increment of the value of *limit* does not depend on the dimensions, but on the exact values. For example, for the integral I_{1D}, after the estimated integral stabilized, the absolute errors $|I(t) - I(0)|$ changed in the fifth decimal place while the estimates $I(0)$ were around $1.6356\ldots$. The change in the relative error was therefore also in the fifth decimal place. For this reason, we determined that the appropriate *limit* of the integral I_{1D} was 10^{-4}. For the integral I_{6D}, we found that after the estimated integral stabilized, the absolute errors $|I(t) - I(0)|$ changed in the seventh decimal place while the estimates $I(0)$ were around $-0.0000304\ldots$, and the change in the relative error was in the second decimal place. Therefore, the appropriate *limit* of the integral I_{6D} was 10^{-1}.

To compare the accuracy, the average CPU time per run, and the mean final order of the modification factor (f_{final}) of the conventional termination criterion and the second new termination criterion, we performed all independent runs under the same conditions as the first criterion.

Table 4.3: The average CPU time per run (seconds) and mean final order of the modification factor of the conventional termination criterion and the second termination criterion for I_{1D-6D}.

Integral	Conventional criterion		Second new criterion	
	CPU time	f_{final}	CPU time	f_{final}
I_{1D}	7.789(±0.712)	27	5.238(±0.641)	20.41
I_{2D}	19.547(±6.308)		15.234(±5.702)	22.76
I_{3D}	0.361(±0.163)	27	0.197(±0.099)	18.79
I_{4D}	1.917(±0.934)		1.508(±0.770)	23.45
I_{5D}	15.167(±4.905)		10.001(±3.942)	21.93
I_{6D}	17.287(±5.148)		12.446(±5.228)	22.93

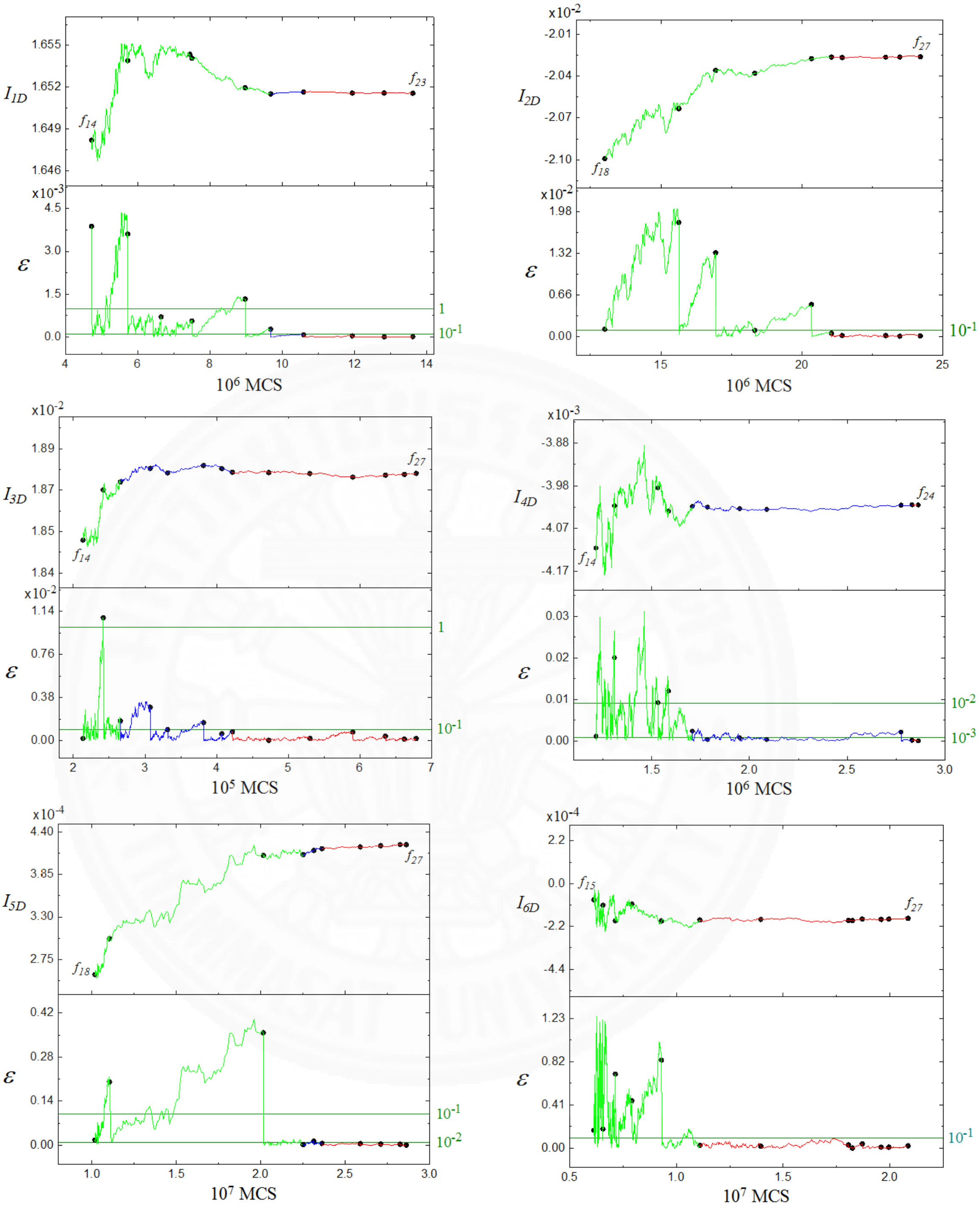


Figure 4.2: (Upper panel) Evolution of the estimated integrals I_{1D-6D} during WLS for a single run. The dots show where the modification factor was updated. (Lower panel) Evolution of the checking parameter ε during the same simulation.

Table 4.3 shows the average CPU time per run, the standard deviation of CPU time, and the mean final order of the modification factor (f_{final})

obtained from all independent runs. The results indicate that the second new termination criterion required significantly lower mean final orders of the modification factor than the conventional criterion. The average CPU times per run of I_{1D-6D} obtained using the second termination criteria were shorter than that of the conventional termination criterion.

Table 4.4: Numerical estimates of integrals obtained using the conventional and the second termination criteria.

Integral	Exact	Conventional criterion		Second new criterion	
		Estimate	Relative error	Estimate	Relative error
I_{1D}	16.35646	16.35563($\pm1.045 \times 10^{-4}$)	0.000498	16.35560($\pm1.050 \times 10^{-4}$)	0.000513
I_{2D}	-17.97993	-18.00034($\pm1.750 \times 10^{-5}$)	1.13520	-17.99839($\pm1.759 \times 10^{-5}$)	1.02708
I_{3D}	17.99627	17.99795($\pm1.016 \times 10^{-5}$)	0.09335	17.99894($\pm1.171 \times 10^{-5}$)	0.14848
I_{4D}	-3.40491	-3.39329($\pm4.837 \times 10^{-6}$)	3.41018	-3.39352($\pm4.846 \times 10^{-6}$)	3.34506
I_{5D}	0.65301	0.65193($\pm1.316 \times 10^{-6}$)	1.65760	0.65302($\pm1.356 \times 10^{-6}$)	0.01340
I_{6D}	-0.000304	-0.000341($\pm2.979 \times 10^{-6}$)	1.22248	-0.000361($\pm3.651 \times 10^{-6}$)	1.88501

Values as follows: I_{2D-5D} multiplied by 10^{-3}, and I_{1D} and I_{6D} multiplied by 10^{-1}.

Table 4.4 shows the average of the numerical estimates, the standard error of mean, and the relative error with respect to the exact value of all independent runs obtained using the conventional and the second termination criteria. We can observe that the estimated integrals of I_{2D}, I_{4D}, and I_{5D} were more accurate than those of the conventional criterion but those of other dimensions were less accurate. As in the case of the first termination criterion, the increment of the order of the modification factor did not guarantee the lowest error. The second termination criterion can also lead to better accuracy.

Finally, we compared the relative error and the average CPU time per run of the conventional and the both new criteria. Table 4.5 shows as percentages the change in relative error and reduction in CPU time when applying the new criteria instead of the conventional criterion. The percentage of relative error represents the comparative performance of the new and the conventional criteria in approximating the exact value. If the sign is negative, the new criterion was more accurate. Even when the relative error obtained using the new criterion was greater than that produced by the conventional criterion, the level of precision

Table 4.5: The percentage of the change in relative error and the percentage of reduction in CPU time for I_{1D-6D} using the new criteria compared with the conventional criterion.

Integral	Percentage of the change in relative error		Percentage of reduction in CPU time	
	First new criterion	Second new criterion	First new criterion	Second new criterion
I_{1D}	$+0.00031$	$+0.00015$	45.83	32.75
I_{2D}	-0.00742	-0.01081	36.68	22.06
I_{3D}	-0.00011	$+0.00551$	55.68	45.43
I_{4D}	$+0.00307$	-0.00651	57.22	21.34
I_{5D}	$+0.03979$	-0.16442	48.26	34.06
I_{6D}	-0.97374	$+6.62528$	33.97	28.01
Average			46.27	30.61

remained acceptable. For example, in I_{6D}, although the percentage of the relative error increased when using the second criterion, the change in the relative error was in the sixth decimal place. The numerical results suggest that the first criterion is more appropriate for reducing the CPU time, while the second criterion is more suitable if the estimation of the integrals needs to be within specified limits.

As well as being applicable to integrals up to six dimensions, the new criteria can also be applied to other functions of integral without any limitations such as polynomial functions, rational functions, and exponential functions. In the next section, to show the efficiency of the new termination criteria for other integrals, we will apply the new criteria to the corner-peak integral.

4.2 The corner-peak integral

The corner-peak integral is a one of six test integrand families designed by Genz [10]. The six test integrand families have properties that allow them to be used to compare the efficiency of algorithms. They are analytically computable to high precision and easy to estimate in a short time. The Genz testing package has been used in many studies [10, 11, 12, 13, 14, 15, 16, 17, 18], and was derived by theoretical analysis in [14]. The corner-peak integrand is defined by

$$f(x) = \frac{1}{(1 + \mathbf{c} \cdot \mathbf{x})^{s+1}}, \tag{4.5}$$

where the integration domain is on $[0,1]^s$, and s is the number of dimensions. The parameter $\mathbf{c} = (c_1, \ldots, c_s)$ is generated as a uniform random vector in $[0,1]$. Different tests can be obtained by varying the parameter $\mathbf{c}$. The change in the parameter $\mathbf{c}$ affects the difficulty of the integration, such as the height and position of the peak. Figure 4.3 shows the graphs of the corner-peak integrand in one dimension with the following values of c: 0.10 (red), 0.25 (blue), 0.50 (green), 0.75 (pink), and 1 (sky blue). When c increases, the lower bound of the integrand decreases and the curve is more concave.

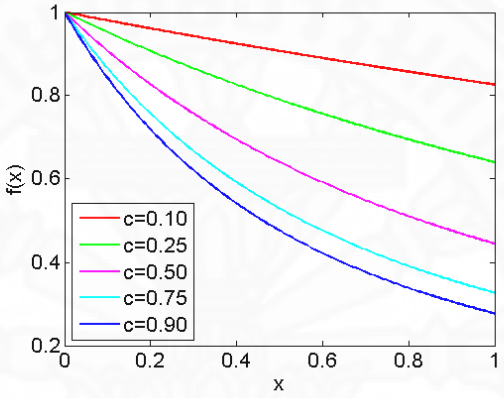
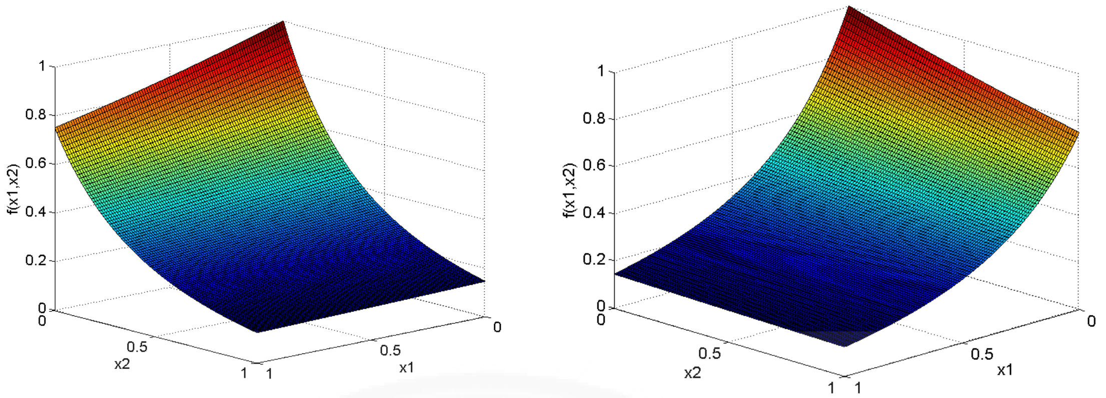


Figure 4.3: Plots of the corner-peak integrand in one dimension for a given c.

Figure 4.4 shows the graphs of the corner-peak integrand in two dimensions with $\mathbf{c} = (c_1, c_2)$: $(0.1, 0.9)$ and $(0.9, 0.1)$, where c_1 and c_2 are the coefficients of x_1 and x_2. The position of one corner is fixed at the point $(x_1, x_2, f(\mathbf{x})) = (0, 0, 1)$ while that of the other corners depends on the value of $\mathbf{c}$. If c_i increases, the corner on the x_i-axis will slide down.

The exact value of the integral in s dimensions can be derived by the analytical method and is defined by

$$\begin{aligned} &\int_{r_s}^{t_s} \cdots \int_{r_1}^{t_1} \left(1 + \sum_{i=1}^{s} c_i x_i\right)^{-(s+1)} dx_1 \cdots dx_s \\ &= \frac{1}{s! \prod_{i=1}^{s} a_i} \sum_{w \in \{0,1\}^s} \frac{(-1)^{\|w\|_1}}{1 + v_{1,w} c_1 + \cdots + v_{s,w} c_s} \end{aligned} \qquad (4.6)$$



(a) $\mathbf{c} = (0.1, 0.9)$

(b) $\mathbf{c} = (0.9, 0.1)$

Figure 4.4: A plot of the corner-peak integrand in two dimensions for a given $\mathbf{c} = (c_1, c_2)$.

with $v_{i,w}$ defined as

$$v_{i,w} = \begin{cases} r_i & w_i = 0 \\ t_i & w_i = 1, \end{cases}$$

where w_i is used to represent the lower or upper boundaries of the integrand in the summation term in Eq. (4.6). The number of terms in the summation is 2^s. For example, in one dimension, the exact value of the integral is defined by

$$\int_0^1 \left(1 + c_1 x_1\right)^{-2} dx_1 = \frac{1}{2! \cdot c_1} \sum_{w \in \{0,1\}^s} \left(\frac{(-1)^{\|w\|_1}}{1 + v_{1,w} c_1} \right).$$

The summation is over $2^s = 2^1 = 2$ terms, and $w = 0$ for the first term and $w = 1$ for the second term. Then, the exact value of the integral is

$$\begin{aligned} \int_0^1 \left(1 + c_1 x_1\right)^{-2} dx_1 &= \frac{1}{2! \cdot c_1} \left(\frac{(-1)^{\|0\|_1}}{1 + v_{1,0} c_1} + \frac{(-1)^{\|1\|_1}}{1 + v_{1,1} c_1} \right) \\ &= \frac{1}{2! \cdot c_1} \left(\frac{1}{1 + 0 \cdot c_1} + \frac{(-1)}{1 + 1 \cdot c_1} \right). \end{aligned}$$

In two dimensions, the exact value of the integral is defined by

$$\int_0^1 \int_0^1 \left(1 + c_1 x_1 + c_2 x_2\right)^{-3} dx_1 dx_2 = \frac{1}{3!(c_1 \cdot c_2)} \sum_{w \in \{0,1\}^s} \left(\frac{(-1)^{\|w\|_1}}{1 + v_{1,w} c_1 + v_{2,w} c_2} \right).$$

There are four possible values of w: $(0,0), (0,1), (1,0)$, and $(1,1)$. The order of values in w corresponds to the terms in the summation. Then, the exact value of the integral is

$$
\begin{aligned}
&\int_0^1 \int_0^1 \left(1 + c_1 x_1 + c_2 x_2\right)^{-3} dx_1 dx_2 \\
&= \frac{1}{3!(c_1 \cdot c_2)} \left(\frac{(-1)^{\|(0,0)\|_1}}{1 + v_{1,0}c_1 + v_{2,0}c_2} + \frac{(-1)^{\|(0,1)\|_1}}{1 + v_{1,0}c_1 + v_{2,1}c_2} + \frac{(-1)^{\|(1,0)\|_1}}{1 + v_{1,1}c_1 + v_{2,0}c_2} \right. \\
&\quad \left. + \frac{(-1)^{\|(1,1)\|_1}}{1 + v_{1,1}c_1 + v_{2,1}c_2} \right) \\
&= \frac{1}{3!(c_1 \cdot c_2)} \left(\frac{1}{1 + 0 \cdot c_1 + 0 \cdot c_2} + \frac{(-1)}{1 + 0 \cdot c_1 + 1 \cdot c_2} + \frac{(-1)}{1 + 1 \cdot c_1 + 0 \cdot c_2} \right. \\
&\quad \left. + \frac{1}{1 + 1 \cdot c_1 + 1 \cdot c_2} \right).
\end{aligned}
$$

In this work, the two new termination criteria were applied to the approximation of the corner-peak integrals in 5, 8, and 10 dimensions. In order to obtain the estimated result in each dimension, the 20 different sets of the parameter **c** were substituted into Eq. (4.5), as in Refs. [10, 13, 15, 16, 18]. There are two main differences from the previous section.

1. The behavior of the corner-peak integrand has a sharp peak and is narrow only at the corner. If an equal bin width is used, satisfying the flatness criterion will require a very large number of MCSs. This means that some g(y) cannot be normalized. In order to solve this problem, the width of each bin should be fixed to a different length. The interval of $f(x)$ needs to be divided into unequal-length subintervals.

2. Since the different values of **c** provide a different exact integral, in order to compare the accuracy, we use the number of correct digits instead of the relative errors of the estimated integrals. The number of correct digits d [14] is given by

   $$d_k = -\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} |I_i|\right) - \log_{10}|I_k - Q_k|, k = 1, \ldots, n, \quad (4.7)$$

   where I_k is the exact value, Q_k is the estimated result, and n is the number of sets of parameter **c**. The number of correct digits represents the digit precision after the last zero.

Note that the parameter $\mathbf{c}$ is normalized to satisfy $\|\mathbf{c}\| = 0.925$ for $s = 5$, $\|\mathbf{c}\| = 1.48$ for $s = 8$, and $\|\mathbf{c}\| = 1.85$ for $s = 10$, as in Refs. [11, 13, 18]. It is fixed to make the integrand have equal boundaries in all independent runs for each dimension. The exact values of the integral are $0.123282405\ldots (\pm 0.00374)$ for 5 dimensions, $0.010987099\ldots (\pm 0.00069)$ for 8 dimensions, and $0.001544904\ldots$ (± 0.00013) for 10 dimensions, where values in brackets are the standard deviation. The upper bounds of the integrands are 1 for all dimensions. The lower bounds are 0.019652 for 5 dimensions, 0.281796×10^{-3} for 8 dimensions, and 0.992445×10^{-5} for 10 dimensions. In this work, for each value of $\mathbf{c}$, we performed 100 independent runs for 5 and 8 dimensions, and 20 independent runs for 10 dimensions.

Next, we will discuss the numerical results obtained by the first and second new termination criteria. First, we determined the first new termination criterion by investigating the mean of the number of correct digits given by

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i. \tag{4.8}$$

The standard deviation σ is defined by

$$\sigma = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(\bar{d} - d_i)^2}. \tag{4.9}$$

Figs. 4.5, 4.6, and 4.7 shows the behavior of $\bar{d}$ in 5, 8, and 10 dimensions. The error bars along the vertical axis are the values of σ while the error bars along the horizontal axis are the standard deviations of the number of Monte Carlo trials. In 5 dimensions, $\bar{d}$ stabilized at f_{15}. In 8 and 10 dimensions, $\bar{d}$ stabilized at f_{16}. Note that the points of $\bar{d}$ after stabilizing are represented by the red points. The green line is the average of all red points and the blue lines are the bounds of error bar after stabilizing. Therefore, the first new termination criterion was $f_{final} = f_{15}$ for 5 dimensions and $f_{final} = f_{16}$ for 8 and 10 dimensions.

Second, we determined the termination criterion (the appropriate *limit*) by considering the behavior of the estimated integrals after stabilizing. In 5 dimensions, the absolute errors $|I(t) - I(0)|$ changed in the fifth decimal place while the estimates $I(0)$ were around $0.1\ldots$, and the change in the relative error was

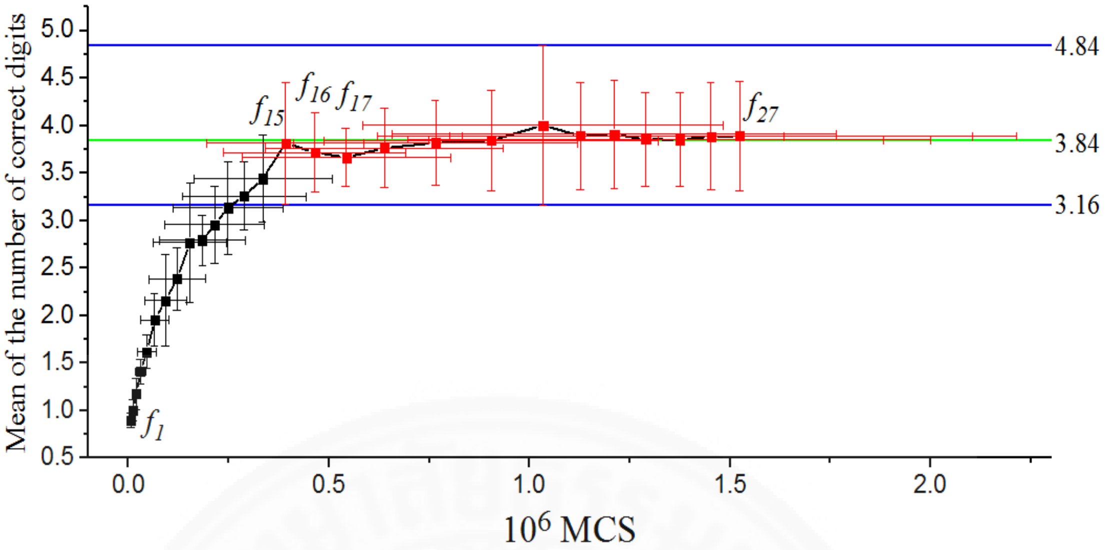


Figure 4.5: $\bar{d}$ of the corner-peak integral in 5 dimensions.

in the fourth decimal place. Therefore, the appropriate *limit* of the corner-peak integral in 5 dimensions was 10^{-3}. In 8 dimensions, the absolute errors $|I(t)-I(0)|$ changed in the sixth decimal place while the estimates $I(0)$ were around $0.01\ldots$, and the change in the relative error was in the fourth decimal place. In 10 dimensions, the absolute errors $|I(t)-I(0)|$ changed in the seventh decimal place while the estimates $I(0)$ were around $0.001\ldots$, and the change in the relative error was in the fourth decimal place. Thus, the suitable *limit* for 8 and 10 dimensions were 10^{-3}.

Next, we compared the accuracy and the CPU time of the conventional criterion and the new criteria. Table 4.6 shows the mean final order of the modification factor, the average CPU time per run (seconds), the mean of the number of correct digits, and the standard deviation obtained using the conventional termination criterion and the two new termination criterion for the corner-peak integral in 5, 8, and 10 dimensions. Each value was obtained by performing the simulations under the same conditions as in the first criterion. The results, as shown in Table 4.6, indicated that the two new termination criteria required significantly lower mean final orders of the modification factor and that the first criterion required less than the second.

In 5 dimensions, the average CPU times per run obtained using the

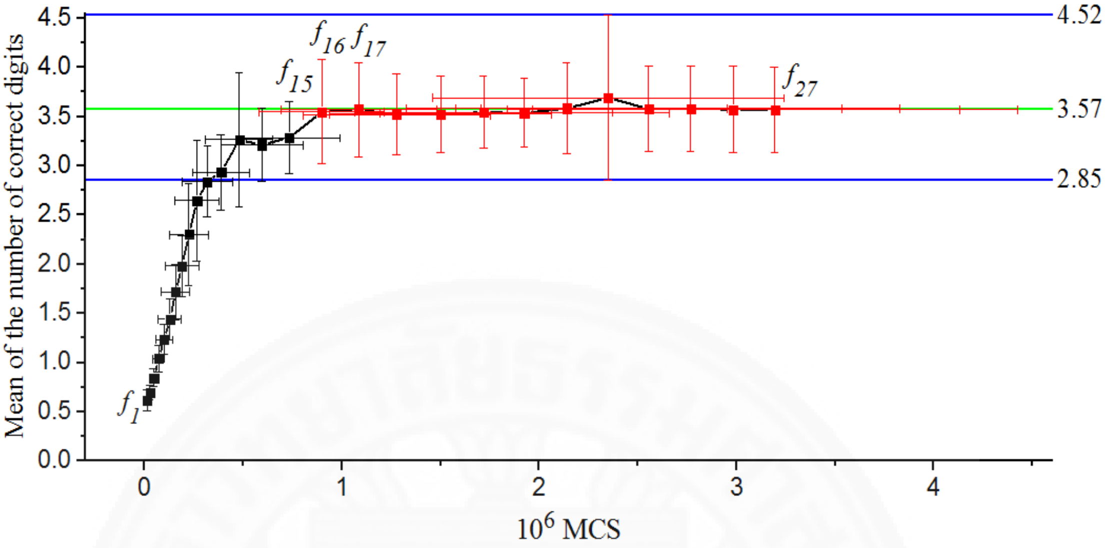


Figure 4.6: $\bar{d}$ of the corner-peak integral in 8 dimensions.

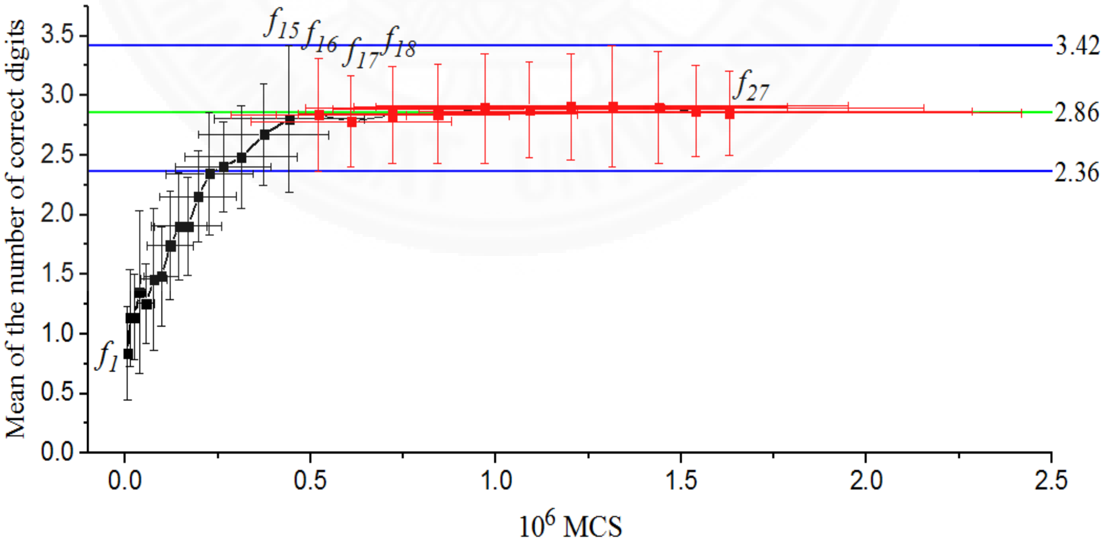


Figure 4.7: $\bar{d}$ of the corner-peak integral in 10 dimensions.

Table 4.6: The mean final order of the modification factor, the average CPU time per run (seconds), the mean of the number of correct digits, and the standard deviation obtained using the conventional termination criterion and the two new termination criteria for the corner-peak integral.

5 dimensions

	Conventional criterion	First new criterion	Second new criterion
f_{final}	27	15	16.49
CPU time	0.3770	0.0971	0.1273
$\bar{d}$	3.8913	3.8164	3.8267
σ	0.5649	0.6253	0.5797

8 dimensions

	Conventional criterion	First new criterion	Second new criterion
f_{final}	27	16	18.19
CPU time	1.0956	0.3083	0.4641
$\bar{d}$	3.5699	3.5479	3.5085
σ	0.4320	0.5303	0.3669

10 dimensions

	Conventional criterion	First new criterion	Second new criterion
f_{final}	27	16	19.42
CPU time	0.5229	0.1427	0.2502
$\bar{d}$	2.8527	2.8048	2.9170
σ	0.3469	0.5967	0.8495

first and the second termination criteria were shorter than that of the conventional termination criterion by 74.24 and 66.23 percent while the values of $\bar{d}$ using the first and the second termination criteria were less than that of the conventional termination criterion by only 1.92 and 1.66 percent. Since the exact value of the integral was 0.1232824046 . . ., three correct digits means that the estimated values had an accuracy of three decimal points.

In 8 dimensions, the average CPU times per run obtained using the first and the second termination criteria were shorter than that of the conventional termination criterion by 71.86 and 57.64 percent. The values of $\bar{d}$ using the first and the second termination criteria were less than that of the conventional termination criterion by only 0.62 and 1.72 percent. Since the exact value of the integral was 0.010987099 . . ., three correct digit means that the estimated values had an accuracy of four decimal points.

In 10 dimensions, the average CPU times per run obtained using the first and the second termination criteria were shorter than that of the conventional termination criterion by 72.71 and 52.15 percent. The value of $\bar{d}$ using the first termination criterion was lower than that using the conventional termination criterion by 1.68 percent, while the value of the second was greater than that of the conventional criterion by 2.25 percent. This indicated that the increment of the order of the modification factor did not guarantee better accuracy as was the case for the integrals discussed in the previous section. Since the exact value of the integral was 0.001492143 . . ., two correct digit means that the estimated values had an accuracy of four decimal points.

This demonstrated that the two new criteria can reduce the CPU time. Although the new criteria provide less precision than the conventional criterion in most dimensions, the precision of the estimated value was within acceptable limits and the second new criterion gave better precision than the conventional criterion in 10 dimensions.

CHAPTER 5

CONCLUSIONS

In this work, we presented two new termination criteria from Caparica and Cunha-Netto [8] and Caparica [9] for Wang-Landau sampling in high-dimensional numerical integration. Our main objective was to reduce the CPU time while keeping the accuracy within an acceptable range. An investigation of the behavior of estimated integrals showed them to stabilize before the simulation completed, making it unnecessary to perform the simulation to the end. The new criteria suggest the use of a new order of modification factor (f_{final}) to terminate the simulation, instead of the fixed order of the modification factor used in the conventional Wang-Landau algorithm. In addition, we used the approximation proposed by Atisattapong and Maruphanton [7] to calculate the integrals. This approximation eliminates the bin width effect and the saturation of error for large bin widths.

The procedures and findings of the two new termination criteria applied to numerical integration were as follows:

1. **The first termination criterion from Caparica and Cunha-Netto [8]**
   We investigated the evolution of integrals for some independent runs. We found that the estimated integrals or the mean of the number of correct digits stabilized at some value before the process completed. Therefore, we chose a new termination criterion (or f_{final}) at which point the estimated integrals stabilized.

2. **The second termination criterion from Caparica [9]**
   We checked the relative error defined by Eq. (3.10) during the simulation. We determined the appropriate *limit* for halting the simulations by investigating the relative error throughout a single run. We found that after the estimated integral stabilized, all val-

ues of relative error were less than a threshold limit value. Then, we determined this value to be the appropriate *limit* for all WLS independent runs. The procedure of the second termination criterion was that when the relative error during the interval $[f_k, f_{k+1}]$ was less than the appropriate *limit*, then the simulation terminated at f_k. The second termination criterion has the additional benefit that it is no longer necessary to determine an f_{final} in advance, as different runs can terminate at different f_{final} values.

The proposed criteria were applied to the estimation of integrals up to six dimensions. The numerical results showed that either the first or the second criterion yielded more accurate results than the conventional criterion. Both new termination criteria also required significantly shorter CPU time, with the first criterion requiring less than the second. Although the relative errors obtained using the new criteria were greater than those from the conventional criterion in some dimensions, the integrals were always estimated within acceptable limits. In short, the first criterion was more appropriate for reducing the CPU time, while the second criterion was more suitable if the estimation of the integrals needs to be within specified limits.

We also applied the two new termination criteria to the corner-peak integral in 5, 8, and 10 dimensions. The corner-peak integrand is a one of six test integrand families designed by Genz [10] which can be used to compare the efficiency of algorithms. We compared the accuracy and CPU time of the conventional and the two new criteria by investigating the evolution of the mean of the number of correct digits. The results showed that both new criteria could reduce significantly the CPU time, and could provide more accurate results. Even when the new criteria provided less precision than the conventional criterion, the accuracy was within acceptance criteria.

To conclude, the great advantage of the new criteria is a reduction of the CPU time. The conventional criterion of the Wang-Landau algorithm does not always provide more accurate results than the first and second new termination criteria, which always achieve acceptable precision, and in some cases outperform the conventional criterion. This suggests that, after the estimated integral has stabilized, an increment of the number of Monte Carlo trials does not affect the convergence of the estimated integral. The new criteria can be easily coded and implemented in programming, by adding or changing a few lines of the Wang-Landau algorithm.

The new termination criteria can be applied to other multidimensional integrals such as ill-behaved integrals. They are also straightforward to implement in the $1/t$ algorithm when approximating integrals in high dimensions.

REFERENCES

[1] Anton, H., Bivens, I. and Davis, S. (2010). *Calculus Late Transcendentals.* 9th edition, John Wiley and Sons, Inc.

[2] Grasselli, M. and Pelinovsky, D. (2008). *Numerical Mathematics.* Jones and Bartlett Publisher, Inc.

[3] Li, Y.W., Wüst, T., Landau, D. P. and Lin, H. Q. (2007). Numerical integration using Wang-Landau sampling, *Comput. Phys. Commun.*, **177**, 524.

[4] Wang, F. and Landau, D. P. (2001). Efficient multiple-range random walk algorithm to calculate the density of states, *Phys. Rev. Lett.*, **86**, 2050.

[5] Belardinelli, R. E. and Pereyra, V. D. (2007). Fast algorithm to calculate density of states, *Phys. Rev. E.*, **75**, 046701.

[6] Belardinelli, R. E., Manzi, S. and Pereyra, V. D. (2008). Analysis of the convergence of the $1/t$ and Wang-Landau algorithms in the calculation of multidimensional integrals, *Phys. Rev. E.*, **78**, 067701.

[7] Atisattapong, W. and Maruphanton, P. (2015). Obviating the bin width effect of the $1/t$ algorithm for multidimensional numerical integration, *Applied Numerical Mathematics*, vol. **104**, 133-140.

[8] Caparica, A. A. and Cunha-Netto, A. G. (2012). Wang-Landau sampling: Improving accuracy, *Phys. Rev. E.*, **85**, 046702.

[9] Caparica, A. A. (2014). Wang-Landau sampling: A criterion for halting the simulations, *Phys. Rev. E.*, **89**, 043301.

[10] Genz, A. (1984). Testing multidimensional integration routines, *Tools, Methods and Languages for Scientic and Engineering Computation*, 81-94.

[11] Novak, E. and Ritter, K. (1996). High dimensional integration of smooth functions over cubes, *Numerische Mathematik*, **75**, 79-97.

[12] Genz, A. (1987). A package for testing multiple integration subroutines, *Numerical Integration: Recent Developments, Software and Applications*, 337-340.

[13] Barthelmann, V., Novak, E. and Ritter, K. (2000). High dimensional polynomial interpolation on sparse grids, *Adv. Comput. Math.*, **12**, 273-288.

[14] Schürer, R. M. (2001). *High-dimensional numerical integration on parallel computers*. na.

[15] Hahn, T. (2005). CUBA-a library for multidimensional numerical integration, *Computer Physics Communications*, **168**, 78.

[16] Ascasibar, Y. (2008). FiEstAS sampling - a Monte Carlo algorithm for multidimensional numerical integration, *Computer Physics Communications*, **179**, 881.

[17] Agarwal, N. and Aluru, N. R. (2011). Weighted Smolyak algorithm for solution of stochastic differential equations on non-uniform probability measures, *Int. J. Numer. Meth. Engng.*, **85**, 1365–1389.

[18] Harase, S. (2015). Quasi-Monte Carlo point sets with small t-values and WAFOM, *Applied Mathematics and Computation*, vol. **254**, 318-326.

[19] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007). *Numerical Recipes: the Art of Scientific Computing*. Cambridge University Press, Cambridge.

[20] Chapra, S. C. and Canale, R. P. (2002). *Numerical Methods for Engineers with Software and Programming Applications*. McGraw-Hill Higher Education.

[21] Landau, D. P. and Binder, K. (2005). *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, New York.

[22] Belardinelli, R. E. , Manzi, S. and Pereyra, V. D. (2008). Analysis of the convergence of the 1/t and Wang-Landau algorithms in the calculation of multidimension, *Phys. Rev. E.*, **78**, 067701.

[23] Katzgraber, H.G. (2011). *Introduction to Monte Carlo Methods.* Texas University Department of Physics and Astronomy College Station.

[24] Li, Y. W., Wüst, T., Landau, D. P. and Lin, H. Q. (2007). Numerical integration using Wang-Landau sampling, *Comput. Phys. Commun.*, **177**, 524.

[25] Gould, H., Tobochnik, J. and Christian, W. (2006). *An Introduction to Computer Simulation Methods: Applications to Physical Systems.* 3rd edition, Addison-Wesley.

[26] Baierlein, R. (1999). *Thermal physics.* United Kingdom at the University Press, Cambridge.

[27] Landau, D. P., Tsai, S. and Exler, M. (2004). A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling, *Am. J. Phys.*, **72**, 1294-1320.

[28] Berg, B. A. and Celik, T. (1992). New approach to spin-glass simulations, *Phys. Rev. Lett.*, **69**, 2292.

[29] Berg, B. A. and Janke, W. (1998). Multioverlap Simulations of the 3D Edwards-Anderson Ising Spin Glass, *Phys. Rev. Lett.*, **80**, 4771.

[30] Landau, D. P., Lewis, S. P. and Schuttler, H. B. (2011). *Computer Simulation Studies in Condensed-matter Physics XII.* Springer, Berlin, Heidelberg.

[31] Alves, N. S. and Hansmann, U. (2000). Partition function zeros and finite size scaling of helix-coil transitions in a polypeptide, *Phys. Rev. Lett.*, **84**, 1836.

[32] Yan, Q. L., Faller, R. and Pablo, J. J. (2002). Density-of-states Monte Carlo method for simulation of fluids, *J. Chem. Phys.*, **116**, 8745.

[33] Jain, T. S. and Pablo, J. J. (2003). Calculation of interfacial tension from density of states, *J. Chem. Phys.*, **118**, 4226.

[34] Vlugt, T. J. H. (2002). Measurement of chemical potentials of systems with strong excluded volume interactions by computing the density of states, *Mol. Phys.*, **100**, 2763 .

[35] Tröster, A. and Dellago, C. (2005). Wang-Landau sampling with self-adaptive range, *Phys. Rev. E.*, **71**, 066705.

[36] Beale, P. D. (1996). Exact distribution of energies in the two-dimensional Ising model, *Phys. Rev. Lett.*, **76**, 78.

[37] Green, M. S. (1971). *Critical Phenomena.* Academic Press, New York.

[38] Fisher, M. E. and Barber, M. N. (1972). Scaling Theory for Finite-Size Effects in the Critical Region, *Phys. Rev. Lett.*, **28**, 1516.

[39] Domb, C. and Lebowitz, J. L. (1983). *Phase Transitions and Critical Phenomena.* Vol. 8, Academic Press, New York.

BIOGRAPHY

Name	Miss Jirarat Chairat
Date of Birth	March 8, 1992
Educational Attainment	2010-2013: Bachelor of Science in Mathematics (First Honor Class), Thammasat University, Thailand
Scholarship	2013: Prof. Dr. Tab Nilaniti's Scholarship 2014-2015: Graduate Scholarship from Faculty of Science and Technology in Thammasat University

THAMMASAT UNIVERSITY