修士論文

GPU を用いたモンテカルロ計算の高速化に関する研究

名古屋大学
工学部物理工学科
量子エネルギー工学コース
大久保卓哉

平成 28 年 2 月

目次

第 1 章 序論 1
1.1. 背景 1
1.2. モンテカルロ法による炉心計算 2
1.3. 計算機進化の傾向と GPU の利用 3
1.4. 本研究の目的 5
1.5. 本論文の構成 5
第 2 章 モンテカルロ法による中性子輸送計算 6
2.1. 本章の概要 6
2.2. 中性子ランダムウォーク 6
2.2.1. 衝突点サンプリング 6
2.2.1. 多領域体系 7
2.2.2. 境界条件 8
2.3. 世代と固有値計算 8
2.4. 非アナログモンテカルロ法 9
2.4.1. ウェイトの導入 10
2.4.2. ロシアンルーレット 12
2.4.3. エスティメータ 13
2.5. 仮想散乱による追跡計算 14
2.5.1. 仮想散乱断面積 14
2.5.2. 衝突エスティメータ 15
2.5.3. 完全反射境界条件 16
2.6. 本章のまとめ 19
第 3 章 並列計算と GPU 20
3.1. 本章の概要 20
3.2. GPU の構造と特徴 20
3.3. 並列計算 21
3.3.1. 並列化の基礎的概念 21
3.3.2. 並列化プログラミングモデル 22
3.3.3. アムダールの法則 25
3.3.4. メモリ競合と競合状態 26
3.4. OpenCL の特徴 27
3.5. OpenCL プログラミングの基本 28
3.6. GPU の特性を考慮したプログラミング手法 30
3.7. 本章のまとめ 32

第 4 章 GPU を用いたモンテカルロ中性子輸送計算コードの実装 33
4.1. 本章の概要 33
4.2. 計算のフローチャート 33
4.3. 乱数ジェネレータ 34
4.4. 固有値計算の実装 35
4.5. 中性子束の計算方法 38
4.5.1. 並列化における問題点 38
4.5.2. 固定小数点数 38
4.5.3. アトミック演算との組合せ 39
4.5.4. 数値計算における精度 40
4.6. 本章のまとめ 40
第 5 章 検証計算 42
5.1. 本章の概要 42
5.2. Takeda ベンチマーク問題 42
5.2.1. 計算体系 42
5.2.2. 実効増倍率の計算結果 44
5.2.3. 中性子束の計算結果 45
5.2.4. 考察 47
5.3. C5G7 ベンチマーク問題 48
5.3.1. 計算体系 48
5.3.2. 実効増倍率の計算結果 51
5.3.3. 核分裂率の計算結果 51
5.3.4. 考察 52
5.4. 乱数同一の場合の計算効率 52
5.4.1. 計算条件 52
5.4.2. 計算結果 53
5.4.3. 考察 53
5.5. Regular-tracking 法での性能評価 54
5.5.1. 追跡方法の違いによる性能差 54
5.5.2. 計算体系 54
5.5.3. 計算結果 55
5.5.4. 考察 56
5.6. 本章のまとめ 56
第 6 章 結論 58
6.1. 結論 58
6.2. 今後の課題 59

参考文献 61
謝辞 62
Appendix..**63**
A.　Takeda ベンチマーク問題計算結果..**63**
B.　C5G7 ベンチマーク問題計算結果 **64**
C.　OpenCL による開発..**71**
公刊論文リスト **76**

第1章　序論

1.1. 背景

原子力エネルギーの利用は、世界中で増え続けるエネルギー需要をまかなうためには不可欠なものである。化石燃料の利用を継続させることは、二酸化炭素やその他の有害物質の排出、資源開発による環境破壊を招く。また、資源の枯渇による値段の高騰というリスクもあり、化石燃料への依存は好ましくない。化石燃料以外に大規模で安定したエネルギー源は現在のところ核分裂エネルギーを利用した原子力しかない。特に日本のような国土にエネルギー源が乏しい国家では、安定してエネルギー源を確保する手段を確保しておくことはエネルギー安全保障上重要である。

しかしながら、原子力には他のエネルギー源と異なるリスクを有する。放射性物質を扱うため、事故時の社会的な影響が大きく、安全確保のために必要なコストが大きい。また原子力発電所は、燃料費の割合が小さい代わりに初期費用が高い。原子力利用を拡大していくためには、安全性を高めることと、十分な経済性を有していることが条件となる。そのため、安全性・経済性を高めるための研究開発が継続して行われている。

原子力発電所では、核分裂反応を起こさせる原子炉、熱エネルギーを回収するための冷却設備、熱エネルギーから電気エネルギーに変換するためのタービン・発電機、そしてその他の補助的な設備から構成されている。このうち、原子炉は原子力発電の安全性、経済性に大きく影響する要素である。原子炉の設計には計算機を用いたコンピュータシミュレーションが用いられており、コンピュータシミュレーションの精度を高めることが原子炉、そして原子力発電所全体の安全性・経済性を高めることにつながる。特に近年では、原子力利用にはより高度な安全性と経済性が求められており、それに応じてコンピュータシミュレーションもより高度化する必要がある。

原子炉の設計は主に次の 3 つの要素から成る。炉内の中性子の振る舞いを評価する核設計、炉内の冷却材の振る舞いを評価する熱水力設計、そして核燃料にかかる圧力や応力、耐食性などを評価する燃料設計である。計算機の能力の向上や、シミュレーションの精度向上により、これら原子炉の設計を高度化することができる。本研究では、このうち核設計の高度化に注目する。

核設計においては、物質内の中性子の輸送を評価する。この輸送を模擬する方法は 2 つに大別でき、決定論的手法と確率論的手法となる。決定論的手法は、中性子の振る舞いを表す輸送方程式に近似を導入し、数値計算によってこれを解く手法である。輸送方程式を近似の導入なしに解くことは、一般的な体系では不可能である。そのため、近似を導入することで輸送方程式を解析解が求められる形に変形して解くことになる。一方の確率論的手法はモンテカルロ法とも呼ばれ、中性子の挙動は乱数を用いて直接シミュレーションされる。乱数を用いるために結果には統計誤差が付随する。統計誤差を小さくするためには

膨大な計算コストが必要であるが、近似を導入せずに物理現象を忠実に取り扱うことができる。近年では、原子炉の安全性・経済性を高めるために、複雑な体系でも精度よく計算が出来る手法のニーズが高まっており、また計算機の性能の向上もあり、モンテカルロ法の利用範囲は広がっている。しかしながら、いまだ計算速度の向上は大きな課題となっている。

一方で、ここ数年の計算機の発達の方向性は、並列計算の性能の向上へとシフトしている。プロセッサのプロセス微細化による周波数の向上が頭打ちとなり、1 つのコアしか持たないプロセッサから、複数のコアを持つプロセッサへと移行している。現状、大規模な数値計算を行う場合は複数のプロセッサへの対応が必須と言える。

より並列計算に特化したプロセッサとして、GPU が近年注目されている。GPU は Graphics Processing Unit の略で、本来画像処理を担当するプロセッサである。GPU は CPU と比べるとはるかに多くの演算コアを有しており、その数は数 100〜数 1000 である。ただし、CPU に比べて 1 つの演算コアは非常に単純な構造をしており、1 つの演算コアの計算性能は低い。しかし、演算コアの数が多いため、プロセッサ全体としては大きな FLOPS を有している。以上で述べたような GPU の性能を活かすためには、この多数の演算コアを利用できるよう、計算アルゴリズムが並列計算に適合していなければならない。しかしながら、モンテカルロ法による炉心計算が GPU 上で行えるようになれば、大幅な速度向上が見込める。

1.2. モンテカルロ法による炉心計算

前節に述べた通り、モンテカルロ法では乱数を用いて中性子輸送をシミュレーションする。多数の中性子の生成から消滅までを追跡し、それらの結果を統計処理することで最終的な結果となる。炉心内の核分裂源から発生した中性子は、原子核に衝突して散乱反応・吸収反応いずれかを起こすか、体系外に漏れ出す。吸収反応の起こった原子核の一部は、核分裂を起こして新たな核分裂源となる。このとき、中性子の飛行方向と飛行距離、衝突点での反応の種類、核分裂が起こるか否かといった事象は乱数を振ることで決定される。Fig. 1.1 に中性子の起こす反応の例を示す。

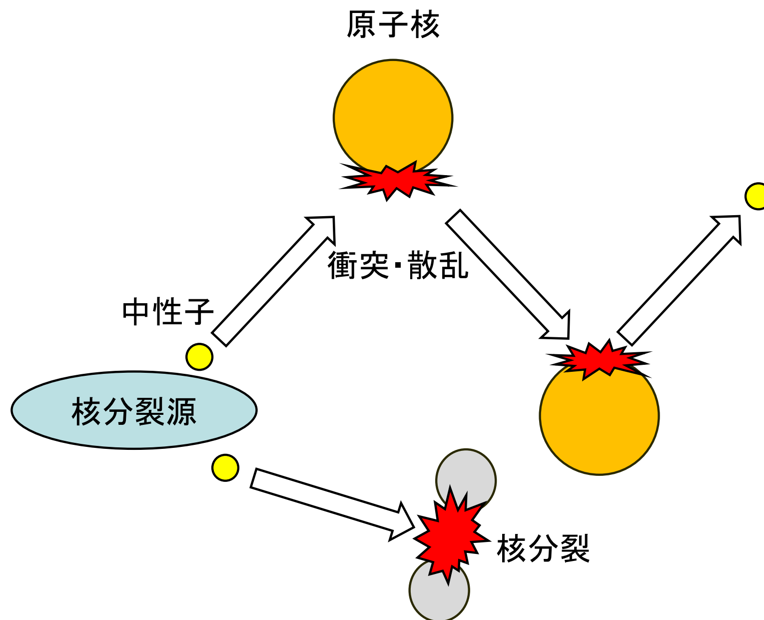

Fig. 1.1 中性子の起こす反応

追跡に使われる中性子の数はヒストリー数と呼ばれるが、このヒストリー数が大きいほど統計誤差は小さくなり、計算結果は信頼のおけるものになる。だが、統計誤差を小さくするためにヒストリー数を大きくとるほど、計算時間は膨大なものとなる。しかし、モンテカルロ法では物理現象を近似なしにそのまま取り扱うことが可能であるという長所が存在する。例えば、曲面を含むような幾何形状を取り扱うことが出来、連続エネルギーモンテカルロコードでは、中性子のエネルギーを群に分けることなく取り扱っている。決定論的手法では、近似の導入により取り扱える問題の範囲が決定されるが、モンテカルロ法は原理的にはあらゆる体系に直接適用することが可能である。

1.3. 計算機進化の傾向と GPU の利用

これまでの計算機の進歩の歴史の中で、CPU(中央演算処理装置)は主要な演算装置として利用されてきた。半導体技術の進歩により、微細化、高集積化が進み、計算機の性能は飛躍的な向上を果たした。これまでの CPU の進歩は、計算機の性能は指数関数的に向上することを示すムーアの法則に従っており、計算機の利用者はその性能向上の恩恵にあずかることができた。しかし、微細化が進み回路が小さくなるにつれ、リーク電流といった技術的困難が生じるようになった。近年は CPU の周波数の向上はほとんど無くなり、プロセッサコアを微細化・複雑化して性能を向上させる手法は頭打ちになりつつある。

そこで注目されているのが並列計算である。プロセッサのコア 1 つだけで 2 倍の性能の向上を達成するのは難しいが、プロセッサコアを 2 つ用意すれば単純に 2 倍の性能が得られる。数式で表すと、プロセッサの性能は式(1.1)で表される。

(1 秒間あたりの計算性能)
=(1 コア 1 クロックあたりの計算性能)×(周波数)×(コア数) (1.1)

1 コア 1 クロックあたりの計算性能は IPC(Instructions Per Clock cycle)と呼ばれる。1 コアあたりの性能を高くするためには IPC を高めるか、コアの動作周波数を高めればよい。しかし、IPC を上げることはプロセッサのアーキテクチャを改良することであり、簡単ではない。また、プロセッサのアーキテクチャに合わせて、コンパイラ等のソフトウェア面での対応も必要になる。周波数については、前述の通りこれまでは回路の微細化により消費電力を増やさずに向上させることが出来たが、近年はほとんど向上しなくなった。周波数を 2 倍にするためには、一般的には 2 倍の電圧が必要であり、消費電力は電圧の 2 乗に比例し、周波数に比例する[1]。すなわち、周波数を増やすことで性能を 2 倍にするには、おおよそ 2^3=8 倍の電力が必要になる。一方で、コア数と消費電力は比例の関係にあり、2 倍の性能を得るために必要な電力は 2 倍である。これが、コアを増やすことで性能を高める動機である。

近年は、CPU のプロセッサコアを複数に増やした製品も製造されるようになった。ただし、複数のプロセッサコアを活用するには、負荷を分割してそれぞれのプロセッサコアが計算を行う必要がある。これまでは、ソフトウェアは並列計算への対応をしなくともハードウェアの進化を享受できたが、今後はソフトウェア側の進歩がなければハードウェアの能力を引き出せなくなると予想される。

CPU の持つプロセッサコアは多くとも 10 個程度となっているが、さらにプロセッサコアを増やし、並列計算による処理を主眼に置いた演算装置が近年注目されている。それが GPU である。GPU は本来、画像処理を担当するプロセッサであったが、画像処理の需要増大に伴い着実に進化してきた。GPU はその役割上、大量のデータを並列処理するのに向いた構造になっている。GPU は数 100～数 1000 にもなる大量のプロセッサコアを有し、単純な時間当たりの演算能力では CPU を凌駕するまでに至っている。ただ、多数のプロセッサコアを持っているとはいえ、プロセッサコア 1 個あたりの性能は CPU に遠く及ばない。GPU の演算能力を活用するには、多数のプロセッサコアに適切に計算負荷を分配することが必須となる。すなわち、ソフトウェアが並列計算に対応することが必要となる。

計算コードを並列計算に対応させ、GPU の演算能力を利用するには、何かしらの開発フレームワークが必要となる。既存の研究では、CUDA が GPU 開発フレームワークとして利用されることが多かった[2][3]。本研究では、開発フレームワークとして OpenCL[4]を

用いる。OpenCL は並列計算を各種計算資源で実行できるよう策定されたオープン規格であり、GPU だけでなく CPU、その他のコプロセッサの演算能力を活用できる。

1.4. 本研究の目的

モンテカルロ法は、中性子の輸送を直接取り扱うことが出来、精度のよい結果が得られるが、付随する統計誤差を小さくするためには大きな計算コストがかかる。そこで本研究では、計算コストの大きいモンテカルロ法を活用するため、GPU の計算能力を適用することにより高速化を行うことを目的とする。具体的には、並列計算が得意である GPU の特徴を生かし、中性子の追跡計算を並列化して高速化を図る。そして CPU と比較して GPU が有意に高速となるよう実装を行うことを目的とする。GPU の計算に適するようにモンテカルロ計算のアルゴリズムに改良を加え、高速化手法を確立する。

1.5. 本論文の構成

第 1 章では、モンテカルロ計算の高速化に対する需要と、近年の計算機技術の進歩の傾向について述べた。第 2 章ではモンテカルロ法を用いた炉心計算の手法について解説する。第 3 章では並列計算の基礎的概念と GPU プログラミングについて述べる。第 4 章では GPU 上でのモンテカルロ法の実装方法について述べ、第 5 章で検証計算の結果を示す。最後に、第 6 章で結論を示す。

第2章 モンテカルロ法による中性子輸送計算

2.1. 本章の概要

本研究ではモンテカルロ法による炉心計算の高速化を目的としているが、本章において一般的な CPU 上におけるモンテカルロ法による炉心計算の実装方法について説明する。中性子追跡においては、モンテカルロ法にはアナログモンテカルロ法と非アナログモンテカルロ法の 2 つが存在するが、この 2 つについてもそれぞれ説明する[5][6]。

2.2. 中性子ランダムウォーク

本節では、多群法を想定したアナログモンテカルロ法における中性子ランダムウォークの実際の手順について説明する。アナログモンテカルロ法では、1 つの中性子の起こす物理現象を忠実に模擬する。

モンテカルロ法の基本は乱数にある。ここで、0 から 1 の一様乱数 ξ を導入する。この乱数は乱数ジェネレータによって生成された擬似乱数である。

2.2.1. 衝突点サンプリング

中性子の追跡計算は、まず初期中性子源から出発する。初期中性子源の配置は問題の入力として与えられ、ある 1 点に設定されるか、ある領域から一様にランダムにサンプリングされることが多い。

中性子源から出発する中性子の飛行方向(方位角ϕ、極角θ)、飛行距離 L は、乱数 ξ を用いての式(2.1)～(2.3)のようにサンプリングされる。中性子ランダムウォークの行われる体系において、全断面積はΣt、散乱断面積はΣt、吸収断面積はΣa、核分裂断面積はΣf とする。

$$\phi = 2\pi \times \xi \tag{2.1}$$

$$\cos\theta = 2 \times \xi - 1 \tag{2.2}$$

$$L = \frac{-\ln(\xi)}{\Sigma_t} \tag{2.3}$$

アナログモンテカルロ法の場合は、衝突点において吸収、散乱反応のどちらの反応になるかについても乱数ξを振ることで決定する。

$$0 \le \xi < \frac{\Sigma_a}{\Sigma_t} \quad \rightarrow \text{ 吸収}$$

$$\frac{\Sigma_a}{\Sigma_t} \leq \xi < 1 \quad \rightarrow \text{散乱}$$

吸収反応が起こった領域において、核分裂断面積が 0 でなければΣf /Σa の確率で核分裂反応を起こし、中性子を 2〜3 個放出する。放出数の期待値は、1 核分裂あたりの中性子発生数であるν値によって決定される。核分裂を起こすか、捕獲だけなのかに関係なく、反応を起こした中性子そのものは消滅し、追跡終了となる。

散乱反応が起こった場合、中性子の飛行方向を変えて追跡を続ける。等方散乱のみを想定した場合、新しい飛行方向は式(2.1)と(2.2)を用いてサンプリングする。2 群以上の計算では、散乱反応において中性子のエネルギー群が変化する可能性があり、このエネルギーの変化も散乱マトリックスに基づいて乱数により決定する。

2.2.1. 多領域体系

衝突点のサンプリングには断面積を用いるが、多領域で複数のマテリアルが存在する体系では断面積は必ずしも一定ではない。このため、中性子の追跡計算において、領域を跨ぐような衝突点がサンプリングされた場合、領域境界から飛程を再計算する必要がある。例えば、Fig. 2.1 のような A、B の 2 種類の領域から成る体系があるとし、それぞれの全断面積はΣtA、ΣtB とする。領域 A から出発した中性子はΣtA という全断面積に基づいて飛程をサンプリングする。衝突点位置が領域境界を越えて領域 B になった場合、領域と飛程の交点を求め、交点からはΣtB という断面積に基づいて改めて飛程を再計算する。

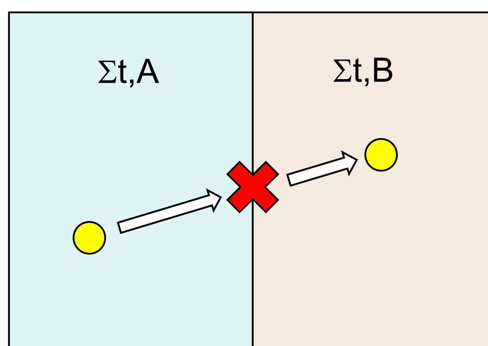

Fig. 2.1 複数領域に跨った追跡

つまり、モンテカルロ計算では線と面の交点を求める処理が頻繁に行われることになる。境界面が複数ある場合、どの境界面と最初に交わるかも判定しなければならない。3 次元で複雑な幾何形状を扱う場合、これは計算コストが増大しやすい処理になる。

2.2.2. 境界条件

炉心計算では、計算体系に境界条件を設定することがある。以下では、真空境界条件と完全反射境界条件をモンテカルロ法で設定する方法を述べる。

・真空境界

真空境界では、その境界から漏れ出した中性子が戻ってこない。モンテカルロ法では、この現象をそのまま模擬する。具体的には、中性子の飛程が真空境界を超えた場合、その中性子は消滅したものとして追跡を終了する。

・完全反射境界

完全反射境界では、中性子は鏡面に反射されたのと同じように、エネルギーはそのままで速度ベクトルを境界面に対して反転させる(Fig. 2.2)。

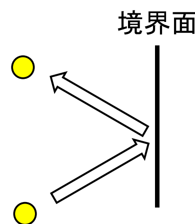

Fig. 2.2 完全反射境界の概念図

2.3. 世代と固有値計算

核分裂源が既知のものとして与えられる固定源計算では、初期中性子源から発生した中性子のランダムウォークを行い、その結果を統計処理して計算結果を得る。しかし固有値計算において、中性子源分布は未知であり、求める必要のあるパラメータである。

モンテカルロ計算では、固有値計算には世代と呼ばれる概念を導入する。1 世代とは、中性子源から得られた中性子が、吸収や漏れによって消滅するまでを指す。中性子ランダムウォークの過程で発生した中性子は、次の世代の中性子源(核分裂源)となる。

モンテカルロ法による固有値計算は次のステップにより行われる。

1. 1 世代あたりに発生させる中性子数(バッチサイズ)N と、世代の数 L を決定する。
2. 1 世代目の初期中性子源分布を与える。
3. 中性子源から N 個の中性子を発生させる。
4. N 個の中性子について、ランダムウォークを行う。核分裂反応が起こった場合には、その位置と発生中性子数を記録する。
5. ステップ 4 で求められた核分裂反応を集計する。核分裂によって生じた中性子数を合計 M とすると、この世代における実効増倍率 k_i は $k_i=M / N$ となる。

6. 発生した M 個の中性子から、N 個の中性子をサンプリングする。$M < N$ である場合には重複を許す。この処理は、各世代で追跡を行う中性子数を等しくするための処理である。さもなければ、$k_{\mathrm{eff}} < 1$ の体系では追跡すべき中性子が際限なく減り、逆に $k_{\mathrm{eff}} > 1$ の体系では中性子が増加し、計算リソースが足りなくなってしまう。
7. サンプリングして得られた中性子を、次の世代の中性子源として設定し、ステップ 4 に戻る。
8. L 世代の計算が終了した場合、各世代で得られた k_i を平均して最終的な実効増倍率とする。

1 世代目の初期中性子源は入力として与えられるが、世代を経るごとに真の核分裂源分布に近づいていく。逆に、収束する前の世代は真の核分裂源と異なる分布になるため、最終的な結果を得るための計算からは除外する。1～J 世代目の計算結果を捨てる場合、最終的な実効増倍率 k_{eff} とその標準偏差 σ_{keff} は式(2.4)、(2.5)で与えられる。ただし式(2.5)は、各世代の計算結果が互いに相関を持たないと仮定しており、相関を考慮した厳密な分散とは異なる[7]。

$$k_{eff} = \frac{1}{L-J}\sum_{i=J+1}^{L} k_i \tag{2.4}$$

$$\sigma_{keff} = \frac{1}{\sqrt{L-J}}\sqrt{\frac{1}{L-J-1}\sum_{i=J+1}^{L}(k_i - k_{eff})^2} \tag{2.5}$$

2.4. 非アナログモンテカルロ法

これまでにアナログモンテカルロ法の説明をしてきたが、実際の実用的なモンテカルロ計算コードではほとんどアナログモンテカルロ法を用いられておらず、非アナログモンテカルロ法がもっぱら用いられている。アナログモンテカルロ法では、中性子の挙動を忠実にシミュレーションするため、中性子の状態は存在するかしないかのどちらかしかない。そして中性子が衝突反応を起こした場合、散乱か吸収かで反応が明確に分かれてしまう。これは、計算結果の分散が大きくなるという欠点がある。例えば、中性子が低確率で届く領域があるとする。このような領域は、遮蔽計算などで中性子源から離れた領域に相当する。ほとんどの中性子はこの領域に到達するまでに吸収反応を起こし消滅するとし、低確率で生き残って到達した中性子がこの領域の反応に寄与する。すると、たまたま領域内で中性子が反応を起こした時と、起こさなかったときのばらつきが大きい。

非アナログモンテカルロ法では、ウェイト(重み)という概念を導入することで分散を小さくしている。

2.4.1. ウェイトの導入

ウェイトを導入することで、中性子が存在するかしないかの 2 択ではなく、中間の状態を表現できるようになる。ウェイトは中性子の存在確率を表し、吸収反応は中性子が消滅させるのではなく、ウェイトを減少させることによって表現する。

ウェイトを導入した非アナログモンテカルロにおける中性子追跡は、例として次のような処理になる。

1. 追跡開始時の中性子は W(ウェイトの値)を 1 に設定する。アナログモンテカルロと同様に飛行方向、飛程をサンプリングし、衝突点を求める。
2. 衝突位置が決定したら、中性子のウェイトを散乱反応の確率に応じて減らす。つまり、式(2.6)によってウェイト W を W'に更新する。ウェイトを減らすことで吸収反応を表現しているため、実装上は衝突点で散乱反応と吸収反応の区別はない。つまり、乱数で散乱反応か吸収反応かを振り分けることはない。

$$W' = \frac{\Sigma_t - \Sigma_a}{\Sigma_t} W \tag{2.6}$$

3. 衝突点において、核分裂が発生するかを判定する。発生する中性子の数 n は式(2.7)により判定する。

$$n = \mathrm{int}\left(\frac{\nu\Sigma_f}{\Sigma_t} \times \alpha \times W + \xi \right) \tag{2.7}$$

ここで、int()は小数点以下を切り捨てて整数にする関数である。この式を評価するのに用いるウェイトは衝突によって更新されたウェイト W'でなく、更新前のウェイト W である。アナログモンテカルロ法で、無限体系におかれた 1 個の中性子が核分裂によって生み出す中性子の期待値は$\nu\Sigma f/\Sigma a$ である。式(2.7)の積の項$\nu\Sigma f/\Sigma t$ はこの期待値を保存する。非アナログモンテカルロ法では、W は衝突の度に$\Sigma s/\Sigma t$ 倍になるので、全衝突について式(2.7)のように和をとると期待値が保存されていることが分かる。ただし、n 回目の衝突後のウェイトを W_n とし、初めのウェイト W_0 は 1 とする。

$$\sum_{n=0}^{\infty}\frac{\nu\Sigma_f}{\Sigma_t}W_n = \frac{\nu\Sigma_f}{\Sigma_t}\sum_{n=0}^{\infty}\left(\frac{\Sigma_s}{\Sigma_t}\right)^n = \frac{\nu\Sigma_f}{\Sigma_t}\frac{\Sigma_t}{\Sigma_t - \Sigma_s} = \frac{\nu\Sigma_f}{\Sigma_a} \tag{2.8}$$

ξとの和をとってから整数に切り捨てする処理により、積の項で計算された期待値で核分裂による中性子の発生が判定される。たとえば、int(0.4+ξ)なら、ξが 0.0～0.6 の範囲(確率 0.6)で中性子が発生せず、ξが 0.6～1.0 の範囲(確率 0.4)で中性子が発生する。

αは世代ごとに発生する中性子数が大体等しくなるように補正するための係数で、多くの場合、前世代の固有値の逆数を用いる。k_{eff} が 1 より小さい体系では、次の世代の核分裂源を選ぶときに重複して中性子を選ぶことになる。また 1 より大きければ、使われない核分裂源を多く記録する必要がある。この補正値を入れることにより、次の世代に必要な核分裂源を選ぶのに適した数の核分裂が発生するように調整することが出来る。例えば k_{eff} が 0.5 であれば、α を 1/0.5=2 に設定することで毎世代の中性子の発生数が大体等しくなる。

4. 次の衝突点をサンプリングし、ステップ 2、3 を繰り返す。この処理は、次節で示すロシアンルーレットか、体系からの漏れにより消滅するまで繰り返す。

以上のステップを、フローチャートにした場合 Fig. 2.3 のようになる。

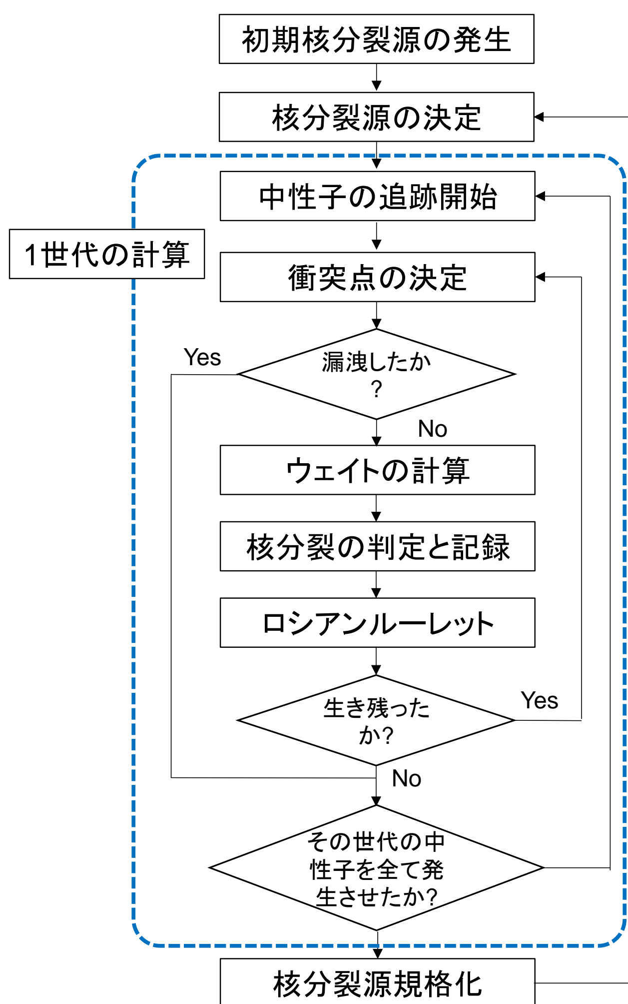

Fig. 2.3 非アナログ計算のフローチャート

非アナログ計算では、吸収反応によって中性子が失われることはないが、ロシアンルーレットによって追跡を終了させる。ロシアンルーレットについては次節において解説する。

2.4.2. ロシアンルーレット

ウェイトを導入した場合、分散を低減することはできるが、中性子が吸収反応によって消滅することがなくなるため、結果的に計算時間の増大をもたらす。ウェイトはどのような値でも計算コストには影響を与えないが、非常に小さいウェイトの中性子の追跡を続けても計算結果にはあまり寄与しない。そのため、ウェイトが十分に小さくなった場合には、その中性子の追跡を終了するほうが望ましい。そのためのアルゴリズムがロシアンルーレットである。

ロシアンルーレットを適用する場合には、中性子の最小ウェイト W_{kill} を設定する。そして、中性子のウェイトが W_{kill} 以下になった場合、ある確率で中性子を消し(kill)、生き残った場合には新しいウェイト $W_{\mathrm{survive}}(>W_{\mathrm{kill}})$を与える。これがロシアンルーレットである。

ロシアンルーレットを行う場合でも、全体的なウェイトの期待値を保存しなければならない。そのため、ロシアンルーレットで中性子が生き残る確率は W / W_{survive} に設定される。これにより、全体的な期待値は式(2.9)のように W のまま保存される。

$$\left(\frac{W}{W_{survive}}\right)W_{survive}+\left(1-\frac{W}{W_{survive}}\right)\times 0=W \tag{2.9}$$

2.4.3. エスティメータ

中性子のランダムウォークを行って物理現象を模擬するだけでは、実際に欲しい計算結果は得られない。そのため、ランダムウォークの過程において、物理現象への寄与を加算していくことで実際の物理量を求める。この手法をエスティメータと呼ぶ。

実効増倍率を求めることは、ある世代の中性子の消滅と生成の比をとることで求められるが、実際にはエスティメータを用いて求めることが多い。例えば、衝突反応が起こるたびに加算される衝突エスティメータでは、ある世代の実効増倍率 k は式(2.10)によって求められる。

$$k=\frac{1}{N}\sum_{i}\frac{\nu_i\Sigma_{f,i}}{\Sigma_{t,i}}W_i \tag{2.10}$$

i はその世代での全ての衝突反応についてのインデックスである。$\Sigma_{t,i}$、$\Sigma_{t,i}$、ν_i はその衝突点における断面積、W_i は衝突前にもっていた中性子のウェイトである。この式は、増倍率の定義通りに元の中性子と生成された中性子の比をとっている。N が元の中性子の数であり、Σの項が生成された中性子の数である。

中性子束を衝突エスティメータによって求める場合、中性子束を求めたい領域ごとに分割する。その領域の体積を V とすると、式(2.11)により中性子束が計算できる。

$$\phi=\sum_{i}\frac{W_i}{\Sigma_{t,i}V} \tag{2.11}$$

式(2.11)は式(2.10)と同様にその世代全ての衝突反応について加算を行う。これは中性子束を求めるための衝突エスティメータである。

衝突エスティメータは、衝突反応が起こらなければ加算されないので、全断面積の小さいボイド領域などでは物理量の評価が行えない。そのため、モンテカルロ計算コードには衝突エスティメータ以外のエスティメータも導入されている。例えば、空間中の中性子の移動距離に応じて寄与を加算する飛程長エスティメータ、境界面を通過した中性子の入射角に応じて寄与を加算する面交差エスティメータが存在する。

2.5. 仮想散乱による追跡計算

2.2.1 節で述べたように、モンテカルロ法で複数の領域が存在する体系を扱う場合、中性子の飛行が領域を跨ぐかどうかの判定を行う必要がある。中性子の飛行距離は全断面積によって決定されるが、全断面積は領域によって異なるため、衝突点を正しく判定するためには領域を跨ぐ度に参照する全断面積を変更しなければならない。また、衝突点で起こる反応を決定する過程においても、その衝突点における反応断面積が必要となるため、中性子がどの領域を飛行しているかの判定が必要となる。中性子の飛行は直線で表され、領域の境界は面となるため、領域の判定には線と面の交点を求める処理が行われる。

この処理は計算機にとっては負担であり、これを回避するための手法が仮想散乱を用いた追跡手法であり、Delta-tracking method とも呼ばれる[8]。

2.5.1. 仮想散乱断面積

仮想散乱を用いる場合、(2.12)式に示すように、体系の各マテリアルの全断面積に仮想散乱断面積と呼ばれる補正値を加える。

$$\Sigma_{t,g}^{*} = \Sigma_{t,g}(r) + \Sigma_{s,g}^{*}(r) \tag{2.12}$$

$\Sigma_{t,g}(r)$:位置 r、エネルギーg 群の全断面積

$\Sigma_{s,g}^{*}(r)$:位置 r、エネルギーg 群の仮想散乱断面積

$\Sigma_{t,g}^{*}$:体系内一定の、エネルギーg 群の仮想散乱を含む全断面積

仮想散乱は中性子のエネルギー、飛行方向を変えないため任意の値に設定可能であり、体系で最大の全断面積にあわせるように、仮想散乱あり全断面積を設定する。衝突点では $\Sigma_s^{*}/\Sigma_t^{*}$ の確率で仮想散乱反応が起こり、仮想散乱反応が起こった場合、粒子は向き・エネルギーを変えない散乱を起こしたものとする。これは、Σ_s^{*} の値は任意の値に設定できるということを意味する。なぜなら、中性子の追跡結果は仮想散乱反応によって影響されないためである。

そこで、式(2.13)、(2.14)に示すように、Σ_t^*の値を全てのマテリアルの全断面積の最大値とし、Σ_s^*はこの条件を満たすように設定する。

$$\Sigma_{t,g}^* = \max(\Sigma_{t,g}^1, \Sigma_{t,g}^2, ..., \Sigma_{t,g}^N) \tag{2.13}$$

$$\Sigma_{s,g}^{*,n} = \Sigma_{t,g}^* - \Sigma_{t,g}^n \tag{2.14}$$

$\Sigma_{t,g}^1, ..., \Sigma_{t,g}^N$ ：体系に存在する各マテリアル(1～N)の全断面積

$\Sigma_{s,g}^{*,n}$ ：マテリアル番号 n、エネルギー群 g の仮想散乱断面積

$\Sigma_{t,g}^n$ ：マテリアル番号 n、エネルギー群 g の全断面積

これにより、体系内のあるエネルギー群 g についての全断面積は$\Sigma_{t,g}^*$で一定となる。体系内で全断面積が一様であるため、領域を跨ぐときに全断面積を更新する必要が無くなる。また、中性子の追跡に必要な線と面の交点の判定が、衝突点でのマテリアルの判定をするだけになり、処理は大幅に簡略化される。これにより負荷を軽減することが出来る。

仮想散乱を用いる場合は、中性子追跡の途中に仮想散乱反応が混じることになる。結果として、全体での衝突回数は増えることになる。特に、もともと全断面積が比較的小さい領域では仮想散乱が多く起きることになる。体系の一部に全断面積が大きい領域が含まれている場合、他の領域で仮想散乱が増える。そのため、全断面積の領域ごとの差が大きい体系では、仮想散乱反応の増大によって計算コストが増えることになる。

2.5.2. 衝突エスティメータ

仮想散乱を導入した場合、計算上の全断面積は増加するため衝突反応の数も増加する。そのため、衝突エスティメータは期待値を保存するように変形する必要がある。全断面積Σ_tがΣ_t^*に増加するため、1 単位の距離を移動する間に起こる衝突の頻度はΣ_t^*/Σ_t倍になる。このため期待値を保存するには、衝突エスティメータへ加算する前に、Σ_t^*/Σ_tの逆数Σ_t/Σ_t^*を掛ければ良い。

そのため、式(2.10)と式(2.11)の各エスティメータはそれぞれ式(2.15)、(2.16)に変形される。

$$k = \frac{1}{N}\sum_i \frac{\nu_i \Sigma_{f,i}}{\Sigma^*_{t,g}} W_i \tag{2.15}$$

$$\phi = \sum_i \frac{W_i}{\Sigma^*_{t,i} V} \tag{2.16}$$

また、核分裂発生についてのサンプリングも、式(2.7)から式(2.17)のように変形される。

$$n = \mathrm{int}\left(\frac{\nu\Sigma_f}{\Sigma^*_t} \times \alpha \times W + \xi \right) \tag{2.17}$$

式(2.15)から(2.17)を見てわかるように、マテリアルごとの全断面積が仮想散乱あり全断面積に置換されている。

仮想散乱を用いた場合、元々全断面積の小さい領域でも衝突反応が他領域と同程度に起こるようになるため、真空領域の中性子束も衝突エスティメータによって評価することが出来る。衝突回数が増えることは計算コスト上不利だが、逆に衝突エスティメータの利用にとっては利点となる。ただし、領域境界を明示的に扱わないため、領域境界との交点の計算が必要となる飛程長エスティメータや面交差エスティメータを実装する上では不利である。これらのエスティメータを計算するのに必要な中性子と領域境界との交点は、本来追跡途中に計算されているが、仮想散乱を用いる場合は別途計算の必要があるためである。

2.5.3. 完全反射境界条件

完全反射境界条件は、モンテカルロ計算では中性子の速度ベクトルを境界に対して反転させることで取り扱う。これは厳密な方法であるが、仮想散乱を用いた追跡計算を行う場合には問題が生じる。中性子が完全反射境界と交差する点から速度ベクトルを反転させなければならないので、領域境界を明示的に取り扱わない仮想散乱の場合、交点座標の計算による追加の計算コストが発生することになる。つまりは、仮想散乱の導入によって不要になった、線と面の交点の計算が必要になってしまう。しかし、座標変換を用いることで、境界面との交点を明示的に取り扱わずに完全反射境界を表現できる。

座標変換を用いた場合の完全反射境界の取り扱いについて説明する。簡単のため、x 座標についての 1 次元体系について考える。中性子追跡計算において得られた衝突点の座標 x から、完全反射を考慮して行き着いた先の座標を x'とする。長さ l の 1 次元体系の場合で、原

点が真空境界、$x=l$ が完全反射境界条件だとする。するとに示すように、この体系は $x=l$ 以降、$x=l$ について線対称な体系が存在する、長さ $2l$ の 1 次元体系と同じとみなせる。

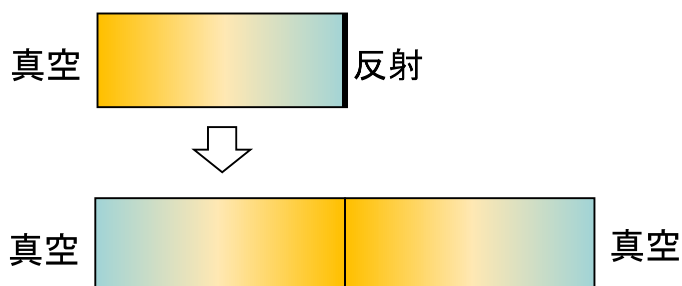

Fig. 2.4　片方が反射境界の場合の体系

この考え方から、仮想散乱を用いて全領域が均質だとして求めた衝突点 x を、反射を考慮した衝突点 x'に変換することができる。x と x'の変換をグラフ化すると Fig. 2.5 のようになる。

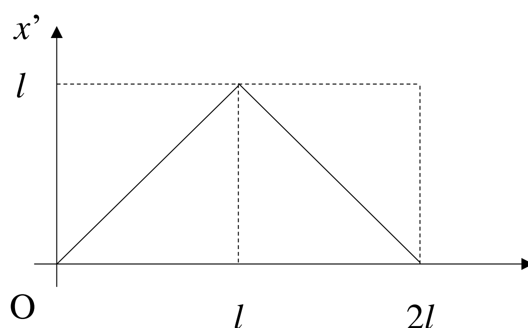

Fig. 2.5　衝突点の変換関数のグラフ

数式で表すと、式(2.18)になる。

$$x'=|x-l|+l \tag{2.18}$$

また、反射による速度ベクトルの反転についても考慮しなければならない。Fig. 2.6 のように、衝突点 x が 1 以上の場所にあれば、1 回反射境界を通過しているので、座標変換と同時に速度ベクトルを反転させてやらなければならない。

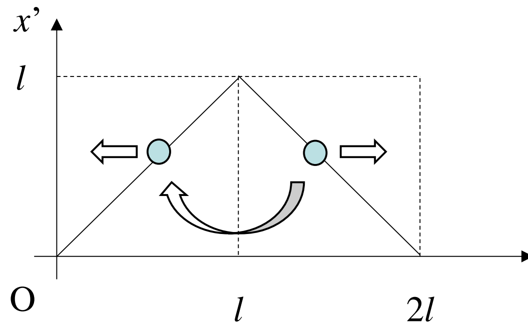

Fig. 2.6　速度ベクトルの反転

また、長さ l の 1 次元両端完全反射体系の場合でも同じように座標変換が行える。両端反射の場合は、x と x'の変換をグラフ化すると Fig. 2.7 のようになる。

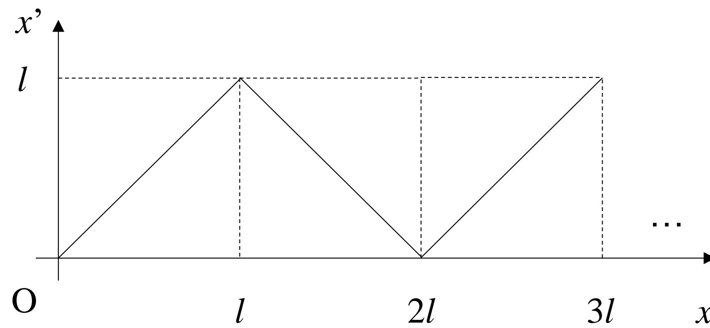

Fig. 2.7　両端反射体系の場合の座標変換

つまり、x と x'の関係式は式(2.19)になる。

$$\begin{cases} x' = x \bmod l \quad ([x/l]\text{が偶}) \\ x' = l - x \bmod l \,([x/l]\text{が奇}) \end{cases} \tag{2.19}$$

また、$[x/l]$が偶なら速度ベクトルは変化しないが、奇なら速度ベクトルの反転を行わなければならない。$[x / l]$が奇とは、つまり奇数回反射境界を通過したということであり、奇数回速度ベクトルの反転が行われたのと同じで、元の速度ベクトルから反転している。

完全反射条件だけでなく、周期境界についても座標変換を行うことが出来る。周期境界から漏れた中性子は、対応する別の境界から速度ベクトルを保持したまま発生する。このような条件は、同じ体系を繰り返し配置したい場合に用いられる。1 次元体系周期境界条件での x と x' の関係式は式(2.20)になる。

$$x' = x \bmod l \tag{2.20}$$

上記では 1 次元の場合について論じたが、多次元の場合もそれぞれの次元について独立にこの座標変換の方法を適用できる。体系全体が直方体であれば、各面に任意に真空境界と完全反射境界、周期境界を設定することが出来る。

2.6. 本章のまとめ

本章では、中性子輸送モンテカルロコードを実装する上で、必要な概念と実装方法について説明を行った。2.2 節では、アナログモンテカルロ法に基づく中性子ランダムウォークの方法について述べた。2.3 節では、実効増倍率を求めるのに必要な固有値計算の概念と実装について述べた。2.4 節では、実用的なモンテカルロ計算コードで採用されている、非アナログモンテカルロについて説明した。ウェイトやロシアンルーレット、エスティメータなどの概念を述べた。2.5 節では、本研究でも採用している仮想散乱を用いた幾何形状簡略化の方法について説明した。仮想散乱を用いる場合、中性子と領域境界の交点を明示的に計算する必要が無くなり、幾何形状の取り扱いを比較的簡単に実装出来るようになる。また、仮想散乱を用いた場合は衝突回数が増えるため、エスティメータの計算方法を修正する必要があり、その方法についても説明を行った。さらに、完全反射境界条件をより簡易に取り扱う方法として、座標変換を行うことで反射境界と中性子の交点を求める計算を不要にする方法について述べた。

第3章 並列計算と GPU

3.1. 本章の概要

本章では、GPU の構造と、並列計算を実際に行う手法について述べる。その後、本研究で利用する OpenCL の具体的な説明を行い、実際の計算コードを実装する上での方針を述べる。

3.2. GPU の構造と特徴

GPU は本来、画像処理のための演算装置である。多くの計算機では、ビデオカードと呼ばれる拡張ボード上に搭載されている。そのため、PCIe と呼ばれる入出力インターフェイスを通じて CPU と GPU の間でデータをやり取りする。

GPU のアーキテクチャの概念図を Fig. 3.1 に示す。

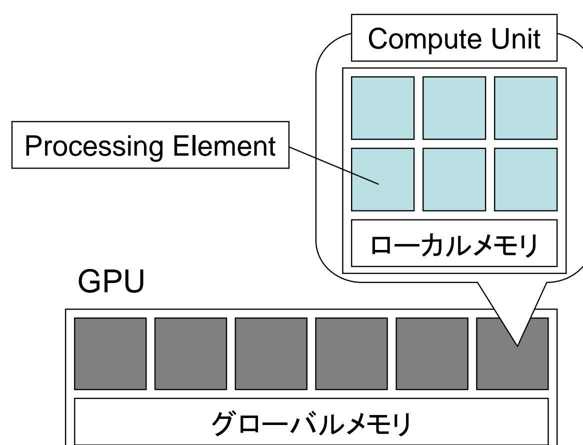

Fig. 3.1 GPU の構造

GPU デバイスは複数の Compute Unit(CU)を持ち、各 CU は複数の Processing Element(PE)から構成されている。各 PE が並列に計算を実行することで並列計算が実現されている。本研究で検証に利用した Radeon HD7950 は、CU は 28 個持っている。PE は 1CU あたり 64 個あり、全体で 1792 個となる[9]。GPU には、全ての CU から利用できるグローバルメモリと、また各 CU で共有されるローカルメモリが存在する。

高機能な演算用のコア 1～数個から構成される CPU に対し、GPU は並列計算に特化した構造になっている。このため、並列計算に適したアルゴリズムにおいては、GPU は CPU を超える性能を発揮する。計算機の性能を示す指標として FLOPS(FLoting point Operations Per Second)があり、これは 1 秒間に浮動小数点の演算が何回行えるかを示している。Table 3.1 では、CPU と GPU の単精度浮動小数点の FLOPS を比較して示している。

Table 3.1 CPU と GPU の FLOPS 比較

種類	CPU	GPU	GPU
名称	Intel Core i7-3770 3.40GHz(4Core 8Thread)	Radeon HD7950 800MHz	GeForce GTX 960 1127MHz
FLOPS	250GFLOPS[10]	2.87TFLOPS[9]	2.3TFLOPS[11]

Table 3.1 から分かるように、FLOPS で比較すれば GPU は CPU を上回っている。しかし、GPU 用に最適化されていないアルゴリズムの場合には、CPU の性能を下回ることもある。GPU の演算コアは CPU に比べると各機能を省いた単純な構造になっており、1 コアあたりの性能は大きく劣る。例えば、CPU には分岐予測機能があるが GPU には存在しない。そのため複雑な分岐が存在したり、PE ごとに行う処理が大きく異なる場合には大きな性能低下をもたらすことになる。

3.3. 並列計算

3.3.1. 並列化の基礎的概念

前節で述べたように、GPU の演算能力を引き出すためには並列計算を行う必要がある。本節では、並列計算の基礎的概念を述べる。

あるプログラムで、処理全体を複数の部分に分割することを考える。この分割された処理をタスクと呼ぶ。ここでは、処理を 3 つのタスク 1～3 に分割したとする。タスク 1～3 の処理を実行することを考えると、並列計算を行わない場合、タスク 1 が終わればタスク 2 の処理に移り、タスク 2 が終わればタスク 3 の処理に移る。並列計算を行った場合、タスク 1～3 の処理を同時に実行することになる。Fig. 3.2 に、並列計算の概念図を示す。

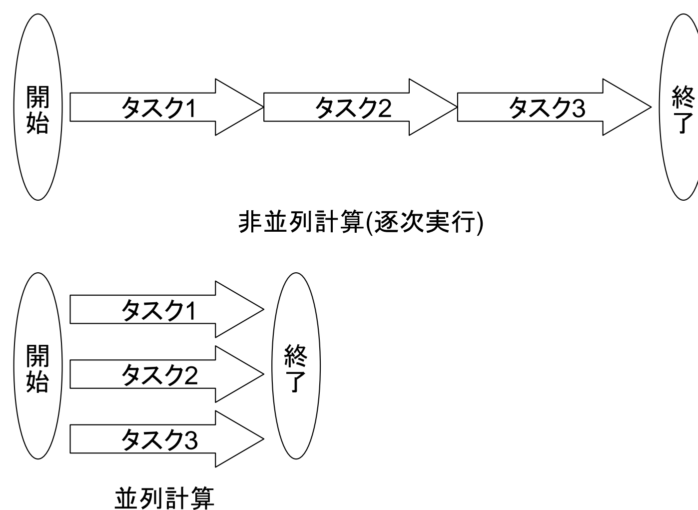

Fig. 3.2 タスクの並列化

並列計算では、これらのタスクを異なるプロセッサに割り当てることで、複数のタスクを同時に実行する。これにより、並列計算を行わない場合に比べて計算時間が短縮できる。ただし、並列計算を行うにはそれぞれの計算処理が、並列化したい他の計算処理の結果に依存しないことが必須条件となる。例えば、Fig. 3.2 でタスク 2 の開始にタスク 1 の計算結果が必要な場合、タスク 1 の終了までタスク 2 が開始できない。この場合、並列計算が行えない。このため、並列計算を行ってどれだけ計算時間が短縮できるかは、どれだけプログラム上で並列計算化が可能な部分を増やすかが重要となる。

また、分割されたタスクは実行時間が均等になるよう処理を配分しなければならない。これをロードバランスと言う。例えば、Fig. 3.2 でタスク 1、2 の処理が早く終了し、タスク 3 の処理に時間がかかる場合、タスク 3 の処理時間のために全体の処理時間が長くかかってしまう。また、タスク間のデータ共有や通信が速度面で障害になり得る。タスクを異なる計算機で分散させ、互いにネットワークでデータを送受信する場合など、計算機の性能がネットワークの速度によって制限される可能性がある。

なお、並列計算 (Parallel computing) とよく似た概念として、並行計算 (Concurrent computing) というものが存在する。並列計算は、タスクを分割しハードウェア上で分散して実行することで、計算速度を短縮することを目的とする。計算速度に関わりなく、一般的に分散したタスク間の相互作用を取り扱う場合には並行計算という概念が用いられる。本論文では、計算時間の短縮を目的とするため、一貫して並列計算という用語を用いている。

3.3.2. 並列化プログラミングモデル

並列化を行うためのプログラミングモデルは複数考案されており、共有メモリモデル、スレッドモデル、メッセージパッシングモデル、データ並列モデルなどがある[12]。本節ではこれらのプログラミングモデルそれぞれについて説明する。

(1) 共有メモリモデル

Fig. 3.3 に示すようにタスクがアドレス空間を共有し、メモリには非同期アクセスが行われる。タスク間の通信は明示的に行われず、メモリの読み書きだけで完結するため、単純に実装することができる。共有メモリにアクセスするより、タスク毎に独立したメモリにデータをキャッシュする方が一般的に高速であり、最適化においてはデータのキャッシュを意識する必要がある。また、後述するメモリアクセスの競合を避けるため、ロック等の同期処理が必要になる。

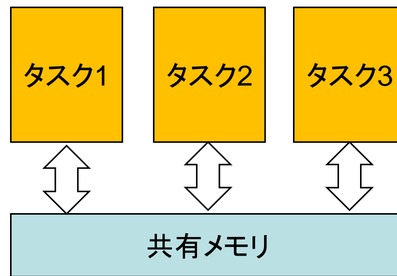

Fig. 3.3 共有メモリモデル

(2) スレッドモデル

Fig. 3.4 に示すように、1 つのプロセスが複数の実行フローを持つ。この複数の実行フローはスレッドと呼ばれる。スレッドはプロセス全体のメモリ領域を共有することもでき、スレッド固有のメモリ領域を持つこともできる。スレッドはそれぞれ異なるタスクを処理する。スレッドは、プロセス上で実行されるサブルーチンとして記述される。

スレッド生成などの処理の実装は、OS 等の機能として用意されていることが多い。例として UNIX 標準で定められている POSIX スレッドが存在する。

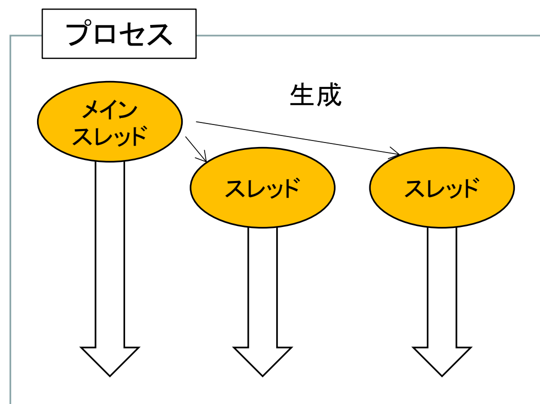

Fig. 3.4 スレッドモデル

(3) メッセージパッシングモデル

分割されたタスクは、それぞれ固有のメモリ領域を持っている。タスク間のデータ共有は、あるタスクから別のタスクへメッセージを送信することで行われる。

それぞれのタスクは、物理的に同じ計算機で処理されることあり、異なる複数の計算機で処理されることもある。複数の計算機で分散させる場合、計算機同士をネットワークで接続し、ネットワークを介してメッセージの送受信を行う。

このモデルでは、例えば Fig. 3.5 に示すように、タスク 1 からタスク 2 にデータを送信し、それぞれのタスクで異なる処理を行う。タスク 1 側において、タスク 2 の処理結果が

必要になったところでタスク 2 側の計算結果をメッセージにより受け取る。このとき、タスク 2 の処理が終了していなければ、メッセージが受信されるまで待機する。

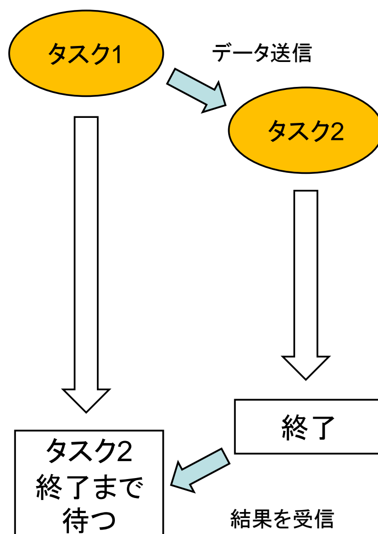

Fig. 3.5 メッセージパッシングモデル

(4) データ並列モデル

データ並列では、処理すべきデータを分割することで並列計算を実現する。Fig. 3.6 に示すように、配列であればインデックスの値に基づきタスクが分割される。

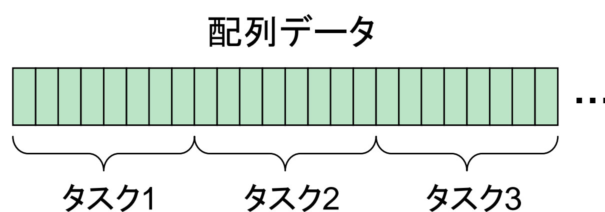

Fig. 3.6 データ並列モデル

分割されたタスクが並列して処理されることで、並列計算を行うことができる。

データ並列の分かりやすい例が、for ループによる配列演算である。例えば、長さ 100 の配列の要素それぞれを足し合わせたいとする。

```
int a[100], b[100], c[100];
…
for(int i = 0; i < 100; i++){
    c[i] = a[i] + b[i];
}
```

プロセッサが 2 つ存在する計算機であれば、for ループを 0〜49 と 50〜99 の 2 つに分割し、独立して計算を行える。データ並列では、処理するデータの範囲が異なるだけで処理内容は同じである。上の例では、足し算を行うという処理は変わらない。GPU は構造上、異なるデータに対し同じ処理を行うのに適しているため、データ並列モデルは GPU による並列計算に当てはまるモデルである。

3.3.3. アムダールの法則

並列計算の適用によりどれだけ性能が向上するかを、理論的に示したものがアムダールの法則である[12]。あるプログラムについて並列計算で動作する部分と、並列計算を行わない部分に分割する。並列計算を行う部分の割合を a とすると、行わない部分の割合は 1- a である。よって、プログラム全体の実行時間を T とするならば、T は式(3.1)のように表される。

$$T = Ta + T(1-a) \tag{3.1}$$

ここで、n 個のプロセッサによって並列計算を行った場合の計算時間を T_n とすると、並列計算が適用できる処理に必要な時間は $1/n$ になるため、T_n は式(3.2)で表される。

$$T_n = \frac{T_1 a}{n} + T_1(1-a) \tag{3.2}$$

プロセッサコア 1 つの場合に対する性能向上率 T_1 / T_n は、式(3.3)のようになる。

$$\frac{T_1}{T_n} = \frac{T_1}{\frac{T_1 a}{n} + T_1(1-a)} = \frac{1}{\frac{a}{n} + (1-a)} \tag{3.3}$$

式(3.3)が示しているのは、プロセッサ数に対する性能の向上は線形より小さいことである。例えば、プロセッサを大量に(無限に)用意した場合の性能向上率 T_∞ は式(3.4)のようになる。

$$T_\infty = \lim_{n \to \infty} \frac{T_1}{T_n} = \frac{1}{1-a} \tag{3.4}$$

プログラムのうち並列化可能な部分が 50%であれば、どれだけプロセッサ数を増やしても性能は最大 2 倍にしかならない。

以上が並列化に関するアムダールの法則である。この法則から、並列化ができない逐次実行の部分を極力減らすことが性能向上に重要であることがわかる。並列計算が向いていない処理に関しても、なるべく多くのコアを処理に使えるように考慮する。例えば、積算処理を複数コアに分散させるなどである。

3.3.4. メモリ競合と競合状態

並列計算において、同一のメモリ領域に複数のプロセッサから書き込みを行うとデータの整合性が破綻することがある。これがメモリの競合(Data race)である。

例えば、加算代入を行う場合にメモリ競合が問題になりやすい。これは、計算機上では加算代入が以下のステップで実行されているためである。

1. 加算代入の対象となるメモリ領域から、一時的なメモリ領域(レジスタ)にデータをコピーする。
2. レジスタ上のデータに加算処理を行う。
3. レジスタ上のデータを対象のメモリにコピーする。

この一連の処理の途中に、他のプロセッサが割り込むと演算が正しく行われない。例えば、プロセッサ A がメモリからレジスタにコピーした後、プロセッサ A が処理結果を書き戻す前にプロセッサ B がメモリにアクセスした場合、プロセッサ A による更新の前のデータを得ることになる。プロセッサ B がこのデータをもとにメモリをさらに更新する場合、結果はプロセッサ A と B が順番に処理したものとは異なってしまう。例えば Fig. 3.7 に示すように、メモリ上の 20 という数値に、プロセッサ A,B がそれぞれ 10 と 12 を書き込もうとする。この場合、それぞれのプロセッサがそれぞれの計算結果”30”と”32”を元のメモリ領域に書き込もうとする。この数値はいずれも正しい答えである”20+10+12=42”にはならない。

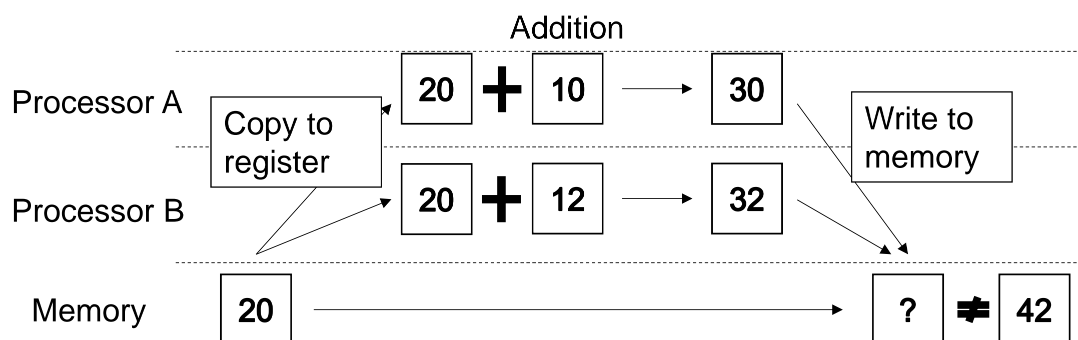

Fig. 3.7 データ競合の例

この結果はハードウェアの実装依存である。処理系は結果に何の保証もしない、つまりC 言語での不正なメモリアドレスへのアクセスなどと同様に、「処理系依存」の操作となる。

結果の整合性を保証するために、アトミック操作やロックによる同期処理が用いられる。アトミック操作は、これを実行する途中で他のプロセッサから割り込みができないことが保証される操作であり、ロックは共有資源へのアクセスを 1 つのプロセッサからのみに制限する。ただし、これらの処理は性能を大きく低下させる可能性があり、なるべく使わないことが望ましい。プロセッサごとのデータの独立性を高め、メモリ競合が起こらないように計算アルゴリズムを組むことが重要になる。

データ競合とよく似た概念に、競合状態(Race condition)がある。データ競合はハードウェア上の実装に起因する問題であるが、競合状態は処理するタイミングが各タスクで異なることに起因する。計算ロジックによっては、上記のアトミック操作やロックを用いても、処理のタイミングが異なることにより計算結果が実行の度に異なることが起きる。データ競合は起きなくても、処理のタイミングによって結果が不定になることを競合状態と呼ぶ。これを防ぐためには、適切なロックの利用や、計算結果の後処理が必要となる。

3.4. OpenCL の特徴

GPU を汎用計算に用いることを GPGPU[12]と言い、これは General Purpose computing on GPU の略である。GPU を汎用計算に用いるためには専用のフレームワークが必要となる。フレームワークとは、ある処理を実行するために必要な枠組みを提供するものを指す。本研究では、GPU を利用するためのフレームワークとして OpenCL[4]を用いる。OpenCL の特徴は、次の通りである。

- GPU だけでなく、CPU やその他の計算資源にも対応する。
- GPU で利用できるメモリ管理機能を提供する。
- GPU 側の処理は OpenCL C という C 言語のサブセットで記述し、ホスト(CPU)から API(Application Programming Interface、ライブラリ等への機能にアクセスするためのインターフェイス)を介して GPU を利用する。
- OpenGL と同じく、Khronos Group によってオープン標準として策定されている。

GPGPU の分野で先行していた技術である CUDA 等と比べると、GPU だけでなく並列処理に用いることのできるデバイスは全て統一的に扱えるよう設計されており、他プラットフォームへの移植が簡単になっている。GPU だけでなく、CPU やその他のコプロセッサ等が扱えるのは利点である。しかし OpenCL がこれらのハードウェアの差異を完全に吸収することは難しく、現状では計算速度を求める場合に人の手によるコードのハードウェアへの最適化はある程度必要になると思われる。

3.5. OpenCL プログラミングの基本

本節では、OpenCL を用いたプログラミング方法について説明する。

OpenCL では、CPU 上で実行されるプロセスをホストとし、ホストは GPU 等のデバイスを操作する。ホストがデバイスを操作することによって、GPU に計算を行わせることができる。OpenCL プログラミングにおけるフローチャートを Fig. 3.8 に示す。

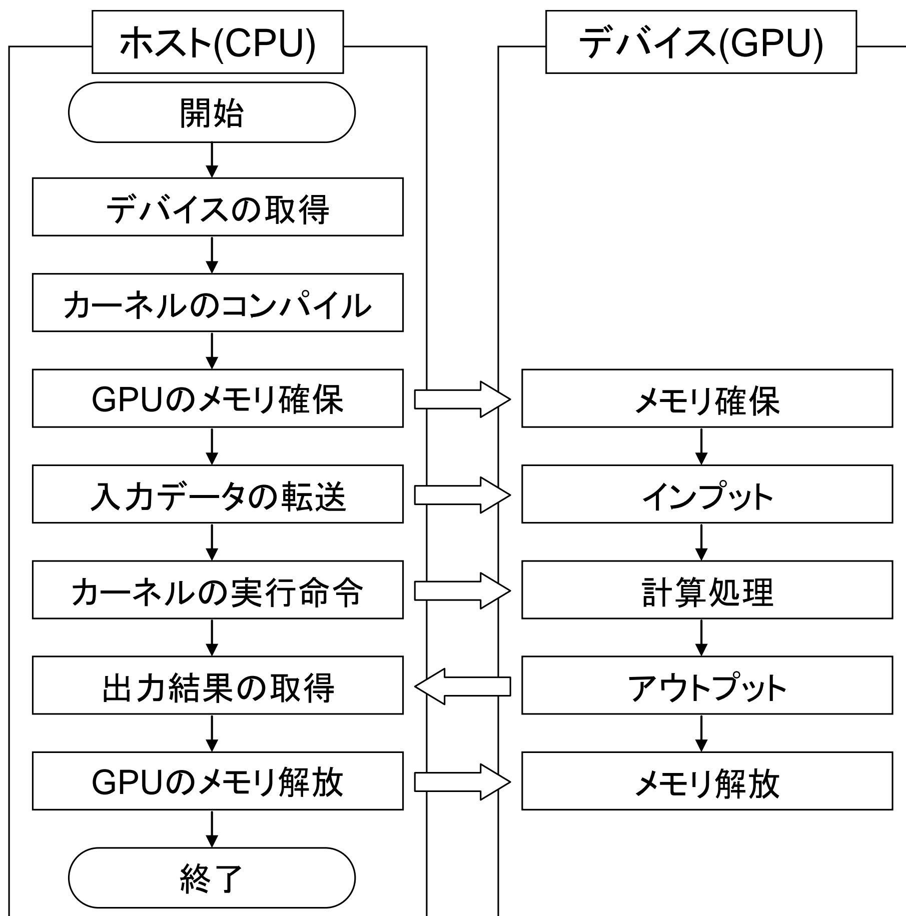

Fig. 3.8 OpenCL 処理フローチャート

デバイスで実行される処理は OpenCL C によって記述される。OpenCL C で記述されたソースコードを文字列として OpenCL の処理系に渡すと、カーネルにコンパイルされる。カーネルはデバイス上で実行できる形式に変換された関数を示している。デバイス上の処理の実行時にカーネルを指定することで、カーネルはホストからデバイス側に転送される。ホストがカーネルの実行命令を発行することで、デバイス上でカーネルが実行される。ホストとデバイス間の入出力は、メモリ確保とデータ転送をホスト側が明示的に行うことで実現される。ホスト側(CPU)のメインメモリと、デバイス側(GPU)の VRAM はハードウ

ェア上異なる場所に存在し、ホスト側とデバイス側で共通のメモリアドレス空間を持つことはできない。そのためホストとデバイスのメモリ領域は共有されないので、明示的にこの 2 つの離れたメモリ間でデータのコピーを行う必要がある。また、カーネルが VRAM 上に確保したメモリ領域にアクセスするためには、そのメモリ領域をカーネルの引数として指定しなければならない。

続いて、OpenCL の実行モデルについて説明する。OpenCL では並列化のために、デバイスで実行されるカーネルを 1〜3 次元の空間に配置する。この空間座標は 0 以上の整数(ID)で表され、空間上にマッピングされたカーネルをワークアイテムと呼ぶ。ワークアイテムはグループ化することが可能であり、このグループをワークグループと呼ぶ。ID の範囲と次元数はプログラマが柔軟に決めることができ、並列化させたい処理の単位に合わせて決定する。例えば、2 次元画像で画像を構成するピクセルごとに処理を実行したい場合、2 次元の ID を振り分ければ ID とピクセル座標が一致するためコード上で簡潔に記述できる。1 次元座標にマッピングした場合の概念図を、Fig. 3.9 に示す。

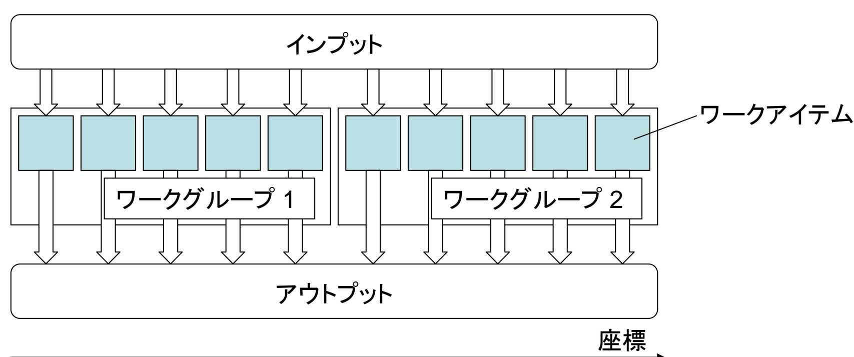

Fig. 3.9　1 次元座標へのカーネルのマッピング

ハードウェア上では、ワークグループは CU に、ワークアイテムは PE に割り当てられる。各ワークアイテムはそれぞれ異なる ID が渡され、ID に基づき異なる入力データを処理することによって、並列処理が実現されている。

続いて、OpenCL のメモリモデルについて説明する。OpenCL で明示的に扱われるメモリには、デバイス(GPU)上に存在するプライベートメモリ、ローカルメモリ、グローバルメモリ、コンスタントメモリの 4 種がある。ホスト(CPU)側のメモリ領域はホストメモリと呼ぶが、OpenCL の処理系が管理するわけではなく、単に OS 上のプロセスが実行されるアドレス空間である(malloc で確保したヒープ領域など、通常の C 言語においてポインタで扱うメモリ領域)。Fig. 3.10 にメモリモデルを示す。

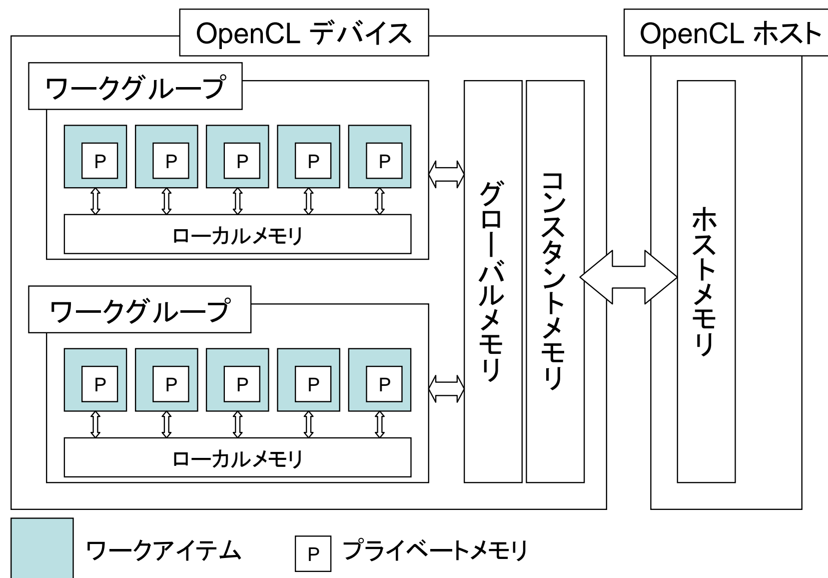

Fig. 3.10 OpenCL のメモリモデル

ホスト(CPU)から読み書きができるのは、グローバルメモリとコンスタントメモリの 2 つである。ワークアイテムからは、プライベートメモリ、自分の所属するワークグループのローカルメモリ、グローバルメモリにアクセス可能であり、コンスタントメモリは読み込みのみ許される。そのため、コンスタントメモリは入力データを置く領域になる。また、ワークアイテムからホストメモリに直接アクセスすることはできない。そのためワークアイテムで得られた計算結果をホストが得るためには、一旦グローバルメモリに結果を書き出し、ホストメモリにコピーしなければならない。グローバルメモリは入力・出力どちらにも使われるメモリ領域である。グローバルメモリはグラフィックボードの VRAM に相当し、数百 MB～数 GB の比較的大きな領域になる。コンスタントメモリは、実装によっては VRAM でなくキャッシュに確保される場合があり、この場合メモリ領域の大きさはグローバルメモリより小さくなる。

ローカルメモリはワークグループで共有されるメモリだが、グローバルメモリより容量が小さい(数 10KB)。その代わり、グローバルメモリよりはるかに(10 倍以上)アクセスが高速である[9]。そのため、アクセスの頻度が高く、ワークアイテム間の共有が活用できるデータに用いる。

プライベートメモリは一時的かつ独立のデータを置く領域であり、関数のローカル変数に近い扱いとなる。PE それぞれが持つ固有のメモリ領域に割り当てられ、容量は小さい。

3.6. GPU の特性を考慮したプログラミング手法

GPU 上で並列計算を行うコードを書くためには、CPU で逐次実行のコードとはまた異なるアプローチが必要になる。ここでは、GPU プログラミングにおいて高速化を目指す上で考慮すべき点を説明する。

(1) 条件分岐

GPU は本来グラフィクス処理を主に担うプロセッサであった。そのため、GPU のアーキテクチャは SIMD(Single Instruction Multiple Data)かそれに近いものとなっている。SIMD 演算では、同じ命令を複数の演算コアが同時に実行する。同じ命令を個々の演算コアが解釈するのではなく、命令を解釈するユニットが複数の演算コアを操作することで SIMD 演算が行われている。グラフィックス処理で主に行われるのは、座標ベクトルに対する演算であり、SIMD のアーキテクチャに適合していた。しかし、一般のアルゴリズムでは条件分岐により、全ての演算コアが同じ演算をするわけではない。

現代的な GPU では条件分岐を取り扱うために、どの分岐の演算結果が必要なのかを実行中に判定する仕組みになっている。条件が成立した場合は演算結果を残し、不成立の場合は演算結果を捨ててしまう[1]。条件分岐によって本来不要なはずの計算も、計算結果を捨てることで実質的に行わなかったことにする。そのため、条件分岐が多く複雑になると、後で結果が捨てられる演算に時間が割かれることになり、好ましくない。高速化のためには、複雑な条件分岐を回避することが必要になる。

(2) メモリアクセス

前節では、OpenCL による GPU プログラミングにおいては、複数種類のメモリ領域が存在することを説明した。この何種類か区別されたメモリ領域のうち、最適化に重要な要素となるのはローカルメモリである。ローカルメモリは、グローバルメモリよりアクセスが速く、容量が小さいので、頻繁にアクセスするデータを保持するのに向いている。また、ワークグループ内でローカルメモリは共有されるので、計算アルゴリズムによってはデータ共有により高速な動作が可能となる。ローカルメモリの利用は、グローバルメモリへのアクセスを減らし、グローバルメモリへの通信量(バンド幅)を減らすことができることも利点となる。ただし、ローカルメモリの容量はグローバルメモリに比べて小さいので、何のデータを保存するかは適切に選択しなければならない。

(3) ホストとデバイスのデータ転送

ホストとデバイス(CPU と GPU)のデータ転送はなるべく少なくすることが望ましい。グラフィックボード上の GPU をデバイスとする場合、ホスト-デバイス間のデータのやり取りには、PCIe と呼ばれる入出力インターフェイスを跨ぐ必要がある。これは GPU 内で完結するアクセスよりさらに遅いため、ホスト-デバイス間のデータのやり取りはなるべく最小限とする。

(4) ベクトル演算の利用

近年のプロセッサではベクトル演算が実装されており、これを利用することで高速化が行える場合がある。ベクトル演算とは、例えば 1 つの float 型変数(32bit)を 4 つにまとめ、4 つの変数に同じ命令を発行して処理を高速化する。OpenCL C ではベクトル演算は次のようにサポートされる。

```
//4 つの 32bit 単精度浮動小数点型をまとめる
float4 a = (float4)(1.0f, 2.0f, 3.0f, 4.0f);

float4 b = a * 4; //b の中身は(4.0f, 8.0f, 12.0f, 16.0f)
```

ベクトル演算は、例えば 128bit の処理を一度に出来るプロセッサが、32bit の処理を繰り返し行わずに 128bit 分のデータをそのまま処理するような場合に高速化される。これは CPU の場合でも同じであり、ベクトル演算が行えるように特殊な命令が実装されている。

3.7. 本章のまとめ

本章では、並列計算の基礎的概念と、GPU の構造と性能について説明した。続いて、OpenCL を用いた並列計算の実装方法を説明した。ワークアイテムとワークグループと呼ばれる処理単位によって並列計算が実現されていること、及び 4 つのメモリ領域に分割されたメモリモデルを持っていることを説明した。そして、GPU で実装する計算コードの高速化において必要な事柄を説明した。

第4章 GPU を用いたモンテカルロ中性子輸送計算コードの実装

4.1. 本章の概要

本章では、GPU 上でモンテカルロ中性子輸送計算を実装する手法について解説する。まず、並列化した場合のフローチャートを示す。次に GPU 上で行われる処理について述べ、固有値計算の実装における工夫点を述べる。そして、中性子束の計算方法について解説する。

4.2. 計算のフローチャート

モンテカルロ計算において、最も計算コストを要する箇所は中性子の追跡計算である。中性子同士の相互作用は無いため、中性子追跡計算は各中性子で独立である。そのため、中性子毎に並列化を行うことが出来る。GPU 上で並列化する場合、中性子ヒストリーを各 PE に振り分け、PE は独立に中性子追跡計算を行う。中性子の追跡計算以外で、並列化できない処理は CPU 側で実行することになる。つまり、入出力や核分裂源のサンプリングなどの処理は CPU で実行される。計算のフローチャートは Fig. 4.1 のようになる。

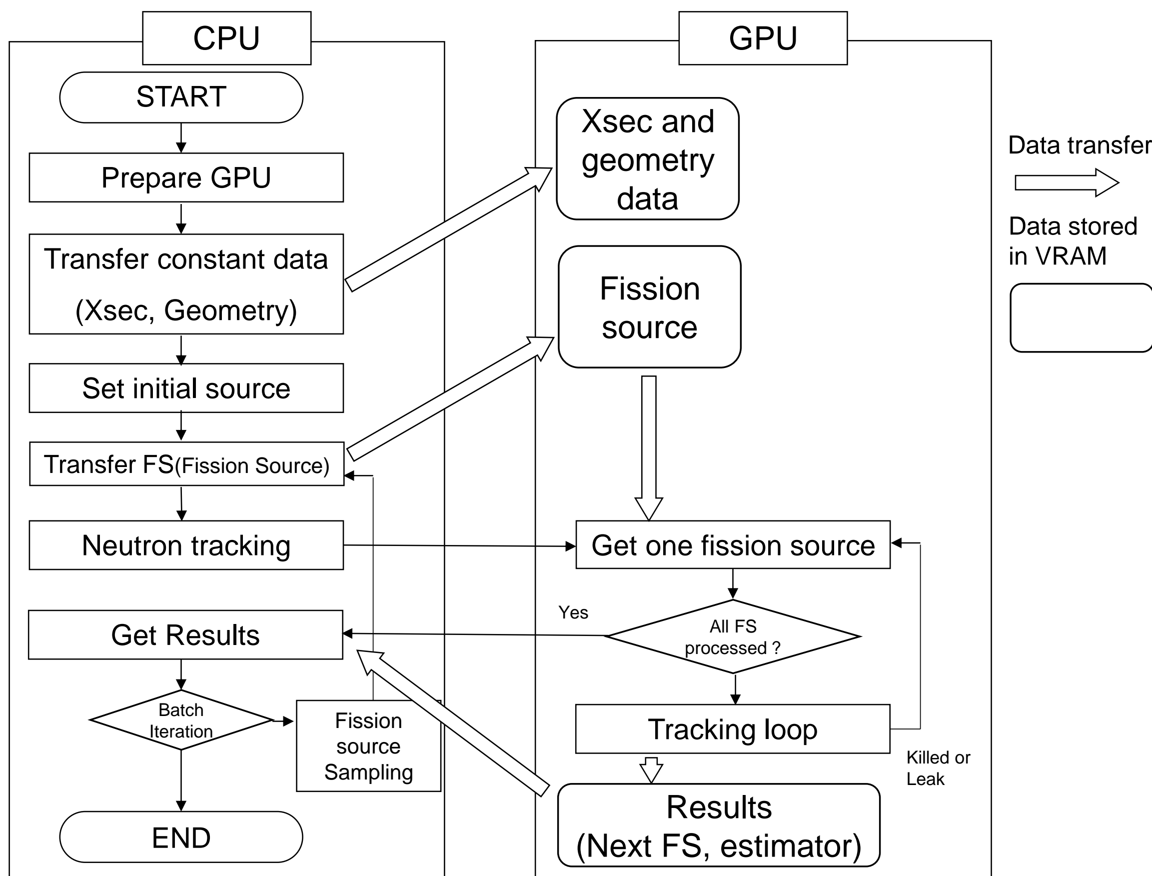

Fig. 4.1 計算のフローチャート

GPU上で追跡計算を行う前には、必要な断面積や幾何形状といったデータは予めVRAM上にコピーする。追跡計算の間は、これらの VRAM 上のデータが利用される。最初の世代の計算前には、中性子源初期分布を VRAM へコピーする。各 PE はこの中性子源から追跡開始座標を読み込み、中性子追跡計算を開始する。漏れかロシアンルーレットによって中性子が消滅したら、次の中性子源から追跡計算を開始する。追跡計算の間に、VRAM 上のエスティメータに加算を行い、次世代の核分裂源位置を記録していく。すべての中性子源を処理し終えたら、エスティメータの計算結果や、固有値計算の場合は発生した核分裂源分布を GPU の VRAM から CPU 側のメインメモリへ移す。次の世代の核分裂源のサンプリングは CPU 側で行う。全ての世代の計算結果が終われば、得られた計算結果を CPU 上で処理し、最終結果を出力する。

4.3. 乱数ジェネレータ

モンテカルロ計算の一番の基本は乱数の利用である。乱数を発生させるルーチンや機器を乱数ジェネレータと呼ぶ。計算機上で乱数を発生させる場合、殆どの場合は擬似乱数を生成する擬似乱数ジェネレータが用いられる。真の乱数を発生する乱数ジェネレータであれば、取り出して得られる値は完全にランダムになる。取り出す順番による影響はなく、取り出した数同士の関連もない。しかし、コンピュータ上でランダムな状態を作り出すには、基になるランダムな状態が必要になる。高いランダム性が求められる時は、コンピュータ上の環境ノイズから乱数を生成する方法もあるが、高速ではない。大抵の場合は乱数として擬似乱数が用いられ、モンテカルロ法においても同様である。

擬似乱数は、一見ランダムに見える数列を生成する。この数列は確定的な計算によって求められる。擬似乱数を生成する擬似乱数ジェネレータには複数の種類が存在するが、そのいずれも、ある初期状態(乱数シード)から漸化式に基づいて乱数列を生成する。擬似乱数生成法にはいくつかの種類が存在し、線形合同法やメルセンヌツイスタなどが有名である。

中性子輸送計算では線形合同法が一般的によく用いられており、MVP[7]や MCNP[13]にも用いられている。線形合同法は乱数としての品質は高くないが、高速であり使用メモリが小さくできる。

線形合同法では、乱数列は式(4.1)により計算される[14]。

$$S_{n+1} = (S_n \times A + C) \bmod M \tag{4.1}$$

S_nは前回に生成した乱数(整数)であり、ここから次の乱数 S_{n+1} が求められる。生成される乱数は 0～M-1 の範囲の一様乱数となる。計算機上では M には多くの場合 2 の累乗が用いられる。M=2^N であれば、あふれた桁を捨てることで余りを求める計算が簡単に実現するからである。

乱数の周期はAとCにどのような値を選ぶかによって変わる。不適切な値を選んだ場合、周期が短くなり、最悪の場合には$S_n = S_{n+1}$と計算されてしまう。適切な値を選んだ場合、乱数列は最大周期 M をもつ。最大周期は乱数を何ビットの整数で表すかで決まり、モンテカルロ計算では 35 ビット以上が推奨されている[15]。
乱数列が最大周期を持つようなAとCの選び方は次のようになる。

・CはMについて互いに素
・A-1 はMの全ての素数について倍数
・もしMが 4 の倍数なら、A-1 も 4 の倍数

また、AがMに対してあまりに小さく、Cも小さいような場合は乱数に偏りが生じることが考えられるのでその点も考慮しなければならない。
本研究では 64bit 線形合同法を用いており、S_nは各ワークアイテムが固有に保持する構造としている。複数の PE から単一の同時に参照、更新が起こると、3.3.4 において述べたようにデータ競合が発生するためである。各ワークアイテムが保持する状態は 8 バイトと小さいので、線形合同法は直接並列化できる。また、初期乱数S_0は CPU 側で生成して与えている。

4.4. 固有値計算の実装

固有値計算を行うためには、次の世代の核分裂源とするために、追跡中に起こった核分裂反応を記録しておく必要がある。また、エスティメータによってk_{eff}を評価する必要がある。これらの操作は、PE 毎の並列化を行っても正しく動作しなければならない。
k_{eff}を評価するエスティメータは、式(2.15)を用いて計算される。

$$k = \frac{1}{N}\sum_i \frac{\nu_i \Sigma_{f,i}}{\Sigma_{t,g}^*} W_i \quad \text{(2.15)(再掲)}$$

式(2.15)から分かるように、エスティメータの計算にはウェイトに断面積を掛けた値の合計値をとらなければならない。GPU 上に合計値を 1 つだけ保持する場合、複数の PE から同時に更新が行われ、データ競合が起こってしまう。これを回避するためには、各ワークアイテムが 1 つずつ加算値を保持し、1 世代の終了後に全ワークアイテムについて合計値を求めればよい。
核分裂源を記録する操作についても、並列化を考慮しなければならない。核分裂源を記録するには、次の 2 つのメモリ領域が必要になる。

・今まで何個の核分裂源が記録されたかを示すインデックス値。初期値は 0
・核分裂源の座標を格納する配列

例えば、あるタイミングで核分裂が発生したとする。インデックス値を読み込んだとして、これが 4 であれば、この核分裂は 4 個目だということがわかる。そして、インデックス値は 1 を足した値(つまり 5)に更新される。これにより、次の核分裂は 5 個目だということがわかる。また、4 個目の核分裂座標は、座標格納配列の 4 番目に格納すればよい。

この方法は、直接並列化を行った場合に問題が生じる。まず、複数の PE から 1 つのインデックス値にアクセスするため、データ競合が発生する。そのため、加算処理が正しく行われない。また、得られるインデックス値が一意にならない。同時にインデックス値を読み込むため、2 個以上の核分裂源について同一のインデックス値が割り振られてしまう可能性がある。

この問題は、エスティメータと同様にインデックス値と座標記録配列を個々のワークアイテムが独立に保持することで回避できる。しかし、どのワークアイテムでも同程度の核分裂が起きるわけではなく、ばらつきが存在する。そのため、実際の核分裂が起きる回数の期待値よりずっと大きな座標記録配列を用意するか、1 世代の計算の途中で座標記録配列にランダムに記録された核分裂源座標を、メモリ上で連続になるように最適化を行う必要が存在する。前者は、膨大なメモリ量が必要となり、またそのメモリがどれだけ必要か予測し難いという問題がある。後者については、1 世代の計算の途中で逐次実行が混ざるため、並列化効率が低下するという問題がある。

個々のワークアイテムにインデックス値と座標記録配列を割り当てることなく、整合性のとれた核分裂座標の記録を行うには、アトミック演算を用いる。OpenCL では、32bit 整数値を atomic_inc 関数によりインクリメントすることが出来る。

```
int atomic_inc(volatile int *p)
```

p:インクリメントしたい int 値
戻り値:更新前に p に収められていた値

インデックス値を更新するときは、atomic_inc 関数を呼び出し、自分のインデックス値はその戻り値を用いればよい。つまり Fig. 4.2 のように、インデックス値をアトミック演算によって問い合わせ・更新し、その値に応じて座標を配列に書き込むことにより、整合性のとれた核分裂源分布が得られる。

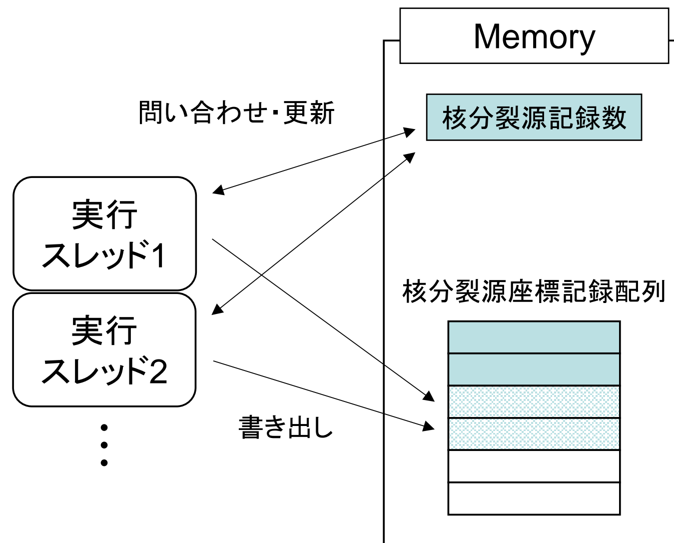

Fig. 4.2　核分裂源の記録

インデックス値をどのメモリ領域に置くかは、アトミック演算のパフォーマンスに依存する。グローバルメモリ上に置いた場合、GPU 内の全ての PE からアクセスされるため、アトミック演算のパフォーマンスが低下する可能性がある。ローカルメモリ上に置いた場合は、VRAM へのアクセスが削減されるため、パフォーマンスが改善される。その場合、CU 毎にヒストリーをグループ化することになる。グループ化されたヒストリーを CPU 上で処理する場合、1 コアが 1CU に相当するものとして割り当てられ、1 グループ内で実行されるヒストリーは常に 1 つになる。このため、実質的にアトミック演算が不要になる。

1 世代の計算が終了したら、次の世代の核分裂源を記録した座標配列からサンプリングする。このサンプリングは逐次実行でなければならないので、CPU 側で処理を行う。このとき、座標記録配列を CPU 側に転送して処理すると、転送しなければならない情報量が多くなる。そのため、実際には、配列が何番まで記録されているかの情報を CPU 側に転送し、CPU では配列の何番の座標を次の核分裂源として利用するかを計算し、GPU 側に転送する。これにより、座標データを直接扱うよりもメモリ転送量が小さくできる。ヒストリーのグループ化を行う場合は、このサンプリング時にはグループ毎に分けずにまとめて実行する。

CPU 側のサンプリングでは、同じ核分裂源が重複して選ばれないように非復元抽出を行う。(2.7)式による補正により、生成される中性子の数は大体同じであるが、核分裂源が未収束の場合や統計的な揺らぎによって生成された核分裂源が次世代のバッチサイズを下回ることもある。この場合は、重複を許すがなるべく偏らないようにサンプリングを行う。つまり、ある核分裂源は 2 回選ばれたが、他の核分裂源は選ばれた数が 0 回になる、といったことは避ける。なお、非復元抽出の高速な実装として、参考文献[14] p.133 の選択サンプリング法を用いている。

4.5. 中性子束の計算方法

4.5.1. 並列化における問題点

中性子束の計算はエスティメータによって行うことは述べたが、このエスティメータの計算をそのまま並列化することはできない。例えば、仮想散乱を用いた場合の中性子束は式(2.16)により計算される。

$$\phi = \sum_i \frac{W_i}{\Sigma_{t,i} V} \qquad \text{(2.16)(再掲)}$$

右辺の分母は、あるエネルギー群、領域について一定であるため、つまりはウェイトの値を追跡計算中に加算し、合計値を求める処理になる。しかし、追跡計算が並列化されていて、加算値を保存するメモリ領域が並列単位毎に独立していない場合、3.3.4 節に述べたようにデータ競合が起こってしまう。

データ競合を回避するためには、記録領域を並列単位ごと、つまりはワークアイテムごとに用意し、合計値を後で逐次的に求めるか、アトミック演算やロックを用いる必要がある。しかし、アトミック演算は浮動小数点数には提供されておらず、ウェイト値を直接扱うことはできない。また、エスティメータの値をワークアイテムごとに保持するか、衝突点座標をそのまま全て記録し、追跡計算後に CPU 側の逐次実行によって計算する方法もある。この方法では、追加のメモリ領域が必要となり、逐次実行による計算時間がかかるため、計算効率が大幅に低下する。

そこで本研究では、ウェイトの合計値を固定小数点数によって表現し、アトミック演算を利用する手法を提案する。

4.5.2. 固定小数点数

浮動小数点数はどんな実数でも $1.x \times 2^y$ の形で仮数部(x)と指数部(y)に分けて正規化されている。そして、仮数部ビットと実数部ビットに分かれて表現されている。固定小数点数では小数点の位置はプログラマが任意の位置で固定する。そして、整数部ビットと小数部ビットの組み合わせで実数を表現する。例えば、整数部に 4bit、小数部に 4bit、計 8bit の固定小数点数はのように表現される。

整数部ビット　小数部ビット

0101.0101 =4+1+1/4+1/16

Fig. 4.3　固定小数点数の例

固定小数点数の利点として、四則演算が整数値と同じようにできることが挙げられる。例えば、小数点位置が同じ固定小数点数同士の足し算を、整数値と全く同じ演算によって行うことができる。十進数で言えば、0.5+0.5=1.0 を、5+5=10 と置き換えて計算できることと同じである。筆算での加減算を思い出すと、小数点位置をそろえた後に整数と同じように加減算を行っていたことがわかる。この考え方は何進数でも同じである。固定小数点数の問題点として、扱える値の範囲が小さいためオーバーフローを起こしやすいという点が挙げられる。とくに、乗算、除算を行う場合に問題になりやすい。整数部ビットを n、小数部ビットを m ビットとすれば、表せる最大値は 2^n-2^{-m}、最小の精度は 2^{-m} になる。浮動小数点数であれば、単精度なら 2^{-126}～2^{+128} の範囲を表現できる。このため固定小数点数は、扱う値の範囲がある程度決まっていて、浮動小数点数用の回路のコストが重くなる信号処理の分野で使われている。

浮動小数点数から、固定小数点数に型変換をする場合は、2^m を掛けた後に整数型へ変換すれば良い。逆に固定小数点数から浮動小数点数に変換する場合は、まず浮動小数点数に変換した後、2^{-m} を掛ける。

4.5.3. アトミック演算との組合せ

固定小数点数は、加減算を整数と全く同じ操作で行うことができる。そのため、ウェイトの合計値を固定小数点数によって表現することで、アトミック演算を適用することが出来る。

中性子束を計算するために、計算したい領域の数だけウェイト合計値記録領域を VRAM 上に設定する。32bit では十分な精度が確保できないため、符号なし 64bit 整数型をウェイトの表現値として用いる。追跡中に衝突が起きた場合は、式(2.16)に従い対応するメモリ領域にアトミック演算を用いてウェイト値を加算していく。

ウェイトの値は最大値が 1 で、最小値もロシアンルーレットにおける W_{kill} によって設定されるため、ウェイトの合計値も同じく値の範囲が予測可能である。そのため、固定小数点数で表現できる。小数点位置が同じ固定小数点数同士の加減算は、プロセッサ上では整数同士の加減算と全く同じ操作になるため、アトミック演算を適用できる。このため、データ競合が起こることなくウェイトの合計値を計算でき、そこから中性子束を計算でき

る。領域の断面積・体積を掛けたり、規格化したりといった処理は、GPU での計算が終了後に CPU 側に転送してから行えばよい。

4.5.4. 数値計算における精度

先述の通り、固定小数点数は小数点位置を固定するため、浮動小数点数に比べると扱える値の範囲が予め決まってしまう。そのため、オーダーが大きく異なる数同士の演算を行う場合、精度が悪化する。ウェイトの場合、最大値は大抵の場合 1 であるため、その精度はウェイトの最小値(Wkill)の値に設定する値によって決まる。そして、W_{kill} の値に小さな値を入れると精度が悪化する可能性がある。単精度浮動小数点の仮数部には 23bit が割り当てられている。小数部に 32bit を割り当てた固定小数点数と比較すると、32-23=9 なので 2-9≈0.002 以下の数値を W_{kill} として指定すると、有効桁が小数部 32bit の固定小数点数の方が小さくなってしまう。単精度浮動小数点と同じだけの精度を保つなら、W_{kill} の値を調整するか、小数部ビットを増やす必要がある。

小数部ビットを増やすと、整数部ビットが少なくなるため、中性子束の大きいタリーではオーバーフローが起こらないように注意する必要がある。結局のところ、以上のような問題が起こるのは、中性子束が位置によって大きな差異が出るような計算体系になる。例えば遮蔽計算などを行う場合、中性子束の小さい箇所のために W_{kill} を小さくとらざるをえなくなり、またそれにあわせて整数部ビットを少なくすると、中性子束の差異が大きいため、線源の近くではオーバーフローの可能性が大きくなる。これは固定小数点数を用いた場合の最大の問題だと思われる。

ただ、タリーごとに小数点位置を個別に変えることは可能なので、中性子束分布にあわせて小数点位置を設定できれば精度の問題は解決できる。つまり、線源の近くでは整数部ビットを多めにとり、減衰した箇所では小数部ビットを多くとる。あらかじめ中性子束の値が分かっていないと設定できないが、だいたいのオーダーが分かれば良いので、低精度の計算を捨てバッチ等で行い、そこから決定できると考えられる。このように、この方法で精度よく中性子束を求めるには、体系によっては工夫が必要になる。ただし、統計誤差の方が大抵の場合は顕著だと思われる。

4.6. 本章のまとめ

本章では、モンテカルロ法を GPU 上で並列に実行するための実装方法について述べた。まず、フローチャートを示し、中性子ごとの並列化を行っていることを述べた。また、並列化において問題となる、核分裂源の記録と中性子束の計算について述べ、またその回避方法について説明した。

並列化されたコアそれぞれで中性子の追跡を行うと、核分裂源はランダムに各コアで発生する。このような場合、核分裂源を適切に記録するには各コア間で通信が必要になる。このため、核分裂源記録数をアトミック演算により管理する方法を用いている。

中性子束の計算では、データ競合を回避するため、固定小数点数とアトミック演算の組合せを用いている。アトミック演算は浮動小数点数に適用できないため、ウェイトの合計値を固定小数点数によって表現している。

第5章 検証計算

5.1. 本章の概要

本章では、本研究で GPU 上に実装したモンテカルロ計算コードを実行した場合、CPU に比べて計算時間の点で優位かどうかを検証する。計算結果が正しく求められるか、条件による計算時間の変化などについても議論する。検証体系として、Takeda ベンチマーク問題 Model1[16]と C5G7 ベンチマーク問題 3D unrodded[17]体系を取り上げている。また、GPU での条件分岐による性能低下を評価するため、条件分岐の無い理想的な場合のパフォーマンスの測定と、仮想散乱を用いない通常の追跡方法での性能評価を行っている。

5.2. Takeda ベンチマーク問題

5.2.1. 計算体系

Takeda ベンチマークは、三次元の 1/4 炉心を簡単な幾何形状で模擬した体系のベンチマーク問題である。本研究では、Takeda ベンチマーク問題のうち Model 1 を用いている。この体系の概念図を Fig. 5.1 に示す(参考文献[16] Fig.1 より引用)。

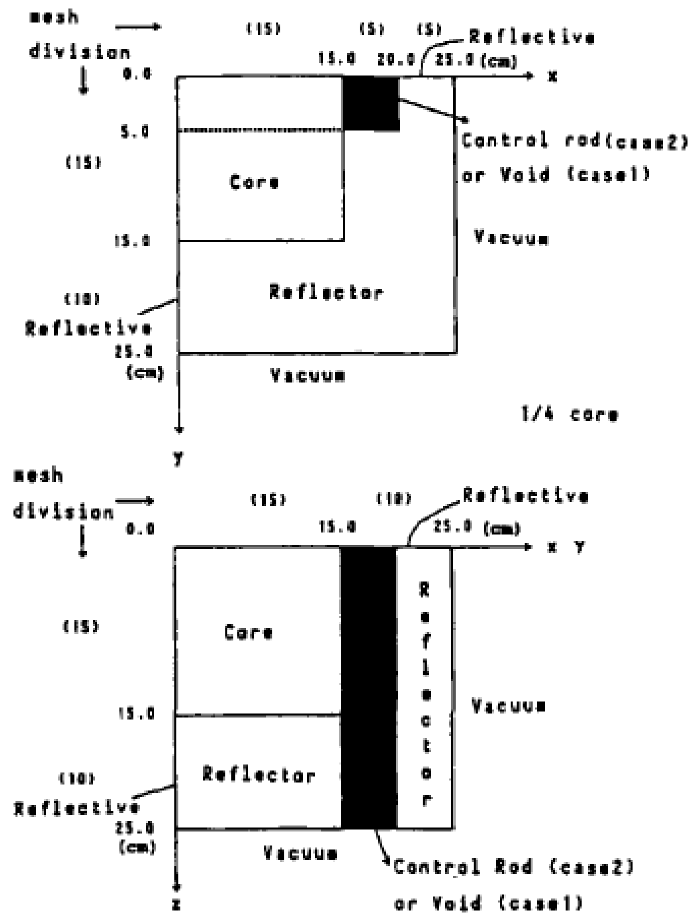

Fig. 5.1 Takeda ベンチマーク Model 1 体系

この体系は、1 辺 25cm の立方体であり、中心部に燃料領域、外周部に反射体領域、燃料領域に隣接してボイド(制御棒)領域が存在する。

各領域の断面積を Table 5.1 に示す。

Table 5.1 Takeda ベンチマーク問題の 2 群巨視的断面積

		Scattering		Transport	Absorption	Production	Fission Spectrum
	group	1→g	2→g				
Core	1	1.92423E-01	0.00000E+00	2.23775E-01	8.52709E-03	9.09319E-03	1.00000E+00
	2	2.28253E-02	8.80439E-01	1.03864E+00	1.58196E-01	2.90183E-01	0.00000E+00
Reflector	1	1.93446E-01	0.00000E+00	2.50367E-01	4.16392E-04	0.00000E+00	0.00000E+00
	2	5.65042E-02	1.62454E+00	1.64482E+00	2.02999E-02	0.00000E+00	0.00000E+00
Control Rod	1	6.77241E-02	0.00000E+00	8.52325E-02	1.74439E-02	0.00000E+00	0.00000E+00
	2	6.45461E-05	3.52358E-02	2.17460E-01	1.82224E-01	0.00000E+00	0.00000E+00
Empty(Void)	1	1.27700E-02	0.00000E+00	1.28407E-02	4.65132E-05	0.00000E+00	0.00000E+00
	2	2.40997E-05	1.07387E-02	1.20676E-02	1.32890E-03	0.00000E+00	0.00000E+00

単位: [1/cm]

以上の体系において、実効増倍率の計算と中性子束の計算を行った。また、検証には制御棒が挿入されていないボイドを含む Rod-out 体系を使用した。

計算時間の比較のため、OpenCL で記述されたコードを GPU・CPU 向けにコンパイルし、それぞれのハードウェアで実行している。利用したデバイスは Table 5.2 の通りである。また、CPU で実行する場合は、4Core8Thread で並列実行されている。

Table 5.2 計算に使用したデバイス

種類	CPU	GPU
名称	Intel Core i7-3770 3.40GHz(4Core 8Thread)	Radeon HD7950 800MHz
FLOPS	250GFLOPS	2.87TFLOPS

5.2.2. 実効増倍率の計算結果

バッチサイズ等のパラメータは Table 5.3 のように設定し、実効増倍率の計算を行った。実効増倍率の計算結果及び計算時間は Table 5.4 に示す。また、計算時間は、中性子束の計算を無効にした場合の結果を示している。なお、全バッチ数は捨てバッチを含めた世代数である。

Table 5.3　モンテカルロ計算のパラメータ

バッチサイズ	全バッチ数	捨てバッチ	最小ウェイト
2560000	100	10	0.01

Table 5.4　実効増倍率の計算結果

計算方法	実効増倍率 [-]	計算時間 [s]
GPU 並列	0.97739±0.00046	39.64
CPU 並列	0.97731±0.00046	262.1
参照解[16]	0.9778±0.0005	-

より、実効増倍率は不確かさの範囲で一致した。また、計算速度では、GPU は CPU の 6 倍程度の速度となっている。

5.2.3. 中性子束の計算結果

まず、アトミック演算を利用せずに直接計算を行った場合について示す。このときの計算パラメータは Table 5.3 の通りである。体系を 10x10x10 領域に分割し、炉心の xy 断面のうち z=0 の xy 面に接するメッシュの熱群の平均中性子束をグラフ化したものを Fig. 5.2 と Fig. 5.3 に示す。Fig. 5.2 は CPU、Fig. 5.3 は GPU で計算された結果を示している。中性子束の値は式(2.16)に基づいて計算したものである。

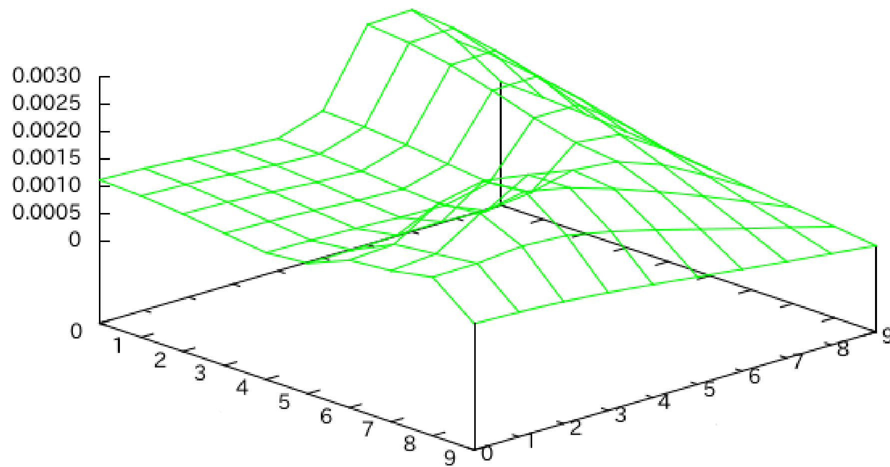

Fig. 5.2　Takeda ベンチマーク 2 群中性子束(CPU、アトミック演算なし)

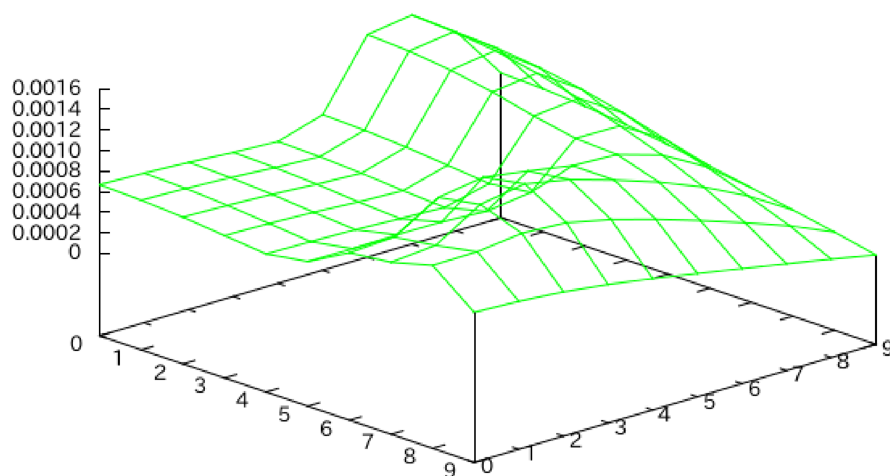

Fig. 5.3　Takeda ベンチマーク 2 群中性子束(GPU、アトミック演算なし)

縦軸のスケールで比較すると、Fig. 5.3 では中性子束の値が Fig. 5.2 より明らかに小さくなっていることがわかる。これは、データ競合に起因するものだと思われる。アトミック演算と、固定小数点数演算によって、同体系を同じく Table 5.3 の条件で中性子束の計算を行ったところ、中性子束分布は Fig. 5.4 のようになった。

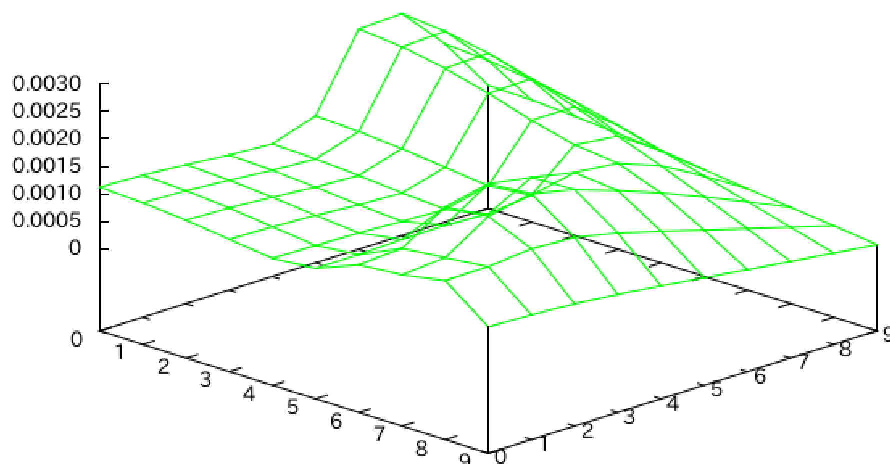

Fig. 5.4　Takeda ベンチマーク 2 群中性子束(GPU、アトミック演算あり)

また、アトミック演算を利用した中性子束の計算を行った場合と行わなかった場合で、計算時間を比較したものを Table 5.5 に示す。

Table 5.5　計算時間の比較

計算方法	中性子束計算なし	中性子束計算あり
GPU 並列	39.64	46.73
CPU 並列	262.1-	312.7

単位: 秒

Table 5.5 から、アトミック演算を利用して中性子束計算を行うと、GPU、CPU どちらのデバイスでも計算時間が増加することが分かった。

また、Core、Reflector、Void の各領域の平均中性子束を、10x10x10 分割の各メッシュから計算した結果、Table 5.6 のようになった。1 群燃料領域中性子束を 1.0 として規格化している。参照解は、論文[16]に記載された値から 1 群燃料領域中性子束を 1.0 として規格化し直したものである。

Table 5.6　各領域の平均中性子束

エネルギー群	領域	GPU による計算結果 [-]	参照解 [-]
1	Core	1.00000±0.00055	1.0000±0.0008
	Reflector	0.12464±0.00009	0.1249±0.0001
	Void	0.30355±0.00052	0.3033±0.0007
2	Core	0.18293±0.00011	0.1837±0.0002
	Reflector	0.19228±0.00011	0.1925±0.0002
	Void	0.20371±0.00027	0.2044±0.0007

Table 5.6 では、いずれの領域でも概ね不確かさの範囲で中性子束が一致している。

5.2.4. 考察

Takeda ベンチマーク問題により、CPU と GPU の計算時間を比較したが、おおよそ GPU の方が CPU より 6 倍程度高速化できていることを確認した。また、実効増倍率と中性子束の比較を行い、どちらも統計誤差の範囲で一致していた。

中性子束の計算では、アトミック演算を用いずに計算を行った場合、GPU と CPU での計算結果に大きな差が生じることが分かった。データ競合が起こった計算の結果は、デバイス依存の結果になるが、GPU の場合は中性子束の値が過小評価されるものとして表れている。CPU で計算した場合も、低頻度ながらデータ競合は起こっているものと推測される。

固定小数点数とアトミック演算を利用した場合、GPU で計算した場合に中性子束の結果が過小評価されることは無くなった。また、領域平均の中性子束を計算し、参照解と比較したところ、不確かさの範囲で一致していることを確かめた。アトミック演算を利用すると、10%程度の計算時間増加が起こったが、GPU において中性子束を正しく計算する方法としては高効率だと考えられる。

アトミック演算を用いずに中性子束を計算する場合、全ての衝突点位置を記録する必要があり、後から CPU 上で逐次実行によって計算することになる。記録される必要な情報は、少なくとも 3 次元座標とウェイトであり、単精度なら 16 バイトになる。もし、Σa/Σs が 1/20 程度であり、最小ウェイトが 0.1 だとすれば、ロシアンルーレットで中性子が消滅するまでに起こる衝突は 45 回程度(ln(0.10)/ln(19/20) ≈ 44.8)である。そしてバッチサイズが現在の計算条件と同じように 2560000 なら、記録に必要なメモリ領域は、40×2560000×16 = 1.6 GB になり、1.6GB を PCIe で GPU から CPU に転送するのに必要な時間は、PCIe の理論速度(最大 16GB/s)で 0.1 秒、本研究に使用した PC での実測値(9GB/s)で 0.18 秒になる。1 世代の計算にかかる時間は、Takeda ベンチマーク問題の体系で 0.4 秒なので、衝突点の記録を行った場合には速度が 40%低下すると推定される。一方、本研究で採用したアトミック演算を用いる方法では 10%程度の計算時間増なので優位性があるといえる。また、記

録された衝突点から中性子束を計算する方法ではCPUのリソースを多く使い、また1.6GBの衝突点データを処理するための計算時間がGPU側の処理より長くなることで律速になる可能性がある。最小ウェイトを小さくする場合、仮想散乱を導入した場合、散乱が多い場合などでは衝突点が増えるので、条件によってはそもそもVRAMが不足する可能性もある。よって、どのような条件でも10%程度の追加コストで計算が行えるアトミック演算による計算方法が優位だと言える。

なお、本節ではCPUとGPUの計算時間を比較するためにバッチ数を小さめにしている。Rod-in 体系を含めた、バッチ数を増やした計算については、GPU で実行した結果をAppendix A に記載する。

5.3. C5G7 ベンチマーク問題

5.3.1. 計算体系

C5G7 ベンチマーク問題の体系は、燃料集合体 4 つが減速材に囲まれたものである。燃料集合体は PWR を模擬しており、UO_2燃料と MOX 燃料の 2 種類がある。燃料集合体内には、円柱の燃料棒(一部は案内管)が 17x17 に整列しており、その隙間には減速材が存在する。エネルギー群は 7 群となっている。C5G7 ベンチマーク問題には、2 次元か 3 次元か、または制御棒の位置の違いによっていくつかのモデルが存在するが、本章では 3 次元unrodded(制御棒未挿入)体系を用いた検証計算について述べる。

C5G7 体系を上部から見た図を Fig. 5.5 に示す。また、各集合体の配置図および断面図をFig. 5.6、Fig. 5.7 に示す(参考文献[17] Figure1,5,6 より引用)。

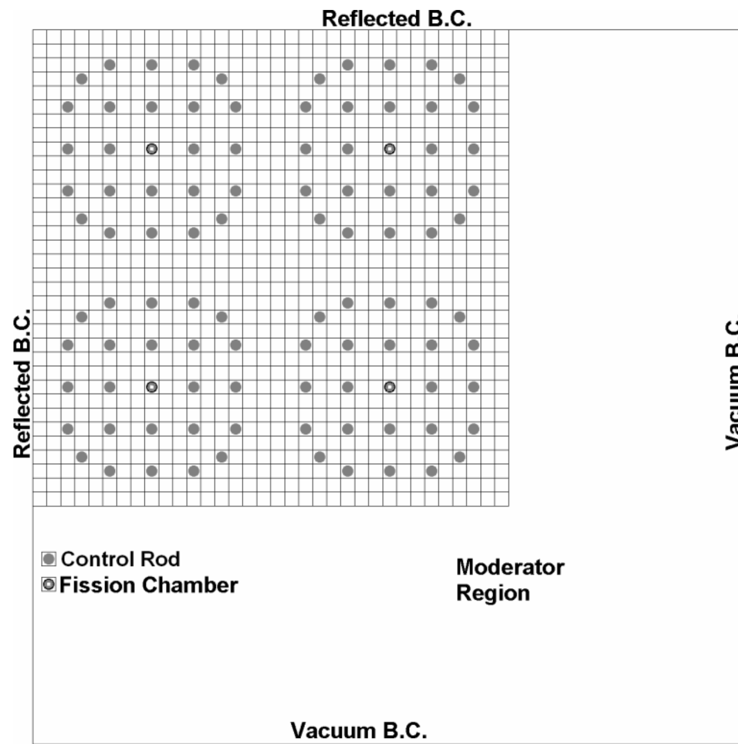

Fig. 5.5　C5G7 炉心上部から見た図

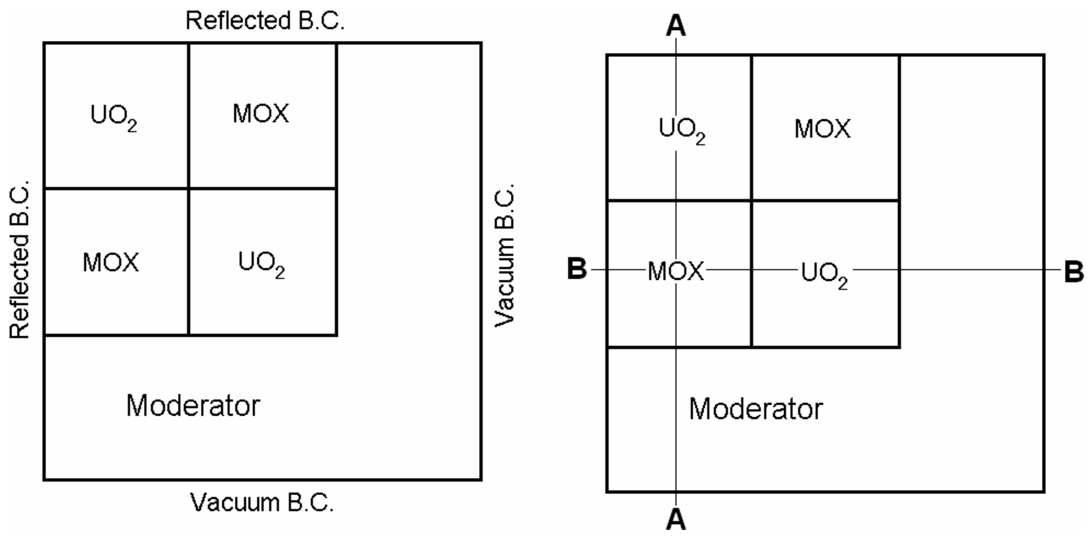

Fig. 5.6　集合体の配置図

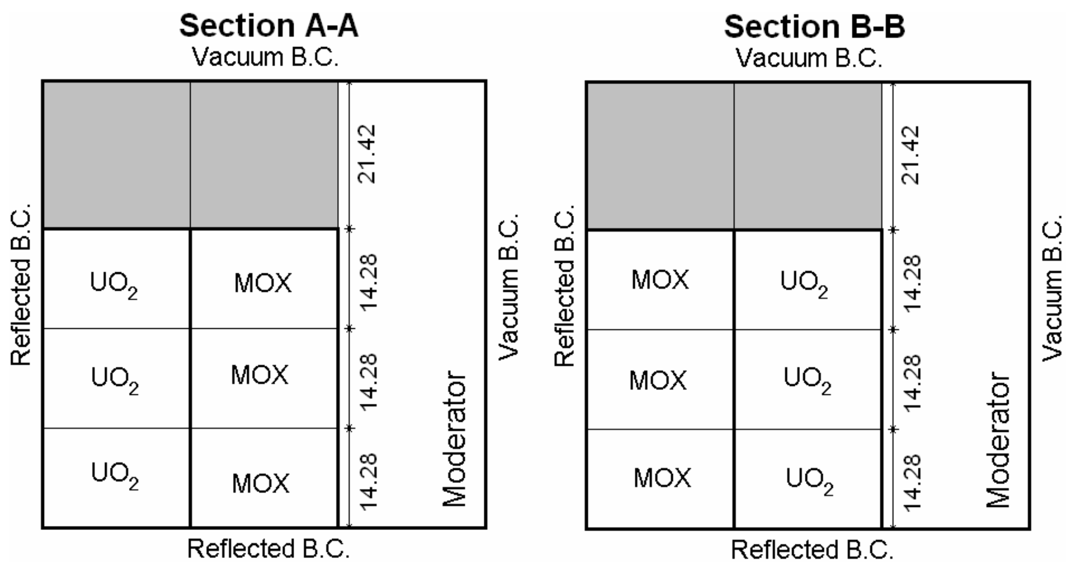

Fig. 5.7　Fig. 5.6 における AB についての断面図

C5G7 3 次元体系では、炉心の上部に減速材が存在するが、17x17 に並んだ円柱から構成される集合体のうち、マテリアルが案内管に割り当てられている円柱の上部には制御棒が存在する(Fig. 5.7 の灰色で塗りつぶされた箇所)。Unrodded 体系では、制御棒は集合体の上部にあるが、Rodded A 体系および Rodded B 体系においてはこの制御棒が集合体内に一部挿入される。つまり、案内管の一部が制御棒に置換される。

また、この体系の検証計算においても、CPU と GPU の両方で計算を行った。このときのモンテカルロ計算のパラメータは Table 5.7 に示すとおりで、また使用したデバイスは Table 5.2 と同じである。

Table 5.7　モンテカルロ計算のパラメータ

バッチサイズ	全バッチ数	捨てバッチ	最小ウェイト
2560000	1100	100	0.01

この計算においては、燃料棒領域におけるマテリアルを 2 次元の配列データとして与えている。C5G7 の体系は、円柱状の燃料棒(一部は案内管)と減速材の 2 領域からなる正方形のセルが整列した体系であり、それぞれのセルに対応する燃料棒のマテリアルを 2 次元配列として管理する。衝突点座標がどこのセルに対応するかは、各セルの縦横の幅が全て等しいため簡単に求められる。各座標をセルの幅で割り、余りを切り捨てればよい。セル内で円柱内か外かは、円柱がセルの中央に位置しているため、衝突点のセル中央からの距離と円筒の半径を比較することで判定することが出来る。また、z 軸方向の幾何形状は、z 座標に関して条件分岐を行う処理を挿入することによって判定している。

5.3.2. 実効増倍率の計算結果

計算された実効増倍率と、計算終了までに要した計算時間を Table 5.8 に示す。

Table 5.8　実効増倍率計算結果および計算時間

計算方法	実効増倍率 [-]	計算時間 [s]
GPU	1.143058±0.000014	453
CPU	1.143021±0.000014	3145
参照解[17]	1.14308±0.00003	-

5.3.3. 核分裂率の計算結果

中性子束の計算が正しく実行できているかを確認するため、各燃料棒の核分裂率分布を計算した。核分裂率分布は、燃料棒内の中性子束を求め、各群の中性子束と核分裂断面積の積によって求めている。C5G7 の集合体には、案内管などの核分裂に寄与しない箇所もあるため、そのような箇所を除いて平均が 1.0 になるように分布を規格化している。

規格化して求めた核分裂率分布の計算結果を Fig. 5.8 に示す。

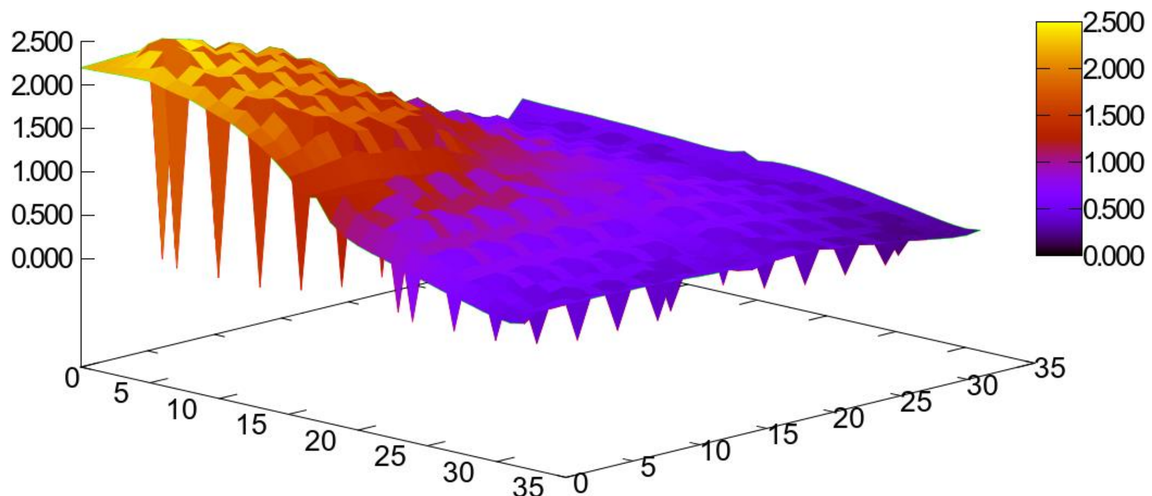

Fig. 5.8　C5G7 unrodded 体系 核分裂率分布

Fig. 5.8 の左側が炉心中央であり、この付近で核分裂率はピークとなっている。外周にいくにつれ小さくなり、炉心の核分裂率を再現できていると思われる。

そこで、MCNP によって計算された核分裂率分布を参照解とし、Fig. 5.8 の結果との相対差異分布を計算した。求められた相対差異分布を Fig. 5.9 に示す。

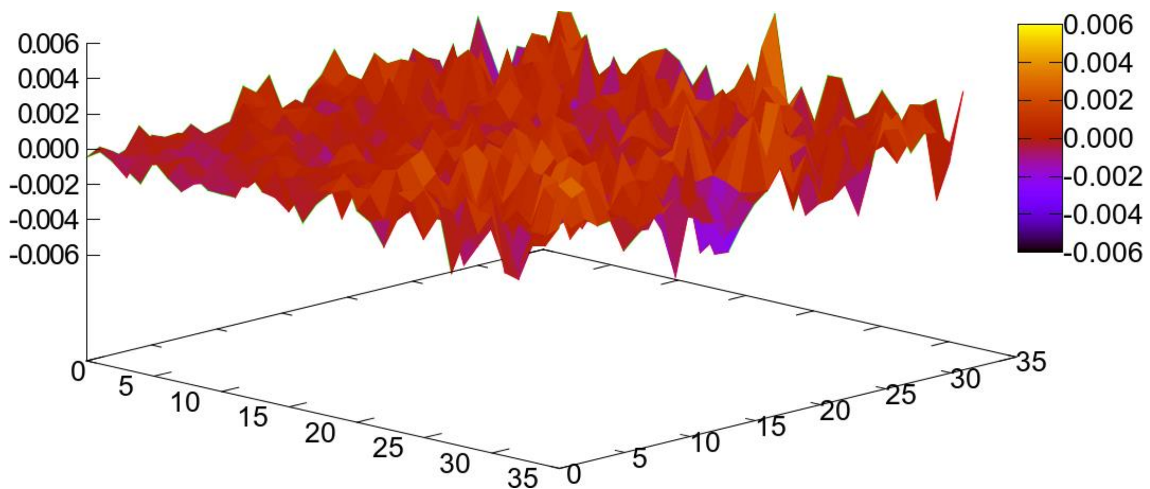

Fig. 5.9 C5G7 unrodded 体系 核分裂率相対差異分布

Fig. 5.9 に示した相対誤差では、系統的な値の差異は見られず、統計誤差に起因すると思われるランダムなノイズ状の誤差が見られる。また、相対誤差の最大値は 0.5%程度であった。

5.3.4. 考察

C5G7 の体系は Takeda ベンチマークより複雑な幾何形状であるが、計算時間の比(CPU/GPU)は 7 倍程度であり、Takeda ベンチマークでの性能比と同じ程度であった。幾何形状の判定が複雑になると、条件分岐の増加により性能比は悪化すると予想されるが、C5G7 の幾何形状ではこのような効果は効いてこなかったと考えられる。

また、計算された中性子束から核分裂率分布を計算し、参照解との比較を行った。系統的な差が見られなかったことから、正しく計算が行えていることがわかった。中性子束の計算による計算時間の増加は、Takeda ベンチマークと同様に 10%程度であり、許容可能な範囲であった。

Unrodded 以外の体系を含めたバッチ数を増やした計算については、GPU で実行した結果を Appendix B に記載する。

5.4. 乱数同一の場合の計算効率

5.4.1. 計算条件

これまでの検討で、自作のモンテカルロコードでの GPU/CPU 計算速度比を調べてきた。計算体系によって異なるものの、Takeda ベンチマークや C5G7 の提携において速度比は概ね 6 倍程度となっている。プロセッサの理論性能比は Table 5.2 に示すように 11 倍程度であり、実際の速度比はこれより小さい。この主な理由は、中性子の寿命が一定でなく、また中性子追跡の途中で複数の条件分岐が存在するため、GPU での並列計算ではプロセッ

サの空き時間が生じてしまうためと考えられる。そのため、もし計算過程で全てのプロセッサが同一の乱数を生成し、同じ計算過程を通過するような計算条件であれば、中性子寿命と条件分岐の差異による効率低下が無い場合の計算性能を計測することが出来る。

計算条件として、まず全ワークアイテムに与える初期乱数を同じ数値とする。また、どの世代でも初期中性子源を原点(0,0,0)に固定する。計算体系は C5G7 unrodded 体系とし、またモンテカルロ計算のパラメータは Table 5.9 とした。使用したデバイスはこれまでと同様に Table 5.2 のものである。

Table 5.9　モンテカルロ計算のパラメータ

バッチサイズ	バッチ数	最小ウェイト
2560000	10	0.01

5.4.2. 計算結果

得られた実効増場率と、計算時間を Table 5.10 に示す。

Table 5.10　乱数同一の場合の計算時間

	実効増倍率 [-]	計算時間 [s]
CPU	1.280095	23.8
GPU	1.280095	1.04

この計算は、全てのヒストリーが同じ乱数を用いるので 1 ヒストリーの計算を行っていることに等しい。並列化されて実行されている各ヒストリーの計算過程と結果は全く同じになるためである。そのため、計算された実効増倍率の値に意味は無いが、CPU と GPU で算出された値が等しいので、辿った計算過程は同じであることが分かる。

5.4.3. 考察

Table 5.10 から分かるように、CPU/GPU の計算時間比は 23 倍となり、Takeda ベンチマークや C5G7 の計算における 6 倍程度より大きな差となった。このことから、プロセッサごとに乱数が異なり、異なる計算フローを経ることによる GPU の効率低下による効果が大きいことが分かる。つまり、今後 GPU 向けの最適化を考える場合は、計算フローの差異による効率低下を念頭におかなくてはならない。具体的には、条件分岐を少なくすることと、中性子毎の寿命の差異の影響を小さくすることである。

理論的な FLOPS 比は、今回の CPU と GPU で 11 倍であったが、この結果ではそれ以上の差が開いている。この原因として考えられるのは、コンパイラの最適化の方法によっては必ずしも理論性能が得られるわけではないこと、またメモリアクセスなど浮動小数点数計算以外の箇所が律速になっている可能性が考えられる。

5.5. Regular-tracking 法での性能評価

5.5.1. 追跡方法の違いによる性能差

仮想散乱を用いず、中性子と境界面との交差を明示的に判定する通常の追跡方法を Regular-tracking 法とここでは呼称する。仮想散乱を用いる Delta-tracking 法では、幾何形状の判定を簡略化することで高速化を図っていた。本節では、Regular-tracking 法を GPU で実行した場合の計算性能を評価し、幾何形状の簡略化を行わなかった場合の性能を評価する。これにより、Delta-tracking 法による実装が実際に GPU 上で有用なのかを確認する。

5.5.2. 計算体系

計算体系は 1 次元で、Fig. 5.10 の体系を左右に 4 回繰り返した体系である。また、左右端の境界条件はどちらも真空とした。エネルギー群は 1 群である。

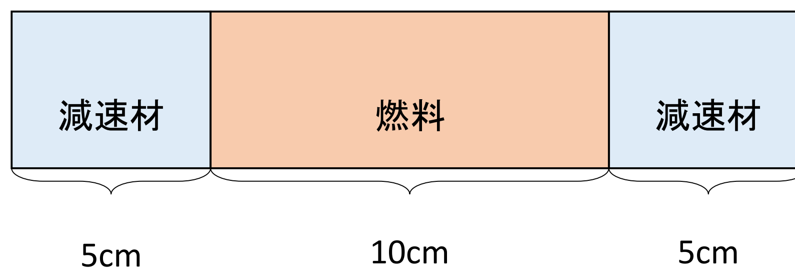

Fig. 5.10　Regular-tracking 性能評価のための計算体系

また、この体系の燃料、減速材の断面積は Table 5.11 の通りである。

Table 5.11　各領域の 1 群巨視的断面積データ

断面積 [cm^{-1}]	燃料	減速材
全断面積 Σt	0.27	0.7
吸収断面積 Σa	0.02	0.008
核分裂断面積 Σf	0.026	0.0

モンテカルロ計算のパラメータは Table 5.12 の通りとした。

Table 5.12　モンテカルロ計算のパラメータ

バッチサイズ	バッチ数	捨てバッチ	最小ウェイト
2560000	150	50	0.01

使用するデバイスであるが、Radeon HD7950 では Regular-tracking 法を実装したコードが GPU 用コンパイラのバグに起因すると思われる問題により実行できなかったため、これまでとは異なり Table 5.13 に示すデバイスを用いている。

Table 5.13　Regular-tracking 法の計算に使用したデバイス

種類	CPU	GPU
名称	Intel Core i7-3770 3.40GHz(4Core 8Thread)	GeForce GTX 960 1127MHz
FLOPS	250GFLOPS	2.3TFLOPS

以上の条件で、領域判定を Delta-tracking 法にした場合、Regular-tracking 法にして領域境界を 5cm、もしくは 1cm 毎にした場合の 3 通りで実効増倍率の計算を行った。領域境界が 5cm の場合、Fig. 5.10 の体系を 5cm ごと分割し、中性子が分割された領域を跨ぐかどうかを明示的に判定し、別領域に移動した場合は断面積を更新する。領域境界が 1cm の場合でも同様に、体系は 1cm ごとに分割した体系から構成されているものとしている。

5.5.3. 計算結果

各デバイスおよび追跡方法において計算された実効増倍率と、掛かった計算時間をに示す。

Table 5.14　各追跡法の計算結果

デバイス	追跡方法	実効増倍率 [-]	計算時間 [s]
GPU	Delta-tracking	0.929915±0.000020	46.8
	Regular-tracking 幅 5cm	0.929943±0.000018	37.3
	Regular-tracking 幅 1cm	0.929927±0.000018	52.0
CPU	Delta-tracking	0.929922±0.000018	831
	Regular-tracking 幅 5cm	0.929934±0.000019	587
	Regular-tracking 幅 1cm	0.929940±0.000019	754

Regular-tracking 法によって追跡を行った場合、領域の幅を小さくすると計算時間が増大している。GPU の場合、幅 5cm であれば Delta-tracking 法より高速であるが、1cm になると逆転し、Regular-tracking 法のほうが計算時間を要する結果となった。CPU の場合でも、逆転はせずともより計算時間を要している。

5.5.4. 考察

減速材領域の全断面積は、燃料領域ごとの全断面積の 2.6 倍となっている。そのため、仮想散乱を用いる Delta-tracking 法では燃料の全断面積が計算上 2.6 倍になる。これだけ衝突回数が増えるため、Delta-tracking 法の結果と Regular-tracking 法・領域幅 5cm の結果を比べると、Regular-tracking 法の方が高速になったと考えられる。

しかし、領域幅を 1cm と小さくした場合、中性子が領域境界を跨ぐ回数が増えることによって計算回数が増加し、計算時間は 2 つの追跡方法で逆転している。今回は 1 次元の簡易な体系であったが、GPU の場合 1cm の幅で Delta-tracking の方が高速な結果となった。CPU の場合、領域幅が狭くなることによる計算時間を増大させる効果は GPU より小さく、1cm の場合でも Delta-tracking の方が低速である。領域幅が狭くなることにより、中性子と領域境界の交点を求める処理の頻度が増え、それに伴い中性子の寿命の差異も大きくなる。CPU は GPU より計算時間の増大割合が小さかったのは、寿命の差異が大きくなる効果が GPU より効きにくいためだと思われる。

今回は 1 次元体系での検証であるが、 2 次元、3 次元となると、領域を跨ぐかどうかの判定回数は 1 次元より大きく増大するため、その場合 Delta-tracking 法の方がより優位となる。

Delta-tracking 法の問題点は、一部領域の全断面積が大きい場合、それに合わせて体系全体の全断面積が増え、衝突回数が増加して計算時間が増大することである。しかし、C5G7 での全断面積の差はせいぜい数倍程度であることから分かるように、多群エネルギーでは極端な断面積の差は生じにくい。そのため、幾何形状の判定を簡略化出来る Delta-tracking 法の方が優位な場合が多いと考えられる。一方で、連続エネルギーモンテカルロ法の計算では断面積の差が大きくなりやすく、Regular-tracking 法が優位になる場合も考えられる。

5.6. 本章のまとめ

本章では、Takeda ベンチマーク問題と C5G7 ベンチマーク問題について GPU と CPU を用いて計算を行った。どちらのベンチマークでも、CPU に対する GPU の性能比は 6 倍程度となり、GPU を用いることで CPU より高いパフォーマンスでモンテカルロ計算を行えることを示した。

Takeda ベンチマーク問題の中性子束を、並列性を考慮せずに直接計算したところ、データ競合により正しく計算が行えないことを示した。これを回避するために導入した、固定小数点数とアトミック演算を組み合わせた計算を実施し、Takeda ベンチマーク問題の中性子束の計算が正しく行われることを示した。また、C5G7 ベンチマーク問題の中性子束計算結果から核分裂率分布を求め、こちらも正しく求められていることを確認した。中性子束計算にアトミック演算を利用するコストは、計算時間にして 10%程度の増加であることが分かったが、これは十分許容できるコストだと考えられる。

また、乱数を同一にした計算を行い、性能を評価したところ、GPU の計算効率は高くなることが分かった。これにより、GPU の計算性能を活かすためには、中性子寿命の差異を小さくし、条件分岐を少なくする必要があることが分かった。

そして、仮想散乱を用いる Delta-tracking 法と、仮想散乱を用いない Regular-tracking 法について、簡単な体系における性能比較を行った。Regular-tracking 法では、異なるマテリアルを取り扱うために体系を複数の領域に分割し、領域境界と中性子の交点を明示的に計算する。この領域分割を細かくし、領域幅が狭くなると Regular-tracking 法の性能は大きく低下することが分かった。領域幅が狭くなると、中性子が領域境界を超える頻度が大きくなるため、交点を求めるための処理が増え、中性子間の寿命の違いも増える。GPU の場合ではこの効果が大きいと考えられ、1 次元の単純な体系でも Delta-tracking 法が Regular-tracking 法より高速である場合がみられた。この効果は 2 次元、3 次元とより体系が複雑になるにつれ大きくなると思われる。

第6章 結論

6.1. 結論

モンテカルロ法には、中性子の輸送を直接取り扱うことが出来るという利点が存在する。複雑な体系でも精度よく計算できるため、原子炉の経済性、安全性を向上させるために役立つ手法であると言える。しかし、計算結果には統計誤差が付随し、これを小さくするためには大きな計算コストが必要であるという欠点がある。本研究ではこの問題を解決するため、GPU の計算性能を利用することで、モンテカルロ法による炉心計算を高速に実行するための方法を考案した。

GPU は CPU と比較すると、並列計算に適合したアルゴリズムでなければ性能が出ず、また複雑な条件分岐を必要とする処理では性能が低下するという欠点を持っている。GPU での高速化を目指す場合、これらの課題を解決しなければならない。このため、モンテカルロ法による中性子輸送計算では、次の課題を解決する必要がある。

① 追跡計算での幾何形状判定が、複雑な条件分岐を含む処理である。

② 固有値計算においては、並列性を考慮して次世代の核分裂源を記録しなければならない。

③ 中性子束をエスティメータによって計算する処理では、並列性を考慮しなければならない。共有された記憶領域に対して直接加算計算を行うとデータ競合が発生する。

④ 中性子の寿命の差異が大きい場合、寿命の長い中性子を処理するコアが終了するまで、他のコアは計算を行えなくなる。これは計算効率の低下をもたらす。

これらの問題についての解説と、解決方法を各章で論じた。

1 章では、炉心計算およびモンテカルロ法の重要性について述べ、また近年の計算機シミュレーションの動向について述べた。そもそもなぜ並列計算が注目されているのか、また並列化によって性能が向上する根本的な理由を説明した。そして、GPU の優位性と欠点、解決すべき課題と本研究の目的を述べた。

2 章では、モンテカルロ法による中性子輸送計算手法について述べた。アナログ法と、ウェイトを導入した場合の非アナログ法それぞれの手法、中性子追跡の具体的な手順、固有値計算の実装方法を述べた。また、仮想散乱を用いた追跡手法について説明した。仮想散乱は、上記の課題①で述べた幾何形状判定の複雑さを緩和できる方法である。

3 章では、並列計算の基礎的概念と、GPU でのプログラミング手法について述べた。OpenCL を本研究では用いているため、OpenCL を用いた GPU プログラミング手法を中心に解説を行った。並列化によって起こるデータ競合の問題について解説し、またプログラミングする上で考慮しなければならない GPU の特性について解説した。

4 章では、GPU 上でモンテカルロ法を実装する手法を解説した。計算のフローチャートと、乱数の発生方法について述べた。その後、固有値計算の実装方法について述べた。上記の課題②で述べたように、並列性を考慮した核分裂源の記録を行わなければならないが、ここではアトミック演算を用いて記録数を管理し、正しく記録が行えるようにする手法を解説した。また、課題③を解決するために、アトミック演算と固定小数点数を用いた中性子束計算手法を提案した。

5 章では、4 章で解説した方法により実装したモンテカルロコードを用いて、検証計算を行った。Takeda ベンチマーク問題と C5G7 ベンチマーク問題の計算を行い、実効増倍率が正しく計算できることを確認した。それぞれのベンチマーク問題について、CPU と GPU それぞれの性能の比較を行い、GPU のほうが 6 倍程度高速に計算できることを確認した。また、中性子束の計算を行い、正しく計算が行えていることを確認し、4 章で提案した中性子束計算手法による計算コストが、実用上問題ないほど小さいことも明らかにした。その他に、コア間で乱数を同一化した計算を行い、課題④で述べた条件分岐と寿命の差異による計算コストの増大がどれだけなのかを見積もった。最後に、仮想散乱を用いない追跡方法と、仮想散乱を用いた追跡方法の性能を簡単な体系で比較した。その結果から、仮想散乱が GPU 上では多くの体系で優位だと言えることがわかった。

以上のように、本研究では GPU 上でモンテカルロ計算を行う上で解決しなければならない課題について解説し、その解決策・改善策を提案した。これらの提案手法により、実機炉心に近い体系である C5G7 ベンチマーク問題を 6 倍高速に計算できることを明らかにした。本研究は、高精度だが計算コストの大きいモンテカルロ法の応用範囲を広げる上で、高速化の面で貢献するものだと言える。

6.2. 今後の課題

今後の課題として、以下のものが挙げられる。

1. 連続エネルギーモンテカルロ法の実装

   本研究で実装したのは多群モンテカルロ法であり、連続エネルギーモンテカルロ法は実装していない。連続エネルギーモンテカルロ法では、断面積の取り扱いが多群法と異なる。このような処理が GPU 上で高速に実行する場合、異なる最適化が必要と考えられる。連続エネルギーの場合、扱う断面積データが増えるため、その点を考慮した最適化が必要になり、追跡計算の処理も複雑になる。また、仮想散乱の導入を行った場合に十分な速度を出せるか検討する必要がある。

2. 一般的な幾何形状の取り扱い

本研究での検証計算では、直方体や円柱のような単純な幾何形状しか取り扱っていない。また、一般的な幾何形状を取り扱うように実装していない。実用のモンテカルロコードでは、複雑な幾何形状を組合せ幾何形状や数式等を用いて記述する。組合せ幾何形状では、複数の図形の積和で幾何形状を表現しているが、このような幾何形状の表現は複雑な実装をもたらす。このような幾何形状の取り扱いを GPU 上で実装する場合、何らかの最適化が必要だと思われる。

3. より詳細・複雑な体系での性能評価

検証計算で行った最も複雑な体系は C5G7 ベンチマーク問題の体系である。これはエネルギー群 7 群で、マテリアルの種類も 8 種類と少ない。より詳細なエネルギー群で、マテリアルや領域の数が増えた場合、メモリアクセスの効率低下により計算時間が増大することが予想される。

参考文献

[1] コンピュータアーキテクチャの話, http://news.mynavi.jp/column/architecture/, (2015).

[2] 小玉泰寛, GPU を用いた MOC の高速化に関する研究, (2010).(修士論文)

[3] X. Du et al., "Evaluation of Vectorized Monte Carlo Algorithms on GPUs for a Neutron Eigenvalue Problem," Proc. M&C2013, Sun Valley, Idaho, May 5-9, 2013, M&C, (2013).

[4] Khronos OpenCL Working Group, "The OpenCL Specification Version:1.2 Document Revision: 19", http://www.khronos.org/registry/cl/specs/opencl-1.2.pdf (2015).

[5] 長屋康展, "粒子輸送モンテカルロ法の基礎原理とそこに潜む不確かさ," 第 44 回炉物理夏期セミナーテキスト, 日本原子力学会, pp.65-99 (2012).

[6] 山本俊弘, "Bolzmann 方程式の解法 –確率論的手法による解法-," 第 34 回炉物理夏期セミナーテキスト, 日本原子力学会, pp.79-91 (2002).

[7] 長家康展, 奥村啓介, 森貴正, 中川正幸, "MVP/GMVP 第 2 版: 連続エネルギー法及び多群法に基づく汎用中性子・光子輸送計算モンテカルロコード", MVP マニュアル, (2006).

[8] Jaakko Leppänen, "Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code," *Annuals of Nuclear Energy*, **37**, 5, p.715, (2010).

[9] Advanced Micro Devices, "AMD Accelerated Parallel Processing OpenCL Programming Guide," http://developer.amd.com/wordpress/media/2013/08/AMD_Accelerated_Parallel_Processing_OpenCL_Programming_Guide.pdf (2013).

[10]FLOPS 算出法, http://wiki.epii.jp/%E3%81%9D%E3%81%AE%E4%BB%96/FLOPS%E7%AE%97%E5%87% BA%E6%B3%95, (2013).

[11]techPowerUp GPU Database, http://www.techpowerup.com/gpudb/2637/geforce-gtx-960.html, (2015).

[12]池田成樹, OpenCL 並列プログラミング マルチコア CPU/GPU のための標準フレームワーク, カットシステム, pp.23-39(2010).

[13]Los Alamos National Laboratory: MCNP Home Page, available online https:://mcnp.lanl.gov

[14]Donald E. Knuth, The Art of Computer Programming Volume 2 Seminumerical Algorithms Third Edition 日本語版, 株式会社アスキー(2004).

[15]Forrenst B.Brown, "Fundamentals of Monte Carlo Particle Transport", available online http://kfe.fjfi.cvut.cz/~horny/ostatni/VU_HORNY/REFS1/LA-UR-05-4983_Monte_Carlo_Lectures.pdf.

[16]Takeda Toshikazu, Ikeda Hideaki, 3-D Neutron Transport Benchmarks, Journal of Nuclear Science and Technology, **28**, p.656-669, (1991).

[17] Nuclear Energy Agency Organisation for Economic Co-operation and Development, Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation.

Appendix

A. Takeda ベンチマーク問題計算結果

A.1.　計算条件

本節では、Takeda ベンチマーク問題のヒストリー数を大きくとった場合の計算結果について述べる。計算体系としては、Model 1 の Rod-out、Rod-in 両方の体系を取り上げている。モンテカルロ計算のパラメータは Table A.1 の通りである。計算に使用したデバイスは Table 5.2 のうち、GPU のみである。

Table A.1　Takeda ベンチマーク詳細計算の計算パラメータ

バッチサイズ	バッチ数	捨てバッチ	最小ウェイト
2560000	11000	1000	0.01

A.2.　計算結果

各体系において計算された実効増倍率と、計算時間は Table A.2 の通りである。

Table A.2　実効増倍率計算結果および計算時間

計算体系	実効増倍率 [-]	計算時間 [s]
Rod-out	0.977354±0.000005	4292
Rod-in	0.962443±0.000005	4160

また、各体系において計算された領域平均中性子束を Table A.3 に示す。

Table A.3　各領域の平均中性子束

エネルギー群	領域	Rod-out 体系 [-]	Rod-in 体系 [-]
1	Core	1.0000000±0.0000051	1.0000000±0.0000051
	Reflector	0.1246505±0.0000008	0.1204542±0.0000008
	Void/Rod	0.3035205±0.0000066	0.2499769±0.0000057
2	Core	0.1829397±0.0000012	0.1771344±0.0000012
	Reflector	0.1923025±0.0000012	0.1800023±0.0000011
	Void/Rod	0.2036734±0.0000046	0.0501463±0.0000014

B. C5G7 ベンチマーク問題計算結果

B.1.　計算条件

本節では、C5G7 ベンチマーク問題のヒストリー数を大きくとった場合の計算結果について述べる。計算体系として、3 次元の Unrodded、Rodded A、Rodded B 体系を取り上げる。モンテカルロ計算のパラメータは Table B.1 の通りである。計算に使用したデバイスは Table 5.2 のうち、GPU のみである。

Table B.1　Takeda ベンチマーク詳細計算の計算パラメータ

バッチサイズ	バッチ数	捨てバッチ	最小ウェイト
2560000	51000	1000	0.01

B.2.　実効増倍率計算結果

各体系において計算された実効増倍率と、計算時間は Table B.2 の通りである。

Table B.2　実効増倍率計算結果および計算時間

計算体系	実効増倍率 [-]	計算時間 [s]
Unrodded	1.1430439±0.0000020	23498
Rodded A	1.1281041±0.0000020	23578
Rodded B	1.0777269±0.0000020	24190

B.3.　核分裂率分布計算結果

Unrodded、Rodded A、Rodded B 体系における、各燃料棒の核分裂率の計算結果を Table B.3〜B.14 に記載する。核分裂率を z=0〜42.84cm まで積分し、値が 0 になる箇所を除いて平均すると 1.0 になるように規格化している。また z 方向の積分範囲として、z=0〜14.28cm、z=14.28〜28.56cm、z=28.56〜42.84cm、z=0〜42.84cm の各範囲における核分裂率分布をそれぞれ示している。

Table B.3　Unrodded 体系　z=0～14.28cm

0.976	0.978	0.982	0.987	0.989	0.990	0.970	0.953	0.941	0.912	0.887	0.865	0.824	0.779	0.725	0.660	0.570	0.584	0.474	0.418	0.386	0.363	0.343	0.319	0.297	0.278	0.255	0.235	0.217	0.196	0.178	0.169	0.183	0.267
0.978	0.983	0.994	1.010	1.022	1.053	1.001	0.982	0.999	0.941	0.915	0.920	0.852	0.798	0.735	0.664	0.570	0.577	0.599	0.522	0.488	0.468	0.467	0.411	0.380	0.375	0.328	0.301	0.294	0.254	0.226	0.210	0.231	0.264
0.982	0.994	1.026	1.085	1.097	0.000	1.058	1.037	0.000	0.993	0.967	0.000	0.915	0.858	0.761	0.672	0.572	0.574	0.589	0.526	0.525	0.500	0.000	0.425	0.389	0.000	0.338	0.308	0.000	0.272	0.243	0.213	0.227	0.262
0.987	1.010	1.084	0.000	1.109	1.090	1.019	0.997	1.017	0.954	0.931	0.952	0.923	0.000	0.806	0.685	0.575	0.575	0.594	0.563	0.000	0.497	0.504	0.428	0.394	0.392	0.340	0.312	0.315	0.264	0.000	0.230	0.229	0.262
0.989	1.022	1.097	1.109	1.066	1.078	1.011	0.991	1.011	0.949	0.925	0.942	0.889	0.877	0.814	0.694	0.577	0.577	0.606	0.575	0.528	0.511	0.484	0.415	0.383	0.382	0.330	0.303	0.303	0.273	0.241	0.232	0.234	0.262
0.990	1.053	0.000	1.090	1.078	0.000	1.042	1.022	0.000	0.979	0.954	0.000	0.900	0.862	0.000	0.717	0.579	0.577	0.635	0.000	0.571	0.511	0.000	0.440	0.403	0.000	0.351	0.320	0.000	0.278	0.258	0.000	0.246	0.262
0.970	1.001	1.058	1.019	1.012	1.042	0.984	0.967	0.988	0.926	0.902	0.913	0.845	0.806	0.786	0.681	0.568	0.567	0.594	0.546	0.512	0.464	0.463	0.403	0.373	0.373	0.322	0.295	0.293	0.251	0.234	0.219	0.230	0.259
0.953	0.983	1.037	0.997	0.991	1.022	0.967	0.950	0.971	0.910	0.886	0.896	0.829	0.790	0.771	0.670	0.560	0.559	0.585	0.535	0.502	0.455	0.455	0.397	0.368	0.368	0.318	0.292	0.288	0.247	0.230	0.215	0.227	0.257
0.941	0.999	0.000	1.017	1.011	0.000	0.988	0.971	0.000	0.931	0.906	0.000	0.849	0.808	0.000	0.684	0.555	0.553	0.607	0.000	0.531	0.480	0.000	0.422	0.389	0.000	0.339	0.308	0.000	0.264	0.242	0.000	0.237	0.255
0.912	0.941	0.993	0.954	0.949	0.979	0.926	0.910	0.931	0.874	0.851	0.861	0.797	0.760	0.743	0.646	0.539	0.540	0.566	0.519	0.487	0.442	0.443	0.386	0.358	0.359	0.311	0.285	0.282	0.242	0.225	0.211	0.222	0.251
0.888	0.915	0.967	0.931	0.925	0.954	0.902	0.886	0.906	0.851	0.829	0.840	0.779	0.744	0.726	0.631	0.527	0.528	0.554	0.510	0.480	0.436	0.436	0.380	0.353	0.353	0.306	0.281	0.279	0.240	0.223	0.209	0.220	0.248
0.865	0.920	0.000	0.952	0.942	0.000	0.913	0.896	0.000	0.861	0.840	0.000	0.796	0.763	0.000	0.639	0.518	0.519	0.573	0.000	0.519	0.466	0.000	0.405	0.372	0.000	0.326	0.297	0.000	0.259	0.241	0.000	0.230	0.246
0.824	0.852	0.915	0.923	0.889	0.900	0.845	0.829	0.848	0.797	0.779	0.796	0.752	0.744	0.695	0.595	0.497	0.499	0.529	0.506	0.465	0.453	0.433	0.372	0.345	0.346	0.300	0.276	0.277	0.250	0.221	0.213	0.215	0.241
0.779	0.798	0.858	0.000	0.876	0.862	0.806	0.790	0.808	0.760	0.744	0.763	0.744	0.000	0.657	0.562	0.475	0.480	0.502	0.482	0.000	0.430	0.439	0.375	0.347	0.347	0.302	0.279	0.282	0.237	0.000	0.208	0.208	0.237
0.725	0.735	0.761	0.806	0.814	0.000	0.786	0.772	0.000	0.743	0.726	0.000	0.695	0.657	0.587	0.523	0.451	0.461	0.483	0.440	0.445	0.428	0.000	0.368	0.340	0.000	0.299	0.274	0.000	0.244	0.219	0.192	0.204	0.235
0.660	0.663	0.672	0.685	0.694	0.717	0.681	0.670	0.684	0.645	0.631	0.640	0.595	0.562	0.523	0.481	0.425	0.448	0.486	0.439	0.420	0.409	0.412	0.366	0.342	0.339	0.299	0.276	0.271	0.235	0.210	0.196	0.213	0.238
0.570	0.570	0.572	0.575	0.577	0.579	0.568	0.560	0.554	0.539	0.527	0.518	0.497	0.475	0.451	0.425	0.391	0.451	0.405	0.381	0.365	0.351	0.337	0.317	0.298	0.281	0.261	0.242	0.225	0.205	0.187	0.177	0.187	0.256
0.584	0.577	0.574	0.575	0.576	0.577	0.567	0.559	0.554	0.540	0.528	0.519	0.499	0.480	0.461	0.448	0.451	0.354	0.352	0.344	0.335	0.324	0.312	0.294	0.277	0.262	0.243	0.225	0.210	0.192	0.176	0.167	0.174	0.223
0.474	0.598	0.589	0.594	0.606	0.635	0.594	0.585	0.607	0.566	0.554	0.573	0.529	0.502	0.483	0.486	0.405	0.352	0.368	0.370	0.368	0.361	0.360	0.329	0.311	0.303	0.273	0.254	0.244	0.217	0.197	0.184	0.189	0.235
0.418	0.522	0.526	0.563	0.575	0.000	0.546	0.535	0.000	0.519	0.510	0.000	0.507	0.482	0.440	0.439	0.381	0.344	0.370	0.385	0.400	0.395	0.000	0.357	0.335	0.000	0.296	0.275	0.000	0.239	0.218	0.196	0.195	0.238
0.386	0.488	0.525	0.000	0.528	0.571	0.512	0.502	0.531	0.487	0.480	0.519	0.465	0.000	0.445	0.420	0.365	0.335	0.368	0.400	0.000	0.397	0.379	0.342	0.322	0.316	0.284	0.264	0.259	0.240	0.000	0.207	0.198	0.236
0.363	0.468	0.500	0.497	0.511	0.511	0.464	0.455	0.480	0.442	0.436	0.466	0.453	0.430	0.428	0.409	0.351	0.324	0.361	0.395	0.397	0.373	0.367	0.333	0.314	0.308	0.278	0.259	0.252	0.228	0.219	0.206	0.197	0.231
0.343	0.467	0.000	0.504	0.484	0.000	0.464	0.455	0.000	0.443	0.436	0.000	0.433	0.439	0.000	0.412	0.337	0.312	0.360	0.000	0.379	0.367	0.000	0.335	0.316	0.000	0.280	0.260	0.000	0.227	0.211	0.000	0.198	0.225
0.319	0.411	0.425	0.428	0.415	0.440	0.403	0.397	0.422	0.386	0.380	0.405	0.372	0.375	0.368	0.366	0.317	0.294	0.329	0.357	0.342	0.333	0.335	0.307	0.290	0.286	0.257	0.240	0.233	0.207	0.192	0.189	0.183	0.215
0.297	0.380	0.389	0.394	0.382	0.403	0.373	0.368	0.389	0.358	0.353	0.372	0.345	0.347	0.340	0.342	0.298	0.277	0.311	0.335	0.322	0.314	0.316	0.290	0.275	0.271	0.245	0.228	0.221	0.196	0.182	0.179	0.174	0.205
0.278	0.375	0.000	0.392	0.382	0.000	0.373	0.368	0.000	0.359	0.353	0.000	0.346	0.347	0.000	0.339	0.281	0.262	0.303	0.000	0.316	0.308	0.000	0.286	0.271	0.000	0.241	0.225	0.000	0.194	0.179	0.000	0.171	0.195
0.255	0.328	0.338	0.340	0.330	0.351	0.322	0.318	0.339	0.311	0.306	0.326	0.300	0.302	0.299	0.299	0.261	0.243	0.273	0.296	0.284	0.278	0.280	0.257	0.245	0.241	0.218	0.203	0.197	0.175	0.162	0.160	0.155	0.183
0.235	0.301	0.308	0.312	0.303	0.320	0.295	0.292	0.308	0.285	0.281	0.297	0.276	0.279	0.274	0.276	0.242	0.226	0.253	0.275	0.264	0.259	0.261	0.240	0.228	0.225	0.203	0.189	0.184	0.164	0.152	0.150	0.145	0.171
0.217	0.294	0.000	0.315	0.303	0.000	0.293	0.288	0.000	0.282	0.279	0.000	0.277	0.282	0.000	0.271	0.225	0.210	0.244	0.000	0.259	0.252	0.000	0.233	0.221	0.000	0.197	0.184	0.000	0.160	0.149	0.000	0.140	0.159
0.196	0.254	0.272	0.263	0.273	0.278	0.251	0.247	0.264	0.242	0.240	0.259	0.250	0.237	0.244	0.235	0.205	0.192	0.217	0.239	0.240	0.228	0.226	0.207	0.196	0.194	0.175	0.164	0.160	0.145	0.139	0.131	0.125	0.146
0.178	0.226	0.243	0.000	0.241	0.258	0.234	0.230	0.242	0.225	0.223	0.241	0.221	0.000	0.219	0.210	0.187	0.176	0.197	0.218	0.000	0.219	0.211	0.192	0.182	0.179	0.162	0.152	0.149	0.139	0.000	0.120	0.114	0.133
0.169	0.210	0.212	0.230	0.232	0.000	0.218	0.215	0.000	0.211	0.209	0.000	0.213	0.208	0.192	0.196	0.177	0.167	0.184	0.196	0.207	0.206	0.000	0.189	0.180	0.000	0.160	0.150	0.000	0.131	0.120	0.107	0.105	0.123
0.183	0.231	0.227	0.229	0.234	0.246	0.230	0.227	0.237	0.222	0.220	0.230	0.215	0.208	0.204	0.213	0.187	0.174	0.189	0.195	0.198	0.197	0.198	0.183	0.174	0.171	0.155	0.145	0.140	0.125	0.114	0.105	0.103	0.118
0.267	0.264	0.262	0.262	0.262	0.262	0.259	0.257	0.255	0.251	0.248	0.246	0.241	0.237	0.235	0.239	0.256	0.223	0.235	0.237	0.236	0.231	0.225	0.215	0.205	0.195	0.183	0.171	0.159	0.146	0.133	0.123	0.118	0.127

Table B.4　Unrodded 体系　z=14.28～28.56cm

0.776	0.778	0.781	0.785	0.787	0.787	0.771	0.758	0.748	0.726	0.706	0.688	0.656	0.620	0.577	0.525	0.454	0.465	0.377	0.333	0.307	0.289	0.273	0.254	0.236	0.221	0.203	0.187	0.173	0.156	0.142	0.134	0.146	0.213
0.778	0.782	0.790	0.803	0.813	0.837	0.796	0.781	0.794	0.748	0.728	0.732	0.678	0.634	0.585	0.528	0.454	0.459	0.476	0.416	0.388	0.373	0.371	0.327	0.303	0.299	0.261	0.239	0.234	0.202	0.180	0.167	0.184	0.210
0.781	0.791	0.816	0.863	0.872	0.000	0.841	0.825	0.000	0.790	0.769	0.000	0.728	0.682	0.605	0.535	0.455	0.457	0.468	0.418	0.417	0.398	0.000	0.338	0.310	0.000	0.269	0.245	0.000	0.216	0.193	0.169	0.181	0.209
0.785	0.803	0.863	0.000	0.882	0.867	0.810	0.793	0.809	0.759	0.741	0.757	0.735	0.000	0.641	0.545	0.458	0.458	0.473	0.448	0.000	0.395	0.401	0.341	0.313	0.312	0.270	0.249	0.250	0.210	0.000	0.183	0.182	0.209
0.787	0.813	0.872	0.882	0.848	0.857	0.805	0.788	0.804	0.755	0.736	0.749	0.707	0.697	0.648	0.552	0.459	0.459	0.482	0.458	0.420	0.407	0.385	0.330	0.304	0.304	0.263	0.242	0.241	0.218	0.192	0.184	0.186	0.209
0.787	0.837	0.000	0.867	0.857	0.000	0.829	0.813	0.000	0.779	0.759	0.000	0.716	0.685	0.000	0.570	0.461	0.459	0.505	0.000	0.455	0.406	0.000	0.350	0.321	0.000	0.280	0.254	0.000	0.221	0.206	0.000	0.196	0.209
0.771	0.796	0.841	0.810	0.804	0.829	0.783	0.769	0.786	0.737	0.717	0.726	0.672	0.641	0.625	0.542	0.452	0.451	0.473	0.434	0.408	0.369	0.369	0.320	0.297	0.297	0.257	0.235	0.233	0.200	0.186	0.174	0.183	0.206
0.758	0.781	0.824	0.793	0.788	0.813	0.769	0.755	0.772	0.724	0.705	0.713	0.660	0.629	0.614	0.533	0.445	0.445	0.465	0.426	0.399	0.362	0.362	0.316	0.293	0.293	0.253	0.232	0.230	0.197	0.183	0.171	0.181	0.204
0.749	0.794	0.000	0.809	0.804	0.000	0.786	0.772	0.000	0.741	0.721	0.000	0.675	0.643	0.000	0.544	0.441	0.440	0.483	0.000	0.423	0.382	0.000	0.336	0.309	0.000	0.270	0.245	0.000	0.210	0.193	0.000	0.189	0.203
0.726	0.748	0.790	0.759	0.755	0.779	0.737	0.724	0.741	0.695	0.677	0.685	0.634	0.605	0.591	0.514	0.429	0.429	0.450	0.413	0.387	0.352	0.353	0.307	0.285	0.286	0.247	0.227	0.225	0.193	0.179	0.168	0.177	0.200
0.706	0.728	0.769	0.741	0.736	0.758	0.717	0.705	0.721	0.677	0.660	0.668	0.620	0.592	0.578	0.502	0.420	0.420	0.441	0.406	0.382	0.347	0.347	0.303	0.281	0.281	0.244	0.224	0.222	0.191	0.178	0.166	0.175	0.198
0.688	0.732	0.000	0.757	0.749	0.000	0.726	0.713	0.000	0.685	0.668	0.000	0.633	0.607	0.000	0.509	0.412	0.413	0.456	0.000	0.413	0.371	0.000	0.322	0.296	0.000	0.260	0.237	0.000	0.206	0.192	0.000	0.183	0.196
0.655	0.678	0.728	0.735	0.707	0.716	0.672	0.660	0.675	0.634	0.620	0.633	0.598	0.592	0.553	0.473	0.396	0.397	0.421	0.403	0.370	0.361	0.344	0.296	0.274	0.275	0.239	0.220	0.221	0.199	0.176	0.170	0.171	0.192
0.620	0.634	0.682	0.000	0.697	0.685	0.641	0.629	0.643	0.605	0.592	0.607	0.592	0.000	0.522	0.447	0.378	0.382	0.399	0.383	0.000	0.342	0.349	0.298	0.276	0.276	0.241	0.222	0.225	0.189	0.000	0.165	0.165	0.189
0.577	0.585	0.605	0.641	0.648	0.000	0.625	0.614	0.000	0.591	0.578	0.000	0.553	0.523	0.467	0.416	0.359	0.367	0.384	0.350	0.354	0.341	0.000	0.293	0.271	0.000	0.238	0.218	0.000	0.194	0.174	0.153	0.163	0.187
0.525	0.528	0.535	0.545	0.552	0.570	0.542	0.533	0.544	0.514	0.502	0.509	0.473	0.447	0.416	0.383	0.338	0.357	0.387	0.350	0.334	0.325	0.328	0.291	0.272	0.270	0.238	0.220	0.215	0.187	0.167	0.156	0.170	0.190
0.453	0.454	0.455	0.458	0.459	0.460	0.452	0.445	0.441	0.429	0.420	0.412	0.396	0.378	0.359	0.338	0.311	0.359	0.322	0.303	0.291	0.279	0.268	0.252	0.237	0.224	0.207	0.192	0.179	0.163	0.149	0.141	0.149	0.204
0.465	0.459	0.457	0.458	0.459	0.459	0.451	0.445	0.440	0.429	0.420	0.413	0.397	0.382	0.367	0.357	0.359	0.282	0.280	0.274	0.266	0.257	0.248	0.234	0.221	0.209	0.193	0.179	0.167	0.152	0.140	0.133	0.138	0.178
0.377	0.476	0.468	0.473	0.482	0.505	0.473	0.465	0.483	0.450	0.441	0.456	0.421	0.399	0.384	0.387	0.322	0.280	0.293	0.295	0.293	0.287	0.286	0.262	0.247	0.241	0.217	0.202	0.194	0.172	0.157	0.147	0.151	0.187
0.333	0.416	0.418	0.448	0.458	0.000	0.434	0.426	0.000	0.413	0.406	0.000	0.403	0.383	0.350	0.350	0.303	0.274	0.295	0.306	0.319	0.314	0.000	0.284	0.267	0.000	0.236	0.219	0.000	0.190	0.173	0.156	0.155	0.189
0.307	0.388	0.417	0.000	0.420	0.455	0.408	0.399	0.423	0.387	0.382	0.413	0.370	0.000	0.354	0.334	0.290	0.266	0.293	0.319	0.000	0.316	0.301	0.272	0.256	0.251	0.226	0.210	0.206	0.191	0.000	0.165	0.157	0.188
0.289	0.373	0.398	0.395	0.407	0.406	0.369	0.362	0.382	0.352	0.347	0.371	0.361	0.342	0.341	0.325	0.279	0.257	0.288	0.314	0.316	0.297	0.292	0.265	0.250	0.245	0.221	0.206	0.201	0.182	0.175	0.164	0.156	0.184
0.273	0.371	0.000	0.401	0.385	0.000	0.369	0.362	0.000	0.353	0.347	0.000	0.345	0.349	0.000	0.328	0.268	0.248	0.286	0.000	0.301	0.292	0.000	0.266	0.252	0.000	0.223	0.207	0.000	0.180	0.168	0.000	0.157	0.179
0.254	0.327	0.338	0.341	0.330	0.350	0.320	0.316	0.336	0.307	0.303	0.322	0.296	0.298	0.293	0.291	0.252	0.234	0.262	0.284	0.272	0.265	0.266	0.244	0.231	0.228	0.205	0.191	0.185	0.165	0.153	0.151	0.146	0.171
0.236	0.303	0.310	0.313	0.305	0.321	0.297	0.293	0.309	0.285	0.281	0.296	0.274	0.276	0.271	0.272	0.237	0.221	0.247	0.267	0.256	0.250	0.252	0.231	0.219	0.216	0.195	0.181	0.176	0.156	0.145	0.143	0.139	0.163
0.221	0.299	0.000	0.312	0.304	0.000	0.297	0.293	0.000	0.286	0.281	0.000	0.275	0.276	0.000	0.270	0.224	0.209	0.241	0.000	0.251	0.245	0.000	0.228	0.216	0.000	0.192	0.179	0.000	0.154	0.143	0.000	0.136	0.155
0.203	0.261	0.269	0.270	0.263	0.280	0.257	0.253	0.270	0.247	0.244	0.260	0.239	0.241	0.238	0.238	0.207	0.193	0.217	0.236	0.226	0.221	0.223	0.205	0.195	0.192	0.173	0.161	0.157	0.139	0.129	0.128	0.124	0.145
0.187	0.239	0.245	0.249	0.242	0.254	0.235	0.232	0.245	0.227	0.224	0.237	0.220	0.222	0.218	0.220	0.192	0.179	0.202	0.219	0.210	0.206	0.207	0.191	0.181	0.179	0.162	0.151	0.146	0.130	0.121	0.119	0.116	0.136
0.173	0.234	0.000	0.250	0.241	0.000	0.233	0.229	0.000	0.225	0.222	0.000	0.221	0.225	0.000	0.216	0.179	0.167	0.194	0.000	0.206	0.200	0.000	0.185	0.176	0.000	0.157	0.146	0.000	0.127	0.118	0.000	0.111	0.127
0.156	0.202	0.216	0.210	0.218	0.221	0.200	0.197	0.210	0.193	0.191	0.206	0.199	0.189	0.194	0.187	0.163	0.152	0.172	0.190	0.191	0.182	0.180	0.165	0.156	0.154	0.139	0.130	0.127	0.115	0.111	0.105	0.099	0.116
0.142	0.180	0.193	0.000	0.192	0.206	0.186	0.183	0.193	0.179	0.178	0.192	0.176	0.000	0.174	0.167	0.149	0.140	0.157	0.173	0.000	0.175	0.167	0.153	0.145	0.142	0.129	0.121	0.118	0.111	0.000	0.096	0.090	0.106
0.134	0.167	0.169	0.183	0.184	0.000	0.174	0.171	0.000	0.168	0.166	0.000	0.170	0.165	0.153	0.156	0.141	0.133	0.147	0.156	0.165	0.164	0.000	0.151	0.143	0.000	0.128	0.119	0.000	0.105	0.096	0.085	0.083	0.098
0.146	0.184	0.181	0.182	0.186	0.195	0.183	0.181	0.189	0.177	0.175	0.183	0.171	0.165	0.163	0.170	0.149	0.138	0.151	0.155	0.157	0.156	0.157	0.146	0.139	0.136	0.124	0.116	0.111	0.099	0.090	0.083	0.082	0.094
0.213	0.210	0.209	0.209	0.209	0.209	0.206	0.204	0.203	0.200	0.198	0.196	0.192	0.189	0.187	0.190	0.204	0.178	0.187	0.189	0.188	0.184	0.179	0.171	0.163	0.155	0.145	0.136	0.127	0.116	0.106	0.098	0.094	0.101

Table B.5　Unrodded 体系　z=28.56～42.84cm

0.446	0.447	0.447	0.448	0.447	0.446	0.437	0.429	0.423	0.411	0.400	0.390	0.373	0.354	0.331	0.303	0.263	0.271	0.220	0.195	0.179	0.168	0.159	0.147	0.137	0.128	0.118	0.108	0.100	0.091	0.083	0.078	0.084	0.120
0.447	0.448	0.451	0.456	0.459	0.470	0.448	0.440	0.445	0.421	0.410	0.411	0.383	0.360	0.334	0.304	0.263	0.267	0.278	0.242	0.225	0.215	0.213	0.188	0.174	0.171	0.150	0.138	0.134	0.116	0.104	0.097	0.106	0.119
0.447	0.451	0.462	0.483	0.487	0.000	0.469	0.460	0.000	0.441	0.429	0.000	0.406	0.383	0.343	0.306	0.262	0.265	0.272	0.241	0.238	0.226	0.000	0.192	0.176	0.000	0.153	0.140	0.000	0.123	0.110	0.097	0.103	0.118
0.448	0.456	0.483	0.000	0.490	0.482	0.453	0.443	0.451	0.424	0.414	0.421	0.408	0.000	0.360	0.310	0.263	0.264	0.273	0.255	0.000	0.223	0.227	0.194	0.179	0.177	0.154	0.142	0.142	0.118	0.000	0.104	0.104	0.117
0.447	0.459	0.487	0.490	0.471	0.476	0.449	0.440	0.448	0.422	0.411	0.417	0.394	0.388	0.362	0.313	0.263	0.264	0.276	0.259	0.237	0.230	0.218	0.188	0.174	0.172	0.150	0.138	0.137	0.123	0.108	0.104	0.106	0.117
0.446	0.470	0.000	0.482	0.476	0.000	0.461	0.452	0.000	0.434	0.422	0.000	0.398	0.382	0.000	0.321	0.263	0.263	0.288	0.000	0.257	0.229	0.000	0.198	0.182	0.000	0.159	0.144	0.000	0.125	0.116	0.000	0.110	0.117
0.437	0.448	0.469	0.452	0.449	0.461	0.438	0.430	0.439	0.413	0.401	0.405	0.376	0.359	0.350	0.306	0.258	0.259	0.270	0.246	0.232	0.210	0.209	0.183	0.170	0.169	0.147	0.134	0.132	0.114	0.106	0.098	0.104	0.116
0.429	0.440	0.460	0.443	0.440	0.452	0.431	0.425	0.434	0.407	0.395	0.397	0.369	0.352	0.343	0.301	0.254	0.255	0.266	0.242	0.227	0.206	0.205	0.180	0.168	0.168	0.146	0.133	0.130	0.112	0.104	0.097	0.102	0.115
0.423	0.445	0.000	0.451	0.448	0.000	0.439	0.434	0.000	0.417	0.403	0.000	0.376	0.359	0.000	0.306	0.251	0.252	0.275	0.000	0.239	0.216	0.000	0.191	0.178	0.000	0.155	0.140	0.000	0.119	0.109	0.000	0.107	0.114
0.411	0.421	0.441	0.424	0.422	0.434	0.413	0.407	0.417	0.391	0.379	0.382	0.355	0.339	0.331	0.290	0.245	0.246	0.257	0.234	0.220	0.200	0.200	0.175	0.164	0.164	0.142	0.130	0.128	0.110	0.102	0.095	0.100	0.112
0.400	0.410	0.429	0.414	0.411	0.422	0.401	0.395	0.403	0.379	0.369	0.372	0.346	0.331	0.323	0.284	0.239	0.241	0.252	0.230	0.217	0.197	0.196	0.173	0.161	0.160	0.139	0.128	0.126	0.109	0.101	0.094	0.099	0.111
0.390	0.411	0.000	0.421	0.417	0.000	0.404	0.397	0.000	0.382	0.372	0.000	0.352	0.338	0.000	0.287	0.235	0.236	0.260	0.000	0.233	0.209	0.000	0.182	0.168	0.000	0.147	0.134	0.000	0.117	0.109	0.000	0.103	0.110
0.373	0.383	0.406	0.408	0.394	0.398	0.376	0.369	0.376	0.355	0.347	0.352	0.333	0.329	0.309	0.268	0.226	0.228	0.241	0.228	0.209	0.204	0.194	0.169	0.156	0.156	0.136	0.125	0.125	0.113	0.099	0.096	0.097	0.108
0.354	0.360	0.383	0.000	0.388	0.382	0.359	0.352	0.359	0.339	0.331	0.338	0.329	0.000	0.294	0.254	0.217	0.220	0.230	0.218	0.000	0.193	0.197	0.170	0.157	0.157	0.137	0.127	0.127	0.106	0.000	0.094	0.094	0.106
0.331	0.334	0.343	0.360	0.362	0.000	0.350	0.343	0.000	0.331	0.323	0.000	0.309	0.294	0.265	0.239	0.207	0.213	0.223	0.201	0.202	0.193	0.000	0.166	0.154	0.000	0.135	0.124	0.000	0.110	0.099	0.087	0.093	0.105
0.303	0.304	0.306	0.310	0.312	0.321	0.306	0.301	0.306	0.290	0.284	0.286	0.268	0.255	0.239	0.221	0.196	0.207	0.225	0.203	0.193	0.187	0.187	0.167	0.156	0.154	0.136	0.126	0.126	0.123	0.107	0.096	0.090	0.097	0.107
0.263	0.262	0.262	0.263	0.263	0.263	0.258	0.254	0.251	0.245	0.239	0.235	0.226	0.217	0.207	0.196	0.181	0.209	0.188	0.176	0.168	0.161	0.154	0.145	0.136	0.129	0.119	0.110	0.103	0.094	0.086	0.081	0.085	0.115
0.270	0.267	0.265	0.264	0.264	0.263	0.258	0.255	0.252	0.246	0.241	0.236	0.228	0.220	0.213	0.207	0.209	0.164	0.163	0.159	0.154	0.148	0.142	0.134	0.126	0.119	0.111	0.103	0.096	0.087	0.080	0.076	0.079	0.100
0.220	0.277	0.272	0.273	0.276	0.288	0.270	0.266	0.275	0.257	0.252	0.260	0.241	0.230	0.223	0.225	0.188	0.163	0.170	0.170	0.168	0.164	0.162	0.149	0.140	0.136	0.123	0.114	0.110	0.098	0.089	0.084	0.085	0.105
0.194	0.242	0.241	0.255	0.259	0.000	0.246	0.242	0.000	0.234	0.230	0.000	0.228	0.218	0.202	0.203	0.176	0.159	0.170	0.175	0.180	0.177	0.000	0.159	0.150	0.000	0.132	0.123	0.000	0.106	0.097	0.088	0.088	0.106
0.179	0.225	0.238	0.000	0.237	0.257	0.232	0.227	0.239	0.220	0.217	0.233	0.209	0.000	0.202	0.193	0.168	0.154	0.168	0.180	0.000	0.177	0.169	0.153	0.144	0.141	0.127	0.118	0.115	0.107	0.000	0.092	0.088	0.105
0.168	0.215	0.226	0.223	0.230	0.230	0.210	0.206	0.216	0.200	0.197	0.209	0.204	0.193	0.193	0.187	0.161	0.148	0.164	0.177	0.177	0.166	0.163	0.149	0.140	0.137	0.124	0.115	0.112	0.101	0.097	0.091	0.087	0.102
0.159	0.212	0.000	0.227	0.218	0.000	0.209	0.205	0.000	0.200	0.196	0.000	0.195	0.197	0.000	0.187	0.154	0.142	0.162	0.000	0.169	0.163	0.000	0.149	0.141	0.000	0.125	0.116	0.000	0.100	0.093	0.000	0.087	0.099
0.147	0.188	0.192	0.194	0.188	0.198	0.183	0.180	0.191	0.176	0.173	0.182	0.169	0.170	0.166	0.167	0.145	0.134	0.149	0.159	0.153	0.149	0.149	0.137	0.130	0.127	0.115	0.107	0.103	0.092	0.085	0.084	0.081	0.095
0.137	0.174	0.176	0.179	0.174	0.182	0.170	0.168	0.178	0.164	0.161	0.168	0.156	0.157	0.154	0.156	0.136	0.126	0.140	0.150	0.144	0.140	0.141	0.130	0.124	0.121	0.110	0.102	0.098	0.087	0.081	0.079	0.077	0.090
0.128	0.171	0.000	0.177	0.172	0.000	0.169	0.168	0.000	0.164	0.160	0.000	0.156	0.156	0.000	0.154	0.129	0.119	0.136	0.000	0.141	0.137	0.000	0.127	0.122	0.000	0.108	0.100	0.000	0.086	0.079	0.000	0.075	0.086
0.118	0.150	0.153	0.154	0.150	0.159	0.147	0.146	0.156	0.142	0.139	0.147	0.136	0.137	0.135	0.136	0.119	0.111	0.123	0.132	0.127	0.124	0.125	0.115	0.110	0.108	0.098	0.091	0.087	0.078	0.072	0.071	0.069	0.080
0.108	0.138	0.140	0.142	0.138	0.144	0.134	0.133	0.140	0.130	0.128	0.134	0.125	0.127	0.124	0.126	0.111	0.103	0.114	0.123	0.118	0.115	0.116	0.107	0.102	0.100	0.091	0.084	0.081	0.073	0.067	0.066	0.064	0.075
0.100	0.134	0.000	0.142	0.137	0.000	0.132	0.130	0.000	0.128	0.126	0.000	0.125	0.127	0.000	0.123	0.103	0.096	0.110	0.000	0.115	0.112	0.000	0.103	0.098	0.000	0.087	0.081	0.000	0.071	0.066	0.000	0.062	0.070
0.091	0.116	0.123	0.119	0.123	0.125	0.114	0.112	0.119	0.110	0.109	0.117	0.113	0.106	0.110	0.107	0.094	0.087	0.098	0.106	0.107	0.101	0.100	0.092	0.087	0.086	0.078	0.073	0.071	0.064	0.061	0.058	0.055	0.064
0.083	0.104	0.110	0.000	0.108	0.116	0.106	0.104	0.109	0.102	0.101	0.109	0.099	0.000	0.099	0.096	0.086	0.080	0.089	0.097	0.000	0.097	0.093	0.085	0.081	0.079	0.072	0.067	0.066	0.061	0.000	0.053	0.050	0.059
0.078	0.097	0.097	0.104	0.104	0.000	0.098	0.097	0.000	0.095	0.094	0.000	0.096	0.094	0.087	0.090	0.081	0.076	0.084	0.088	0.092	0.091	0.000	0.084	0.079	0.000	0.071	0.066	0.000	0.058	0.053	0.048	0.047	0.054
0.084	0.106	0.104	0.104	0.106	0.110	0.104	0.102	0.107	0.100	0.099	0.103	0.097	0.094	0.093	0.097	0.085	0.079	0.085	0.088	0.088	0.087	0.087	0.081	0.077	0.075	0.069	0.064	0.062	0.055	0.050	0.046	0.046	0.052
0.120	0.119	0.118	0.117	0.117	0.117	0.116	0.114	0.114	0.112	0.111	0.110	0.108	0.106	0.105	0.107	0.115	0.100	0.105	0.106	0.105	0.102	0.099	0.095	0.090	0.086	0.080	0.075	0.070	0.064	0.059	0.054	0.052	0.056

Table B.6　Unrodded 体系　z=0～42.84cm

2.198	2.202	2.210	2.220	2.223	2.223	2.178	2.141	2.113	2.049	1.994	1.944	1.852	1.753	1.633	1.488	1.287	1.320	1.071	0.945	0.872	0.820	0.775	0.720	0.670	0.627	0.576	0.530	0.489	0.443	0.403	0.381	0.413	0.601
2.202	2.212	2.235	2.269	2.294	2.360	2.244	2.204	2.238	2.110	2.053	2.064	1.913	1.793	1.654	1.495	1.286	1.303	1.352	1.180	1.102	1.055	1.051	0.925	0.857	0.845	0.739	0.678	0.662	0.572	0.509	0.475	0.520	0.593
2.210	2.235	2.304	2.431	2.456	0.000	2.369	2.322	0.000	2.224	2.166	0.000	2.050	1.923	1.709	1.513	1.289	1.296	1.329	1.185	1.180	1.123	0.000	0.955	0.875	0.000	0.760	0.693	0.000	0.610	0.546	0.479	0.511	0.589
2.220	2.269	2.430	0.000	2.481	2.439	2.281	2.232	2.277	2.138	2.086	2.131	2.066	0.000	1.807	1.539	1.296	1.297	1.339	1.267	0.000	1.115	1.132	0.963	0.886	0.881	0.765	0.703	0.707	0.592	0.000	0.516	0.516	0.588
2.223	2.294	2.456	2.481	2.385	2.410	2.265	2.219	2.263	2.125	2.072	2.108	1.989	1.962	1.824	1.559	1.299	1.299	1.364	1.292	1.184	1.148	1.088	0.934	0.861	0.858	0.743	0.683	0.682	0.614	0.541	0.521	0.526	0.588
2.223	2.360	0.000	2.439	2.411	0.000	2.332	2.287	0.000	2.192	2.134	0.000	2.014	1.929	0.000	1.609	1.302	1.300	1.428	0.000	1.283	1.147	0.000	0.989	0.907	0.000	0.790	0.719	0.000	0.624	0.580	0.000	0.552	0.588
2.178	2.244	2.369	2.281	2.265	2.332	2.205	2.166	2.212	2.076	2.020	2.044	1.894	1.807	1.761	1.528	1.277	1.277	1.337	1.227	1.152	1.043	1.041	0.906	0.839	0.839	0.726	0.665	0.658	0.565	0.525	0.491	0.516	0.581
2.141	2.204	2.321	2.232	2.218	2.287	2.166	2.130	2.176	2.042	1.986	2.007	1.858	1.771	1.729	1.504	1.259	1.259	1.316	1.203	1.128	1.024	1.022	0.893	0.829	0.829	0.718	0.657	0.648	0.557	0.516	0.483	0.510	0.576
2.113	2.239	0.000	2.277	2.263	0.000	2.213	2.176	0.000	2.089	2.030	0.000	1.900	1.809	0.000	1.535	1.247	1.246	1.365	0.000	1.193	1.078	0.000	0.949	0.876	0.000	0.765	0.694	0.000	0.593	0.544	0.000	0.533	0.572
2.049	2.110	2.224	2.138	2.125	2.192	2.076	2.042	2.089	1.959	1.907	1.929	1.786	1.703	1.665	1.449	1.213	1.215	1.273	1.167	1.094	0.994	0.996	0.869	0.807	0.809	0.701	0.642	0.634	0.544	0.505	0.473	0.500	0.563
1.994	2.053	2.166	2.086	2.072	2.134	2.020	1.986	2.030	1.907	1.858	1.881	1.745	1.668	1.627	1.416	1.187	1.189	1.248	1.145	1.079	0.980	0.979	0.855	0.794	0.794	0.689	0.633	0.626	0.539	0.502	0.469	0.494	0.557
1.944	2.064	0.000	2.131	2.108	0.000	2.044	2.007	0.000	1.929	1.881	0.000	1.781	1.709	0.000	1.435	1.165	1.168	1.289	0.000	1.165	1.046	0.000	0.909	0.837	0.000	0.733	0.668	0.000	0.582	0.542	0.000	0.517	0.552
1.852	1.913	2.049	2.066	1.989	2.014	1.894	1.858	1.900	1.786	1.745	1.781	1.683	1.665	1.558	1.337	1.119	1.125	1.192	1.138	1.044	1.018	0.972	0.837	0.775	0.777	0.675	0.622	0.623	0.561	0.495	0.478	0.484	0.541
1.752	1.792	1.923	0.000	1.961	1.929	1.807	1.771	1.809	1.703	1.667	1.709	1.665	0.000	1.473	1.263	1.071	1.082	1.132	1.083	0.000	0.965	0.985	0.843	0.781	0.780	0.680	0.628	0.633	0.532	0.000	0.467	0.467	0.532
1.633	1.654	1.709	1.806	1.824	0.000	1.761	1.729	0.000	1.665	1.627	0.000	1.557	1.473	1.318	1.178	1.018	1.041	1.090	0.991	1.001	0.961	0.000	0.828	0.765	0.000	0.672	0.616	0.000	0.547	0.491	0.432	0.460	0.527
1.488	1.495	1.513	1.539	1.559	1.608	1.528	1.504	1.535	1.449	1.417	1.435	1.337	1.263	1.178	1.085	0.959	1.012	1.099	0.992	0.947	0.921	0.926	0.824	0.769	0.763	0.673	0.621	0.609	0.529	0.474	0.442	0.480	0.535
1.286	1.286	1.289	1.296	1.299	1.302	1.277	1.259	1.247	1.213	1.186	1.165	1.119	1.071	1.018	0.959	0.884	1.019	0.916	0.861	0.824	0.791	0.760	0.714	0.672	0.634	0.587	0.544	0.506	0.461	0.422	0.399	0.422	0.575
1.320	1.303	1.296	1.297	1.299	1.299	1.277	1.259	1.246	1.215	1.189	1.168	1.125	1.082	1.041	1.012	1.019	0.801	0.796	0.777	0.755	0.729	0.703	0.662	0.624	0.590	0.547	0.508	0.473	0.431	0.396	0.376	0.391	0.501
1.070	1.352	1.329	1.339	1.364	1.428	1.337	1.316	1.365	1.273	1.248	1.289	1.192	1.131	1.090	1.099	0.916	0.796	0.832	0.835	0.828	0.813	0.809	0.741	0.698	0.681	0.614	0.570	0.548	0.487	0.443	0.415	0.425	0.527
0.945	1.180	1.185	1.267	1.292	0.000	1.227	1.203	0.000	1.167	1.145	0.000	1.138	1.083	0.991	0.992	0.861	0.777	0.835	0.866	0.899	0.885	0.000	0.800	0.752	0.000	0.664	0.616	0.000	0.536	0.489	0.440	0.438	0.532
0.872	1.102	1.180	0.000	1.184	1.283	1.152	1.128	1.193	1.094	1.079	1.165	1.044	0.000	1.001	0.947	0.824	0.755	0.828	0.899	0.000	0.889	0.849	0.767	0.722	0.708	0.636	0.593	0.579	0.538	0.000	0.464	0.443	0.528
0.820	1.055	1.123	1.115	1.148	1.146	1.043	1.024	1.078	0.994	0.980	1.046	1.018	0.965	0.961	0.920	0.791	0.729	0.813	0.885	0.889	0.837	0.822	0.747	0.705	0.691	0.622	0.580	0.564	0.511	0.491	0.461	0.440	0.517
0.775	1.051	0.000	1.132	1.088	0.000	1.041	1.022	0.000	0.996	0.980	0.000	0.972	0.985	0.000	0.926	0.760	0.703	0.809	0.000	0.849	0.822	0.000	0.750	0.708	0.000	0.628	0.584	0.000	0.507	0.471	0.000	0.442	0.503
0.720	0.925	0.955	0.963	0.934	0.989	0.906	0.893	0.949	0.869	0.855	0.909	0.837	0.843	0.828	0.824	0.714	0.662	0.741	0.800	0.767	0.747	0.750	0.687	0.651	0.641	0.577	0.538	0.521	0.464	0.430	0.424	0.410	0.481
0.670	0.857	0.875	0.886	0.861	0.907	0.839	0.829	0.875	0.807	0.794	0.837	0.775	0.781	0.765	0.770	0.672	0.624	0.698	0.753	0.722	0.705	0.708	0.651	0.618	0.608	0.549	0.511	0.494	0.440	0.407	0.402	0.390	0.458
0.627	0.845	0.000	0.881	0.858	0.000	0.839	0.829	0.000	0.809	0.794	0.000	0.777	0.780	0.000	0.763	0.634	0.590	0.680	0.000	0.708	0.691	0.000	0.641	0.608	0.000	0.542	0.503	0.000	0.434	0.401	0.000	0.382	0.436
0.576	0.739	0.760	0.765	0.743	0.790	0.726	0.718	0.765	0.700	0.689	0.733	0.674	0.680	0.672	0.673	0.587	0.547	0.614	0.664	0.636	0.622	0.628	0.577	0.549	0.542	0.489	0.455	0.441	0.392	0.363	0.359	0.348	0.408
0.530	0.678	0.693	0.703	0.683	0.719	0.665	0.657	0.694	0.642	0.633	0.668	0.622	0.628	0.616	0.621	0.544	0.508	0.570	0.616	0.593	0.580	0.584	0.537	0.511	0.503	0.455	0.424	0.411	0.367	0.340	0.335	0.325	0.381
0.489	0.662	0.000	0.707	0.682	0.000	0.658	0.648	0.000	0.634	0.626	0.000	0.623	0.634	0.000	0.609	0.506	0.473	0.547	0.000	0.579	0.564	0.000	0.521	0.494	0.000	0.441	0.411	0.000	0.358	0.333	0.000	0.313	0.356
0.443	0.572	0.610	0.592	0.614	0.624	0.565	0.557	0.593	0.544	0.539	0.582	0.561	0.532	0.547	0.529	0.461	0.431	0.487	0.536	0.538	0.511	0.507	0.463	0.440	0.434	0.392	0.367	0.358	0.324	0.311	0.294	0.280	0.326
0.403	0.509	0.546	0.000	0.541	0.580	0.525	0.516	0.544	0.505	0.502	0.542	0.495	0.000	0.491	0.474	0.422	0.396	0.443	0.489	0.000	0.491	0.471	0.430	0.407	0.401	0.363	0.340	0.333	0.311	0.000	0.269	0.254	0.298
0.382	0.475	0.479	0.516	0.520	0.000	0.491	0.483	0.000	0.473	0.469	0.000	0.478	0.467	0.432	0.442	0.399	0.376	0.415	0.440	0.464	0.461	0.000	0.424	0.402	0.000	0.359	0.335	0.000	0.294	0.269	0.240	0.235	0.275
0.413	0.520	0.511	0.516	0.526	0.551	0.516	0.510	0.533	0.500	0.494	0.517	0.484	0.467	0.460	0.480	0.422	0.391	0.425	0.438	0.443	0.440	0.442	0.410	0.390	0.382	0.348	0.325	0.313	0.280	0.254	0.235	0.230	0.264
0.601	0.593	0.589	0.588	0.588	0.588	0.581	0.576	0.572	0.563	0.557	0.552	0.541	0.532	0.527	0.535	0.575	0.501	0.527	0.532	0.528	0.518	0.504	0.481	0.458	0.436	0.408	0.381	0.356	0.326	0.298	0.275	0.264	0.284

Table B.7　Rodded A 体系 z=0～14.28cm

1.051	1.053	1.058	1.064	1.066	1.067	1.046	1.029	1.016	0.986	0.960	0.937	0.893	0.845	0.787	0.717	0.620	0.637	0.517	0.457	0.423	0.398	0.378	0.351	0.327	0.307	0.282	0.260	0.240	0.218	0.198	0.188	0.204	0.298
1.053	1.059	1.071	1.089	1.102	1.136	1.080	1.061	1.079	1.017	0.990	0.997	0.924	0.865	0.798	0.721	0.621	0.629	0.654	0.572	0.535	0.514	0.513	0.452	0.419	0.414	0.363	0.333	0.326	0.282	0.251	0.234	0.257	0.294
1.058	1.071	1.106	1.169	1.183	0.000	1.142	1.119	0.000	1.074	1.046	0.000	0.992	0.930	0.826	0.731	0.623	0.626	0.643	0.575	0.575	0.549	0.000	0.468	0.429	0.000	0.374	0.341	0.000	0.302	0.270	0.237	0.253	0.292
1.064	1.088	1.169	0.000	1.196	1.176	1.099	1.076	1.099	1.032	1.008	1.031	1.001	0.000	0.875	0.744	0.626	0.627	0.649	0.617	0.000	0.546	0.554	0.472	0.435	0.433	0.376	0.346	0.349	0.293	0.000	0.256	0.255	0.292
1.066	1.102	1.183	1.196	1.150	1.163	1.092	1.070	1.093	1.026	1.002	1.021	0.964	0.951	0.885	0.755	0.629	0.629	0.662	0.630	0.579	0.562	0.533	0.458	0.423	0.422	0.366	0.337	0.337	0.304	0.268	0.258	0.261	0.292
1.067	1.136	0.000	1.176	1.163	0.000	1.126	1.104	0.000	1.060	1.033	0.000	0.976	0.936	0.000	0.780	0.631	0.630	0.694	0.000	0.627	0.561	0.000	0.486	0.446	0.000	0.390	0.355	0.000	0.309	0.288	0.000	0.274	0.293
1.046	1.080	1.142	1.100	1.092	1.125	1.064	1.045	1.069	1.003	0.977	0.990	0.918	0.876	0.855	0.741	0.619	0.619	0.650	0.598	0.563	0.511	0.511	0.444	0.412	0.413	0.357	0.328	0.325	0.280	0.260	0.244	0.256	0.289
1.029	1.061	1.119	1.076	1.070	1.104	1.045	1.027	1.051	0.986	0.961	0.972	0.901	0.859	0.840	0.730	0.610	0.611	0.640	0.587	0.551	0.501	0.502	0.438	0.407	0.407	0.353	0.324	0.321	0.276	0.256	0.240	0.254	0.287
1.017	1.079	0.000	1.099	1.093	0.000	1.069	1.051	0.000	1.009	0.983	0.000	0.922	0.879	0.000	0.746	0.605	0.605	0.665	0.000	0.584	0.529	0.000	0.466	0.430	0.000	0.377	0.343	0.000	0.294	0.270	0.000	0.265	0.285
0.986	1.017	1.074	1.032	1.026	1.060	1.003	0.986	1.010	0.948	0.924	0.936	0.867	0.827	0.810	0.704	0.589	0.590	0.620	0.571	0.536	0.487	0.489	0.427	0.397	0.398	0.345	0.317	0.314	0.270	0.251	0.235	0.249	0.281
0.960	0.990	1.046	1.008	1.002	1.033	0.977	0.961	0.983	0.924	0.901	0.913	0.848	0.811	0.792	0.689	0.577	0.578	0.608	0.561	0.529	0.481	0.482	0.421	0.391	0.392	0.340	0.313	0.311	0.268	0.249	0.233	0.246	0.278
0.937	0.997	0.000	1.032	1.021	0.000	0.990	0.972	0.000	0.936	0.913	0.000	0.867	0.832	0.000	0.699	0.567	0.568	0.630	0.000	0.572	0.515	0.000	0.448	0.413	0.000	0.363	0.331	0.000	0.289	0.270	0.000	0.258	0.276
0.893	0.924	0.992	1.001	0.964	0.976	0.918	0.901	0.922	0.867	0.848	0.867	0.820	0.812	0.760	0.651	0.545	0.548	0.582	0.558	0.513	0.501	0.479	0.413	0.383	0.385	0.334	0.308	0.310	0.279	0.247	0.239	0.241	0.271
0.845	0.865	0.930	0.000	0.951	0.936	0.876	0.859	0.879	0.827	0.811	0.832	0.812	0.000	0.718	0.615	0.521	0.527	0.552	0.531	0.000	0.476	0.486	0.416	0.386	0.386	0.337	0.312	0.315	0.265	0.000	0.233	0.233	0.266
0.787	0.798	0.826	0.875	0.885	0.000	0.855	0.840	0.000	0.810	0.792	0.000	0.760	0.718	0.642	0.574	0.495	0.507	0.532	0.485	0.492	0.474	0.000	0.409	0.379	0.000	0.333	0.306	0.000	0.273	0.245	0.215	0.229	0.264
0.717	0.721	0.731	0.745	0.755	0.780	0.741	0.730	0.746	0.704	0.689	0.699	0.651	0.615	0.574	0.528	0.467	0.494	0.537	0.486	0.465	0.453	0.457	0.407	0.381	0.378	0.334	0.309	0.303	0.264	0.236	0.220	0.239	0.268
0.620	0.621	0.622	0.626	0.629	0.631	0.619	0.610	0.605	0.589	0.577	0.567	0.545	0.521	0.495	0.467	0.431	0.498	0.448	0.422	0.405	0.390	0.375	0.353	0.332	0.314	0.291	0.270	0.252	0.230	0.210	0.199	0.211	0.288
0.637	0.629	0.626	0.627	0.629	0.630	0.619	0.611	0.605	0.590	0.578	0.568	0.548	0.527	0.507	0.494	0.498	0.391	0.390	0.381	0.371	0.360	0.347	0.328	0.309	0.293	0.272	0.253	0.235	0.215	0.198	0.187	0.196	0.251
0.517	0.654	0.643	0.649	0.662	0.694	0.650	0.640	0.665	0.620	0.608	0.630	0.582	0.552	0.532	0.537	0.448	0.390	0.408	0.411	0.409	0.402	0.401	0.368	0.347	0.339	0.306	0.284	0.274	0.243	0.222	0.208	0.213	0.265
0.457	0.572	0.575	0.617	0.630	0.000	0.598	0.587	0.000	0.570	0.561	0.000	0.558	0.531	0.486	0.486	0.422	0.381	0.411	0.428	0.445	0.440	0.000	0.398	0.375	0.000	0.332	0.308	0.000	0.269	0.245	0.221	0.220	0.268
0.423	0.535	0.575	0.000	0.579	0.627	0.562	0.552	0.584	0.536	0.529	0.572	0.513	0.000	0.492	0.465	0.405	0.371	0.409	0.445	0.000	0.442	0.423	0.382	0.360	0.354	0.318	0.297	0.291	0.271	0.000	0.233	0.223	0.266
0.399	0.514	0.549	0.546	0.562	0.561	0.510	0.501	0.529	0.487	0.481	0.515	0.501	0.476	0.474	0.453	0.389	0.360	0.402	0.440	0.442	0.417	0.410	0.373	0.352	0.346	0.312	0.291	0.283	0.257	0.247	0.232	0.222	0.261
0.378	0.513	0.000	0.555	0.533	0.000	0.511	0.502	0.000	0.489	0.482	0.000	0.479	0.486	0.000	0.457	0.375	0.347	0.401	0.000	0.423	0.410	0.000	0.375	0.354	0.000	0.315	0.293	0.000	0.255	0.237	0.000	0.223	0.254
0.351	0.452	0.468	0.472	0.458	0.486	0.444	0.438	0.466	0.427	0.421	0.448	0.413	0.416	0.409	0.407	0.353	0.328	0.367	0.398	0.382	0.373	0.375	0.344	0.326	0.321	0.290	0.270	0.262	0.233	0.217	0.214	0.207	0.243
0.327	0.419	0.429	0.435	0.422	0.446	0.412	0.407	0.430	0.397	0.391	0.413	0.383	0.386	0.379	0.381	0.332	0.309	0.347	0.375	0.360	0.352	0.354	0.326	0.310	0.305	0.275	0.257	0.249	0.222	0.205	0.203	0.197	0.232
0.307	0.415	0.000	0.433	0.422	0.000	0.413	0.407	0.000	0.398	0.392	0.000	0.384	0.386	0.000	0.378	0.314	0.293	0.339	0.000	0.354	0.346	0.000	0.321	0.305	0.000	0.272	0.253	0.000	0.219	0.203	0.000	0.194	0.221
0.282	0.363	0.374	0.376	0.366	0.389	0.357	0.353	0.377	0.345	0.340	0.363	0.334	0.337	0.333	0.334	0.291	0.272	0.306	0.332	0.318	0.312	0.315	0.290	0.275	0.272	0.246	0.229	0.223	0.198	0.184	0.182	0.176	0.207
0.260	0.333	0.341	0.346	0.336	0.355	0.328	0.324	0.343	0.317	0.313	0.331	0.308	0.312	0.306	0.309	0.270	0.253	0.284	0.308	0.297	0.291	0.293	0.270	0.257	0.253	0.229	0.214	0.208	0.185	0.172	0.170	0.165	0.194
0.240	0.326	0.000	0.349	0.337	0.000	0.325	0.321	0.000	0.314	0.310	0.000	0.310	0.315	0.000	0.303	0.252	0.235	0.274	0.000	0.291	0.283	0.000	0.262	0.249	0.000	0.223	0.208	0.000	0.181	0.169	0.000	0.159	0.181
0.218	0.282	0.302	0.293	0.304	0.309	0.280	0.276	0.294	0.270	0.268	0.289	0.279	0.265	0.273	0.263	0.230	0.215	0.243	0.269	0.271	0.257	0.255	0.233	0.222	0.219	0.198	0.185	0.181	0.164	0.158	0.149	0.142	0.166
0.198	0.251	0.270	0.000	0.268	0.288	0.260	0.256	0.270	0.251	0.249	0.270	0.247	0.000	0.245	0.236	0.210	0.198	0.222	0.245	0.000	0.247	0.238	0.217	0.206	0.203	0.184	0.172	0.169	0.158	0.000	0.136	0.129	0.152
0.188	0.234	0.237	0.256	0.258	0.000	0.244	0.240	0.000	0.235	0.233	0.000	0.239	0.233	0.215	0.220	0.199	0.187	0.208	0.221	0.233	0.232	0.000	0.214	0.203	0.000	0.182	0.170	0.000	0.149	0.136	0.122	0.119	0.140
0.204	0.257	0.253	0.255	0.261	0.274	0.256	0.254	0.265	0.249	0.246	0.258	0.241	0.233	0.229	0.239	0.211	0.196	0.213	0.220	0.223	0.222	0.223	0.207	0.197	0.194	0.176	0.165	0.159	0.142	0.129	0.119	0.117	0.134
0.298	0.294	0.292	0.292	0.292	0.292	0.289	0.287	0.285	0.281	0.278	0.276	0.271	0.266	0.264	0.268	0.288	0.251	0.265	0.268	0.266	0.261	0.254	0.243	0.232	0.221	0.207	0.194	0.181	0.166	0.152	0.140	0.134	0.145

Table B.8　Rodded A 体系 z=14.28～28.56cm

0.740	0.742	0.744	0.747	0.747	0.748	0.733	0.722	0.713	0.693	0.676	0.661	0.632	0.600	0.562	0.514	0.447	0.461	0.376	0.335	0.311	0.294	0.280	0.261	0.244	0.230	0.212	0.196	0.181	0.165	0.150	0.143	0.155	0.227
0.741	0.745	0.752	0.763	0.771	0.793	0.755	0.742	0.755	0.713	0.696	0.701	0.652	0.613	0.569	0.517	0.447	0.455	0.476	0.418	0.394	0.380	0.381	0.336	0.313	0.310	0.272	0.251	0.246	0.213	0.190	0.178	0.195	0.224
0.744	0.752	0.774	0.815	0.824	0.000	0.795	0.781	0.000	0.751	0.733	0.000	0.697	0.657	0.587	0.523	0.448	0.453	0.468	0.421	0.423	0.406	0.000	0.348	0.321	0.000	0.281	0.257	0.000	0.228	0.205	0.180	0.192	0.223
0.747	0.763	0.815	0.000	0.831	0.818	0.767	0.751	0.767	0.722	0.706	0.723	0.703	0.000	0.619	0.531	0.450	0.453	0.472	0.452	0.000	0.403	0.411	0.351	0.325	0.325	0.283	0.261	0.264	0.222	0.000	0.194	0.195	0.223
0.748	0.771	0.824	0.832	0.800	0.809	0.761	0.747	0.763	0.718	0.702	0.715	0.677	0.670	0.626	0.538	0.451	0.454	0.482	0.461	0.426	0.415	0.395	0.341	0.316	0.316	0.275	0.254	0.255	0.230	0.203	0.196	0.199	0.223
0.748	0.793	0.000	0.818	0.809	0.000	0.784	0.770	0.000	0.741	0.723	0.000	0.686	0.659	0.000	0.555	0.452	0.455	0.505	0.000	0.461	0.415	0.000	0.362	0.333	0.000	0.293	0.268	0.000	0.234	0.218	0.000	0.209	0.223
0.733	0.755	0.796	0.767	0.761	0.784	0.743	0.731	0.748	0.703	0.686	0.695	0.646	0.619	0.605	0.528	0.444	0.448	0.473	0.438	0.414	0.377	0.379	0.331	0.308	0.310	0.269	0.247	0.246	0.212	0.198	0.185	0.195	0.221
0.722	0.742	0.781	0.751	0.747	0.770	0.731	0.720	0.738	0.693	0.676	0.683	0.635	0.608	0.595	0.521	0.438	0.442	0.466	0.430	0.406	0.371	0.373	0.327	0.304	0.306	0.266	0.245	0.243	0.209	0.194	0.183	0.193	0.219
0.713	0.755	0.000	0.767	0.763	0.000	0.748	0.738	0.000	0.711	0.691	0.000	0.650	0.621	0.000	0.532	0.435	0.438	0.485	0.000	0.430	0.392	0.000	0.348	0.322	0.000	0.284	0.259	0.000	0.223	0.205	0.000	0.202	0.218
0.693	0.713	0.751	0.722	0.718	0.741	0.703	0.693	0.711	0.668	0.651	0.659	0.613	0.587	0.575	0.504	0.424	0.428	0.453	0.419	0.395	0.361	0.364	0.319	0.298	0.300	0.260	0.240	0.238	0.205	0.191	0.179	0.190	0.215
0.676	0.696	0.733	0.706	0.702	0.723	0.686	0.676	0.691	0.651	0.636	0.644	0.600	0.576	0.564	0.493	0.416	0.420	0.445	0.412	0.391	0.357	0.359	0.315	0.293	0.295	0.257	0.237	0.235	0.203	0.190	0.178	0.188	0.212
0.661	0.701	0.000	0.723	0.715	0.000	0.695	0.683	0.000	0.659	0.644	0.000	0.613	0.591	0.000	0.501	0.409	0.413	0.462	0.000	0.424	0.383	0.000	0.336	0.310	0.000	0.274	0.251	0.000	0.220	0.206	0.000	0.197	0.211
0.632	0.652	0.697	0.703	0.677	0.686	0.646	0.635	0.650	0.613	0.600	0.613	0.582	0.577	0.542	0.468	0.395	0.400	0.427	0.412	0.381	0.373	0.358	0.310	0.288	0.290	0.253	0.234	0.235	0.213	0.188	0.182	0.184	0.207
0.600	0.613	0.657	0.000	0.670	0.659	0.619	0.608	0.621	0.586	0.576	0.591	0.577	0.000	0.515	0.444	0.379	0.386	0.407	0.393	0.000	0.355	0.364	0.313	0.291	0.292	0.255	0.237	0.240	0.202	0.000	0.178	0.178	0.204
0.562	0.569	0.587	0.620	0.626	0.000	0.605	0.595	0.000	0.575	0.564	0.000	0.542	0.515	0.463	0.416	0.362	0.372	0.393	0.360	0.367	0.354	0.000	0.308	0.286	0.000	0.253	0.233	0.000	0.208	0.187	0.164	0.176	0.202
0.514	0.517	0.523	0.531	0.538	0.555	0.528	0.521	0.532	0.504	0.493	0.501	0.468	0.444	0.417	0.385	0.343	0.364	0.398	0.362	0.347	0.340	0.344	0.307	0.288	0.287	0.253	0.235	0.231	0.201	0.180	0.169	0.184	0.206
0.447	0.447	0.448	0.450	0.451	0.452	0.444	0.439	0.435	0.424	0.416	0.409	0.395	0.379	0.362	0.343	0.317	0.369	0.333	0.315	0.303	0.293	0.282	0.267	0.252	0.239	0.222	0.206	0.192	0.175	0.161	0.153	0.162	0.221
0.461	0.455	0.453	0.453	0.454	0.455	0.448	0.442	0.438	0.428	0.420	0.413	0.400	0.386	0.372	0.364	0.368	0.291	0.291	0.286	0.279	0.271	0.262	0.248	0.235	0.223	0.207	0.193	0.180	0.165	0.152	0.144	0.150	0.193
0.377	0.476	0.468	0.472	0.482	0.506	0.473	0.466	0.485	0.453	0.445	0.461	0.427	0.407	0.393	0.398	0.333	0.291	0.306	0.309	0.308	0.303	0.303	0.279	0.264	0.258	0.233	0.217	0.209	0.186	0.170	0.159	0.164	0.204
0.335	0.418	0.421	0.452	0.461	0.000	0.438	0.430	0.000	0.419	0.412	0.000	0.412	0.393	0.360	0.362	0.315	0.286	0.309	0.322	0.336	0.332	0.000	0.302	0.285	0.000	0.253	0.236	0.000	0.206	0.188	0.170	0.169	0.206
0.311	0.394	0.423	0.000	0.426	0.461	0.414	0.406	0.431	0.395	0.391	0.424	0.381	0.000	0.367	0.347	0.303	0.279	0.308	0.336	0.000	0.335	0.321	0.291	0.274	0.270	0.243	0.227	0.223	0.208	0.000	0.179	0.172	0.205
0.294	0.380	0.406	0.403	0.415	0.415	0.378	0.371	0.392	0.361	0.357	0.383	0.373	0.355	0.355	0.340	0.293	0.271	0.303	0.332	0.335	0.316	0.312	0.284	0.268	0.264	0.239	0.223	0.217	0.198	0.190	0.179	0.171	0.201
0.280	0.381	0.000	0.411	0.396	0.000	0.379	0.373	0.000	0.364	0.359	0.000	0.358	0.364	0.000	0.344	0.282	0.262	0.303	0.000	0.321	0.312	0.000	0.286	0.271	0.000	0.241	0.225	0.000	0.196	0.183	0.000	0.172	0.196
0.261	0.336	0.348	0.351	0.341	0.362	0.331	0.327	0.348	0.319	0.315	0.336	0.310	0.313	0.308	0.307	0.267	0.248	0.279	0.302	0.291	0.284	0.286	0.263	0.249	0.246	0.222	0.207	0.202	0.180	0.167	0.165	0.160	0.188
0.244	0.313	0.321	0.325	0.316	0.333	0.308	0.305	0.322	0.297	0.294	0.311	0.288	0.291	0.286	0.288	0.252	0.235	0.264	0.285	0.274	0.268	0.271	0.249	0.237	0.234	0.211	0.197	0.192	0.171	0.158	0.157	0.152	0.179
0.230	0.311	0.000	0.325	0.316	0.000	0.310	0.306	0.000	0.300	0.295	0.000	0.290	0.292	0.000	0.287	0.239	0.223	0.258	0.000	0.270	0.264	0.000	0.246	0.234	0.000	0.209	0.195	0.000	0.169	0.156	0.000	0.150	0.171
0.212	0.272	0.281	0.283	0.275	0.293	0.269	0.266	0.284	0.260	0.257	0.274	0.252	0.255	0.253	0.253	0.222	0.207	0.233	0.253	0.243	0.238	0.241	0.222	0.211	0.209	0.189	0.177	0.172	0.153	0.142	0.140	0.136	0.160
0.196	0.251	0.257	0.261	0.254	0.268	0.247	0.245	0.259	0.240	0.237	0.251	0.234	0.237	0.233	0.235	0.206	0.193	0.217	0.236	0.227	0.223	0.225	0.207	0.197	0.195	0.177	0.165	0.160	0.143	0.133	0.131	0.127	0.150
0.181	0.246	0.000	0.264	0.255	0.000	0.246	0.243	0.000	0.238	0.235	0.000	0.235	0.240	0.000	0.231	0.192	0.180	0.209	0.000	0.223	0.218	0.000	0.202	0.192	0.000	0.172	0.160	0.000	0.140	0.130	0.000	0.123	0.140
0.165	0.213	0.228	0.222	0.230	0.234	0.212	0.209	0.223	0.205	0.203	0.220	0.213	0.202	0.208	0.201	0.175	0.165	0.186	0.206	0.208	0.198	0.196	0.180	0.171	0.169	0.153	0.143	0.140	0.127	0.122	0.115	0.110	0.128
0.150	0.190	0.205	0.000	0.203	0.218	0.198	0.194	0.205	0.191	0.190	0.206	0.188	0.000	0.187	0.180	0.161	0.152	0.170	0.188	0.000	0.190	0.183	0.167	0.158	0.156	0.142	0.133	0.130	0.122	0.000	0.106	0.100	0.118
0.143	0.178	0.180	0.194	0.196	0.000	0.185	0.183	0.000	0.179	0.178	0.000	0.182	0.178	0.164	0.169	0.153	0.144	0.159	0.170	0.179	0.179	0.000	0.165	0.157	0.000	0.140	0.131	0.000	0.115	0.106	0.095	0.092	0.108
0.155	0.195	0.192	0.194	0.199	0.209	0.195	0.193	0.202	0.190	0.188	0.197	0.184	0.178	0.176	0.184	0.162	0.150	0.164	0.169	0.172	0.171	0.172	0.160	0.152	0.150	0.136	0.127	0.123	0.110	0.100	0.092	0.091	0.104
0.227	0.224	0.223	0.223	0.223	0.223	0.221	0.219	0.218	0.215	0.212	0.211	0.207	0.204	0.202	0.206	0.221	0.193	0.204	0.206	0.205	0.201	0.196	0.188	0.179	0.171	0.160	0.150	0.140	0.128	0.118	0.108	0.104	0.112

Table B.9　Rodded A 体系 z=28.56～42.84cm

0.305	0.302	0.295	0.287	0.278	0.269	0.266	0.261	0.255	0.253	0.248	0.243	0.241	0.239	0.234	0.223	0.201	0.213	0.180	0.164	0.154	0.148	0.142	0.134	0.126	0.120	0.111	0.103	0.096	0.088	0.081	0.077	0.083	0.120
0.302	0.297	0.288	0.276	0.263	0.247	0.253	0.249	0.235	0.240	0.236	0.223	0.228	0.230	0.228	0.220	0.199	0.209	0.226	0.204	0.194	0.189	0.191	0.171	0.161	0.160	0.142	0.132	0.129	0.113	0.102	0.096	0.105	0.118
0.295	0.288	0.272	0.248	0.235	0.000	0.231	0.229	0.000	0.220	0.217	0.000	0.204	0.206	0.215	0.213	0.195	0.206	0.221	0.203	0.206	0.199	0.000	0.175	0.163	0.000	0.145	0.134	0.000	0.119	0.108	0.096	0.103	0.118
0.287	0.276	0.248	0.000	0.224	0.223	0.236	0.234	0.219	0.227	0.221	0.202	0.196	0.000	0.196	0.204	0.190	0.203	0.220	0.215	0.000	0.196	0.203	0.177	0.165	0.166	0.146	0.136	0.137	0.116	0.000	0.102	0.103	0.117
0.278	0.263	0.235	0.224	0.229	0.222	0.234	0.232	0.217	0.224	0.219	0.201	0.201	0.188	0.187	0.196	0.186	0.200	0.223	0.218	0.204	0.203	0.196	0.172	0.161	0.162	0.142	0.132	0.132	0.120	0.106	0.103	0.105	0.117
0.269	0.247	0.000	0.223	0.222	0.000	0.221	0.221	0.000	0.214	0.209	0.000	0.194	0.189	0.000	0.184	0.181	0.198	0.232	0.000	0.221	0.203	0.000	0.181	0.169	0.000	0.151	0.139	0.000	0.122	0.114	0.000	0.110	0.117
0.266	0.253	0.231	0.236	0.234	0.221	0.234	0.236	0.227	0.229	0.220	0.201	0.205	0.200	0.186	0.190	0.180	0.195	0.217	0.207	0.200	0.186	0.188	0.167	0.158	0.159	0.140	0.129	0.128	0.111	0.104	0.098	0.103	0.116
0.261	0.249	0.229	0.234	0.232	0.221	0.236	0.249	0.257	0.242	0.223	0.201	0.204	0.199	0.184	0.188	0.178	0.193	0.214	0.203	0.197	0.183	0.186	0.166	0.157	0.159	0.139	0.128	0.127	0.110	0.103	0.096	0.102	0.115
0.255	0.235	0.000	0.219	0.217	0.000	0.227	0.257	0.000	0.251	0.215	0.000	0.191	0.187	0.000	0.177	0.174	0.191	0.223	0.000	0.207	0.193	0.000	0.176	0.166	0.000	0.148	0.135	0.000	0.117	0.108	0.000	0.107	0.115
0.253	0.240	0.220	0.227	0.224	0.214	0.229	0.242	0.251	0.235	0.217	0.195	0.198	0.193	0.179	0.182	0.173	0.188	0.209	0.199	0.192	0.179	0.182	0.162	0.154	0.156	0.136	0.126	0.125	0.108	0.101	0.095	0.101	0.113
0.248	0.236	0.217	0.221	0.219	0.209	0.220	0.223	0.215	0.217	0.208	0.191	0.194	0.189	0.176	0.180	0.171	0.185	0.207	0.196	0.191	0.177	0.180	0.160	0.151	0.153	0.134	0.124	0.123	0.107	0.101	0.094	0.100	0.112
0.243	0.223	0.000	0.202	0.201	0.000	0.201	0.201	0.000	0.195	0.191	0.000	0.178	0.173	0.000	0.169	0.167	0.183	0.215	0.000	0.206	0.190	0.000	0.170	0.159	0.000	0.142	0.131	0.000	0.115	0.108	0.000	0.104	0.111
0.241	0.228	0.204	0.196	0.201	0.194	0.205	0.204	0.191	0.198	0.194	0.178	0.178	0.168	0.166	0.175	0.166	0.180	0.202	0.198	0.186	0.186	0.180	0.158	0.149	0.150	0.132	0.123	0.123	0.112	0.099	0.096	0.098	0.109
0.239	0.230	0.206	0.000	0.188	0.189	0.200	0.199	0.187	0.193	0.189	0.173	0.168	0.000	0.168	0.177	0.166	0.178	0.195	0.192	0.000	0.177	0.184	0.160	0.151	0.151	0.134	0.125	0.126	0.106	0.000	0.095	0.095	0.108
0.234	0.228	0.215	0.196	0.187	0.000	0.186	0.184	0.000	0.179	0.176	0.000	0.166	0.168	0.177	0.178	0.164	0.176	0.192	0.179	0.184	0.179	0.000	0.158	0.148	0.000	0.132	0.123	0.000	0.110	0.100	0.088	0.095	0.107
0.223	0.220	0.213	0.204	0.196	0.184	0.190	0.188	0.177	0.183	0.180	0.169	0.175	0.177	0.178	0.174	0.161	0.177	0.198	0.183	0.178	0.175	0.177	0.160	0.151	0.150	0.134	0.125	0.123	0.108	0.097	0.091	0.099	0.110
0.201	0.199	0.195	0.190	0.186	0.181	0.180	0.178	0.174	0.173	0.171	0.167	0.166	0.166	0.164	0.161	0.154	0.183	0.169	0.162	0.157	0.152	0.148	0.140	0.133	0.127	0.118	0.110	0.103	0.095	0.087	0.083	0.087	0.118
0.213	0.209	0.206	0.203	0.200	0.198	0.195	0.193	0.191	0.188	0.185	0.183	0.180	0.178	0.176	0.177	0.183	0.147	0.149	0.148	0.145	0.141	0.137	0.131	0.124	0.118	0.110	0.103	0.097	0.089	0.082	0.078	0.081	0.103
0.180	0.226	0.220	0.220	0.223	0.232	0.217	0.214	0.223	0.209	0.206	0.215	0.202	0.195	0.192	0.198	0.169	0.149	0.158	0.160	0.160	0.158	0.158	0.146	0.139	0.136	0.123	0.115	0.111	0.100	0.091	0.086	0.088	0.108
0.164	0.203	0.203	0.215	0.218	0.000	0.207	0.203	0.000	0.199	0.196	0.000	0.198	0.192	0.179	0.183	0.162	0.148	0.160	0.167	0.173	0.171	0.000	0.157	0.149	0.000	0.133	0.124	0.000	0.109	0.100	0.091	0.091	0.110
0.154	0.194	0.206	0.000	0.204	0.221	0.200	0.197	0.207	0.192	0.191	0.206	0.186	0.000	0.184	0.178	0.157	0.145	0.160	0.173	0.000	0.173	0.166	0.152	0.144	0.141	0.128	0.120	0.117	0.109	0.000	0.095	0.092	0.109
0.148	0.189	0.199	0.196	0.203	0.203	0.186	0.183	0.193	0.179	0.177	0.190	0.186	0.177	0.179	0.175	0.152	0.141	0.158	0.171	0.173	0.164	0.162	0.148	0.141	0.139	0.126	0.118	0.115	0.104	0.100	0.095	0.091	0.107
0.142	0.190	0.000	0.203	0.196	0.000	0.188	0.186	0.000	0.182	0.180	0.000	0.180	0.184	0.000	0.177	0.148	0.137	0.158	0.000	0.166	0.162	0.000	0.149	0.142	0.000	0.127	0.119	0.000	0.104	0.097	0.000	0.091	0.104
0.134	0.171	0.175	0.177	0.172	0.181	0.167	0.166	0.176	0.162	0.160	0.170	0.158	0.160	0.158	0.160	0.140	0.131	0.146	0.157	0.152	0.148	0.149	0.138	0.132	0.130	0.118	0.110	0.107	0.095	0.089	0.087	0.085	0.100
0.126	0.161	0.163	0.165	0.161	0.169	0.158	0.157	0.166	0.154	0.151	0.159	0.149	0.151	0.148	0.151	0.133	0.124	0.139	0.149	0.144	0.141	0.142	0.132	0.126	0.125	0.113	0.105	0.102	0.091	0.084	0.083	0.081	0.095
0.120	0.160	0.000	0.166	0.162	0.000	0.159	0.159	0.000	0.156	0.153	0.000	0.150	0.151	0.000	0.150	0.127	0.118	0.136	0.000	0.141	0.139	0.000	0.130	0.125	0.000	0.112	0.104	0.000	0.090	0.083	0.000	0.079	0.091
0.111	0.142	0.145	0.146	0.142	0.151	0.140	0.139	0.148	0.136	0.134	0.142	0.132	0.134	0.132	0.134	0.118	0.110	0.123	0.133	0.128	0.126	0.127	0.118	0.113	0.112	0.101	0.094	0.091	0.081	0.076	0.075	0.073	0.085
0.104	0.132	0.134	0.136	0.132	0.139	0.129	0.128	0.135	0.126	0.124	0.131	0.123	0.125	0.123	0.125	0.110	0.103	0.115	0.124	0.120	0.118	0.119	0.110	0.105	0.104	0.094	0.088	0.085	0.076	0.071	0.070	0.068	0.080
0.096	0.129	0.000	0.137	0.132	0.000	0.128	0.127	0.000	0.125	0.123	0.000	0.123	0.126	0.000	0.123	0.103	0.097	0.111	0.000	0.117	0.115	0.000	0.107	0.102	0.000	0.091	0.085	0.000	0.074	0.069	0.000	0.066	0.074
0.088	0.113	0.119	0.116	0.120	0.122	0.111	0.110	0.117	0.108	0.107	0.115	0.112	0.106	0.110	0.108	0.095	0.089	0.100	0.109	0.109	0.104	0.104	0.095	0.091	0.090	0.081	0.076	0.074	0.068	0.065	0.061	0.059	0.068
0.081	0.102	0.108	0.000	0.106	0.114	0.104	0.103	0.108	0.101	0.101	0.108	0.099	0.000	0.099	0.097	0.087	0.082	0.091	0.100	0.000	0.100	0.097	0.089	0.084	0.083	0.076	0.071	0.069	0.065	0.000	0.056	0.054	0.063
0.077	0.096	0.096	0.102	0.103	0.000	0.098	0.096	0.000	0.095	0.094	0.000	0.096	0.095	0.088	0.091	0.083	0.078	0.086	0.091	0.095	0.095	0.000	0.087	0.083	0.000	0.075	0.070	0.000	0.061	0.056	0.051	0.050	0.058
0.083	0.105	0.103	0.103	0.105	0.110	0.104	0.102	0.107	0.101	0.100	0.104	0.098	0.095	0.095	0.099	0.087	0.081	0.088	0.091	0.092	0.091	0.091	0.085	0.081	0.079	0.073	0.068	0.066	0.059	0.054	0.050	0.049	0.056
0.120	0.118	0.118	0.117	0.117	0.117	0.116	0.115	0.115	0.113	0.112	0.111	0.110	0.108	0.107	0.109	0.118	0.103	0.108	0.110	0.109	0.107	0.104	0.100	0.095	0.091	0.085	0.080	0.074	0.068	0.063	0.058	0.056	0.060

Table B.10　Rodded A 体系 z=0～42.84cm

2.096	2.096	2.097	2.097	2.091	2.084	2.046	2.012	1.985	1.932	1.884	1.840	1.766	1.684	1.583	1.455	1.268	1.312	1.073	0.956	0.889	0.841	0.800	0.746	0.698	0.656	0.605	0.559	0.518	0.471	0.430	0.408	0.443	0.645
2.096	2.100	2.111	2.127	2.136	2.175	2.087	2.052	2.068	1.970	1.922	1.920	1.804	1.708	1.595	1.458	1.266	1.293	1.355	1.194	1.123	1.083	1.084	0.960	0.893	0.885	0.777	0.716	0.701	0.608	0.543	0.507	0.557	0.637
2.097	2.111	2.152	2.232	2.241	0.000	2.168	2.129	0.000	2.045	1.995	0.000	1.893	1.793	1.628	1.466	1.265	1.284	1.332	1.199	1.204	1.154	0.000	0.991	0.913	0.000	0.800	0.732	0.000	0.649	0.582	0.512	0.548	0.633
2.097	2.127	2.232	0.000	2.251	2.217	2.102	2.062	2.085	1.980	1.936	1.956	1.900	0.000	1.690	1.480	1.266	1.283	1.342	1.284	0.000	1.145	1.169	1.000	0.925	0.924	0.805	0.743	0.750	0.630	0.000	0.552	0.553	0.632
2.091	2.136	2.242	2.251	2.180	2.193	2.087	2.049	2.073	1.968	1.923	1.937	1.841	1.809	1.697	1.489	1.265	1.283	1.367	1.309	1.209	1.179	1.124	0.970	0.899	0.900	0.783	0.722	0.724	0.654	0.578	0.557	0.564	0.633
2.084	2.175	0.000	2.217	2.193	0.000	2.131	2.095	0.000	2.014	1.964	0.000	1.856	1.784	0.000	1.519	1.264	1.283	1.432	0.000	1.309	1.179	0.000	1.029	0.948	0.000	0.833	0.761	0.000	0.665	0.620	0.000	0.593	0.633
2.046	2.087	2.168	2.102	2.087	2.131	2.040	2.012	2.043	1.935	1.883	1.886	1.769	1.694	1.646	1.460	1.243	1.262	1.341	1.244	1.177	1.074	1.078	0.943	0.878	0.882	0.766	0.705	0.700	0.603	0.562	0.527	0.555	0.626
2.012	2.052	2.129	2.062	2.049	2.095	2.012	1.997	2.046	1.922	1.859	1.857	1.739	1.665	1.619	1.438	1.226	1.245	1.321	1.221	1.154	1.055	1.060	0.930	0.868	0.872	0.758	0.697	0.690	0.594	0.553	0.519	0.549	0.621
1.985	2.068	0.000	2.085	2.073	0.000	2.043	2.046	0.000	1.971	1.889	0.000	1.763	1.687	0.000	1.455	1.215	1.234	1.373	0.000	1.222	1.113	0.000	0.990	0.918	0.000	0.809	0.737	0.000	0.634	0.584	0.000	0.575	0.618
1.932	1.970	2.045	1.981	1.968	2.014	1.935	1.922	1.971	1.851	1.791	1.790	1.677	1.607	1.564	1.390	1.187	1.206	1.282	1.188	1.123	1.028	1.035	0.908	0.848	0.854	0.742	0.682	0.677	0.583	0.542	0.509	0.539	0.609
1.885	1.922	1.996	1.936	1.923	1.964	1.883	1.859	1.889	1.791	1.744	1.748	1.642	1.575	1.532	1.362	1.163	1.183	1.260	1.169	1.110	1.016	1.021	0.896	0.836	0.840	0.732	0.674	0.669	0.578	0.539	0.505	0.534	0.602
1.840	1.920	0.000	1.956	1.937	0.000	1.887	1.857	0.000	1.790	1.749	0.000	1.658	1.597	0.000	1.369	1.144	1.165	1.306	0.000	1.202	1.087	0.000	0.954	0.883	0.000	0.779	0.713	0.000	0.625	0.584	0.000	0.559	0.598
1.766	1.804	1.893	1.900	1.841	1.856	1.769	1.739	1.763	1.677	1.642	1.658	1.580	1.557	1.468	1.294	1.106	1.128	1.211	1.168	1.080	1.060	1.017	0.881	0.820	0.824	0.718	0.665	0.668	0.604	0.534	0.517	0.524	0.587
1.684	1.708	1.793	0.000	1.809	1.784	1.695	1.665	1.687	1.607	1.575	1.597	1.557	0.000	1.401	1.236	1.066	1.091	1.154	1.117	0.000	1.008	1.034	0.889	0.827	0.830	0.726	0.673	0.681	0.574	0.000	0.505	0.507	0.578
1.584	1.595	1.628	1.690	1.698	0.000	1.646	1.619	0.000	1.564	1.532	0.000	1.468	1.401	1.283	1.168	1.022	1.056	1.118	1.025	1.043	1.007	0.000	0.875	0.812	0.000	0.718	0.661	0.000	0.590	0.531	0.468	0.500	0.574
1.455	1.458	1.467	1.480	1.489	1.519	1.460	1.439	1.455	1.390	1.362	1.369	1.294	1.236	1.168	1.088	0.971	1.034	1.133	1.031	0.990	0.968	0.977	0.873	0.819	0.815	0.721	0.668	0.657	0.572	0.513	0.480	0.522	0.583
1.268	1.266	1.265	1.266	1.265	1.264	1.243	1.227	1.215	1.187	1.163	1.143	1.106	1.066	1.022	0.971	0.902	1.049	0.950	0.899	0.865	0.835	0.805	0.760	0.717	0.679	0.631	0.587	0.547	0.500	0.459	0.435	0.460	0.628
1.312	1.293	1.284	1.283	1.283	1.283	1.262	1.245	1.234	1.206	1.183	1.165	1.128	1.091	1.056	1.034	1.049	0.830	0.830	0.815	0.795	0.772	0.747	0.706	0.668	0.634	0.589	0.548	0.512	0.468	0.431	0.409	0.427	0.548
1.073	1.355	1.332	1.342	1.367	1.432	1.341	1.321	1.373	1.282	1.260	1.306	1.211	1.154	1.118	1.133	0.950	0.830	0.872	0.879	0.876	0.863	0.862	0.792	0.749	0.733	0.662	0.617	0.594	0.529	0.483	0.453	0.465	0.577
0.956	1.194	1.199	1.284	1.309	0.000	1.244	1.221	0.000	1.188	1.169	0.000	1.169	1.117	1.025	1.031	0.899	0.815	0.879	0.916	0.955	0.943	0.000	0.857	0.809	0.000	0.718	0.668	0.000	0.584	0.533	0.481	0.480	0.584
0.889	1.123	1.204	0.000	1.209	1.309	1.177	1.154	1.222	1.123	1.110	1.202	1.080	0.000	1.043	0.990	0.865	0.795	0.876	0.955	0.000	0.950	0.910	0.824	0.778	0.765	0.690	0.644	0.631	0.588	0.000	0.508	0.486	0.580
0.841	1.083	1.154	1.145	1.179	1.179	1.074	1.055	1.113	1.028	1.016	1.087	1.060	1.008	1.007	0.967	0.835	0.772	0.863	0.943	0.950	0.897	0.883	0.805	0.761	0.749	0.676	0.632	0.616	0.559	0.538	0.506	0.484	0.569
0.800	1.084	0.000	1.169	1.124	0.000	1.078	1.060	0.000	1.036	1.021	0.000	1.018	1.034	0.000	0.977	0.805	0.747	0.862	0.000	0.910	0.883	0.000	0.810	0.767	0.000	0.683	0.637	0.000	0.555	0.517	0.000	0.487	0.555
0.746	0.959	0.990	1.000	0.970	1.029	0.943	0.930	0.990	0.908	0.896	0.954	0.881	0.889	0.875	0.873	0.760	0.706	0.792	0.858	0.824	0.805	0.810	0.744	0.707	0.698	0.630	0.587	0.570	0.508	0.472	0.466	0.452	0.530
0.698	0.893	0.913	0.925	0.899	0.948	0.878	0.868	0.918	0.848	0.836	0.883	0.820	0.827	0.812	0.819	0.717	0.668	0.749	0.809	0.778	0.761	0.767	0.707	0.673	0.664	0.600	0.559	0.542	0.483	0.448	0.443	0.430	0.506
0.656	0.885	0.000	0.924	0.900	0.000	0.882	0.872	0.000	0.854	0.840	0.000	0.824	0.830	0.000	0.815	0.679	0.634	0.732	0.000	0.765	0.749	0.000	0.697	0.664	0.000	0.593	0.552	0.000	0.478	0.442	0.000	0.423	0.482
0.605	0.777	0.800	0.805	0.783	0.833	0.766	0.758	0.809	0.742	0.731	0.779	0.718	0.726	0.719	0.721	0.631	0.589	0.662	0.718	0.690	0.676	0.683	0.630	0.600	0.593	0.536	0.500	0.485	0.432	0.401	0.397	0.385	0.452
0.559	0.716	0.732	0.743	0.722	0.761	0.705	0.697	0.737	0.682	0.674	0.713	0.664	0.673	0.661	0.668	0.587	0.548	0.617	0.668	0.644	0.631	0.637	0.587	0.559	0.552	0.500	0.467	0.453	0.405	0.376	0.371	0.360	0.423
0.518	0.701	0.000	0.750	0.724	0.000	0.700	0.690	0.000	0.677	0.669	0.000	0.668	0.681	0.000	0.657	0.547	0.512	0.594	0.000	0.631	0.616	0.000	0.570	0.542	0.000	0.485	0.453	0.000	0.396	0.368	0.000	0.348	0.395
0.471	0.608	0.649	0.630	0.654	0.665	0.603	0.594	0.634	0.583	0.578	0.625	0.604	0.574	0.590	0.572	0.500	0.468	0.529	0.584	0.588	0.559	0.555	0.508	0.483	0.478	0.432	0.405	0.396	0.359	0.345	0.326	0.311	0.363
0.430	0.543	0.582	0.000	0.578	0.620	0.562	0.553	0.584	0.542	0.539	0.584	0.534	0.000	0.531	0.513	0.459	0.431	0.483	0.533	0.000	0.538	0.517	0.472	0.448	0.442	0.401	0.376	0.368	0.345	0.000	0.298	0.283	0.332
0.408	0.507	0.512	0.552	0.557	0.000	0.527	0.519	0.000	0.509	0.505	0.000	0.517	0.505	0.468	0.480	0.435	0.409	0.453	0.481	0.508	0.506	0.000	0.466	0.443	0.000	0.397	0.371	0.000	0.326	0.299	0.267	0.261	0.306
0.443	0.557	0.548	0.553	0.564	0.593	0.555	0.549	0.575	0.539	0.534	0.559	0.524	0.507	0.500	0.522	0.460	0.427	0.465	0.480	0.486	0.484	0.487	0.452	0.430	0.423	0.385	0.360	0.348	0.311	0.283	0.261	0.256	0.294
0.645	0.637	0.633	0.632	0.633	0.633	0.626	0.621	0.618	0.609	0.602	0.598	0.587	0.578	0.574	0.583	0.628	0.548	0.577	0.584	0.580	0.569	0.555	0.530	0.506	0.482	0.452	0.423	0.395	0.363	0.332	0.306	0.294	0.316

Table B.11　Rodded B 体系　z=0～14.28cm

1.066	1.068	1.071	1.076	1.078	1.080	1.061	1.047	1.037	1.011	0.990	0.971	0.933	0.891	0.839	0.772	0.675	0.701	0.577	0.517	0.484	0.461	0.442	0.415	0.390	0.369	0.342	0.318	0.296	0.270	0.247	0.236	0.257	0.377
1.068	1.072	1.083	1.099	1.112	1.145	1.092	1.076	1.097	1.040	1.018	1.030	0.963	0.910	0.848	0.775	0.674	0.691	0.729	0.647	0.613	0.595	0.600	0.534	0.500	0.499	0.440	0.408	0.402	0.349	0.313	0.294	0.324	0.372
1.071	1.083	1.115	1.175	1.188	0.000	1.151	1.132	0.000	1.095	1.073	0.000	1.029	0.974	0.875	0.784	0.676	0.688	0.718	0.652	0.659	0.636	0.000	0.553	0.513	0.000	0.454	0.418	0.000	0.374	0.337	0.297	0.319	0.370
1.076	1.099	1.174	0.000	1.199	1.181	1.110	1.091	1.116	1.054	1.035	1.063	1.038	0.000	0.924	0.797	0.680	0.690	0.725	0.700	0.000	0.632	0.650	0.558	0.520	0.523	0.458	0.425	0.431	0.364	0.000	0.321	0.323	0.371
1.078	1.112	1.188	1.199	1.156	1.170	1.104	1.086	1.111	1.050	1.030	1.053	1.001	0.995	0.934	0.808	0.682	0.692	0.741	0.715	0.665	0.653	0.626	0.543	0.507	0.510	0.446	0.414	0.417	0.378	0.336	0.325	0.330	0.371
1.080	1.145	0.000	1.181	1.170	0.000	1.138	1.121	0.000	1.085	1.062	0.000	1.016	0.981	0.000	0.835	0.685	0.694	0.779	0.000	0.721	0.654	0.000	0.577	0.536	0.000	0.475	0.437	0.000	0.385	0.361	0.000	0.347	0.372
1.061	1.092	1.151	1.110	1.104	1.138	1.081	1.067	1.094	1.032	1.010	1.027	0.960	0.923	0.907	0.797	0.675	0.684	0.730	0.682	0.649	0.597	0.603	0.530	0.496	0.502	0.438	0.405	0.404	0.350	0.328	0.308	0.326	0.369
1.047	1.076	1.132	1.091	1.086	1.121	1.067	1.054	1.083	1.021	0.998	1.013	0.945	0.909	0.895	0.787	0.667	0.677	0.722	0.672	0.639	0.588	0.595	0.524	0.492	0.497	0.434	0.401	0.400	0.346	0.323	0.305	0.323	0.367
1.037	1.097	0.000	1.116	1.111	0.000	1.094	1.083	0.000	1.049	1.024	0.000	0.970	0.932	0.000	0.806	0.664	0.673	0.753	0.000	0.678	0.622	0.000	0.560	0.522	0.000	0.464	0.426	0.000	0.370	0.342	0.000	0.339	0.366
1.011	1.040	1.095	1.054	1.050	1.084	1.032	1.021	1.049	0.989	0.967	0.983	0.918	0.883	0.870	0.766	0.650	0.660	0.705	0.658	0.625	0.576	0.584	0.515	0.483	0.489	0.427	0.395	0.394	0.341	0.319	0.301	0.319	0.362
0.990	1.018	1.073	1.035	1.030	1.062	1.010	0.998	1.024	0.967	0.948	0.964	0.902	0.870	0.856	0.753	0.639	0.650	0.695	0.650	0.620	0.572	0.578	0.510	0.478	0.484	0.423	0.392	0.392	0.339	0.318	0.299	0.317	0.359
0.971	1.030	0.000	1.063	1.053	0.000	1.027	1.013	0.000	0.983	0.964	0.000	0.925	0.896	0.000	0.767	0.632	0.643	0.724	0.000	0.675	0.615	0.000	0.545	0.508	0.000	0.452	0.417	0.000	0.368	0.346	0.000	0.333	0.357
0.933	0.963	1.029	1.038	1.002	1.016	0.960	0.945	0.970	0.918	0.902	0.925	0.882	0.879	0.830	0.721	0.611	0.624	0.673	0.655	0.609	0.601	0.581	0.505	0.473	0.478	0.419	0.389	0.394	0.357	0.317	0.308	0.313	0.352
0.891	0.910	0.974	0.000	0.995	0.981	0.923	0.909	0.932	0.883	0.870	0.896	0.879	0.000	0.792	0.687	0.591	0.605	0.644	0.628	0.000	0.574	0.593	0.512	0.479	0.484	0.425	0.396	0.403	0.341	0.000	0.302	0.303	0.347
0.838	0.848	0.875	0.924	0.934	0.000	0.907	0.895	0.000	0.870	0.856	0.000	0.830	0.791	0.716	0.648	0.567	0.587	0.625	0.578	0.592	0.576	0.000	0.506	0.473	0.000	0.422	0.391	0.000	0.352	0.317	0.280	0.300	0.346
0.772	0.775	0.784	0.797	0.808	0.835	0.797	0.787	0.806	0.766	0.753	0.767	0.721	0.687	0.648	0.604	0.540	0.577	0.637	0.584	0.564	0.555	0.564	0.507	0.478	0.479	0.425	0.396	0.391	0.342	0.307	0.288	0.315	0.353
0.675	0.674	0.676	0.680	0.682	0.685	0.675	0.667	0.664	0.650	0.639	0.632	0.612	0.591	0.567	0.540	0.504	0.589	0.537	0.511	0.495	0.481	0.466	0.443	0.420	0.400	0.373	0.349	0.326	0.299	0.275	0.262	0.278	0.381
0.701	0.691	0.688	0.690	0.692	0.694	0.684	0.677	0.673	0.660	0.650	0.643	0.624	0.605	0.587	0.577	0.589	0.469	0.472	0.466	0.457	0.447	0.435	0.413	0.393	0.375	0.350	0.327	0.307	0.281	0.260	0.248	0.259	0.334
0.577	0.729	0.718	0.725	0.741	0.779	0.730	0.722	0.753	0.705	0.695	0.724	0.673	0.644	0.625	0.637	0.537	0.472	0.498	0.506	0.507	0.502	0.505	0.466	0.443	0.435	0.395	0.369	0.358	0.320	0.293	0.275	0.283	0.352
0.517	0.647	0.652	0.700	0.715	0.000	0.682	0.672	0.000	0.658	0.650	0.000	0.655	0.628	0.578	0.584	0.511	0.466	0.505	0.530	0.556	0.553	0.000	0.507	0.481	0.000	0.430	0.402	0.000	0.354	0.325	0.293	0.294	0.358
0.484	0.613	0.659	0.000	0.665	0.721	0.649	0.639	0.678	0.625	0.621	0.675	0.609	0.000	0.592	0.564	0.495	0.458	0.507	0.556	0.000	0.559	0.538	0.489	0.464	0.459	0.415	0.389	0.383	0.358	0.000	0.311	0.298	0.357
0.461	0.595	0.636	0.632	0.653	0.654	0.597	0.588	0.622	0.576	0.572	0.615	0.601	0.574	0.576	0.555	0.481	0.447	0.502	0.553	0.559	0.530	0.525	0.480	0.456	0.451	0.408	0.383	0.375	0.342	0.330	0.311	0.298	0.352
0.442	0.600	0.000	0.650	0.626	0.000	0.603	0.595	0.000	0.584	0.578	0.000	0.581	0.593	0.000	0.564	0.466	0.435	0.505	0.000	0.538	0.525	0.000	0.486	0.462	0.000	0.415	0.388	0.000	0.341	0.318	0.000	0.301	0.344
0.414	0.534	0.553	0.559	0.543	0.577	0.530	0.524	0.560	0.515	0.510	0.545	0.505	0.512	0.506	0.507	0.443	0.413	0.466	0.507	0.489	0.480	0.486	0.448	0.427	0.423	0.383	0.359	0.350	0.313	0.292	0.289	0.280	0.330
0.390	0.500	0.513	0.520	0.506	0.536	0.496	0.492	0.522	0.483	0.478	0.508	0.473	0.479	0.473	0.478	0.420	0.393	0.443	0.481	0.464	0.456	0.462	0.427	0.408	0.404	0.366	0.343	0.334	0.299	0.278	0.275	0.268	0.316
0.369	0.499	0.000	0.523	0.510	0.000	0.502	0.497	0.000	0.489	0.484	0.000	0.478	0.484	0.000	0.479	0.400	0.375	0.435	0.000	0.459	0.451	0.000	0.423	0.404	0.000	0.364	0.340	0.000	0.296	0.275	0.000	0.264	0.302
0.342	0.440	0.454	0.458	0.446	0.475	0.438	0.434	0.464	0.427	0.423	0.452	0.419	0.425	0.422	0.425	0.373	0.350	0.395	0.430	0.415	0.408	0.415	0.383	0.366	0.364	0.330	0.309	0.301	0.269	0.250	0.248	0.241	0.284
0.318	0.408	0.418	0.425	0.414	0.437	0.405	0.401	0.426	0.395	0.392	0.417	0.389	0.396	0.391	0.396	0.349	0.327	0.369	0.402	0.389	0.383	0.388	0.359	0.343	0.340	0.309	0.290	0.282	0.253	0.236	0.233	0.226	0.267
0.296	0.402	0.000	0.431	0.417	0.000	0.404	0.400	0.000	0.394	0.392	0.000	0.394	0.403	0.000	0.391	0.326	0.306	0.358	0.000	0.383	0.375	0.000	0.350	0.334	0.000	0.301	0.282	0.000	0.248	0.232	0.000	0.219	0.250
0.270	0.349	0.374	0.364	0.378	0.385	0.350	0.346	0.370	0.341	0.339	0.368	0.357	0.341	0.352	0.342	0.299	0.281	0.320	0.354	0.358	0.342	0.341	0.313	0.299	0.296	0.269	0.253	0.248	0.226	0.217	0.206	0.196	0.230
0.247	0.313	0.337	0.000	0.336	0.361	0.328	0.323	0.342	0.319	0.318	0.346	0.317	0.000	0.317	0.307	0.275	0.260	0.292	0.325	0.000	0.330	0.318	0.292	0.278	0.275	0.250	0.236	0.232	0.217	0.000	0.189	0.179	0.211
0.236	0.294	0.297	0.321	0.325	0.000	0.308	0.305	0.000	0.301	0.299	0.000	0.308	0.302	0.280	0.288	0.262	0.248	0.275	0.293	0.311	0.311	0.000	0.289	0.275	0.000	0.248	0.233	0.000	0.206	0.189	0.169	0.166	0.195
0.257	0.324	0.319	0.323	0.330	0.347	0.326	0.323	0.339	0.319	0.317	0.333	0.313	0.303	0.300	0.315	0.278	0.259	0.283	0.293	0.298	0.298	0.301	0.280	0.268	0.264	0.241	0.227	0.219	0.196	0.179	0.166	0.163	0.187
0.377	0.372	0.370	0.371	0.371	0.372	0.369	0.367	0.366	0.362	0.359	0.357	0.352	0.347	0.346	0.353	0.381	0.334	0.352	0.358	0.357	0.352	0.344	0.330	0.316	0.302	0.284	0.267	0.250	0.230	0.211	0.195	0.187	0.202

Table B.12　Rodded B 体系　z=14.28～28.56cm

0.553	0.548	0.537	0.522	0.507	0.493	0.489	0.482	0.472	0.469	0.463	0.454	0.453	0.450	0.443	0.424	0.383	0.409	0.346	0.318	0.302	0.292	0.283	0.268	0.255	0.242	0.226	0.212	0.198	0.182	0.167	0.160	0.176	0.259
0.548	0.540	0.525	0.504	0.483	0.454	0.466	0.460	0.436	0.447	0.442	0.418	0.430	0.434	0.433	0.418	0.379	0.401	0.437	0.397	0.382	0.376	0.383	0.345	0.326	0.327	0.291	0.271	0.268	0.235	0.211	0.200	0.222	0.256
0.537	0.525	0.497	0.455	0.434	0.000	0.428	0.424	0.000	0.412	0.407	0.000	0.387	0.392	0.409	0.406	0.372	0.395	0.428	0.399	0.410	0.401	0.000	0.355	0.333	0.000	0.299	0.278	0.000	0.251	0.227	0.202	0.218	0.255
0.522	0.504	0.455	0.000	0.414	0.413	0.437	0.435	0.409	0.424	0.416	0.382	0.373	0.000	0.375	0.391	0.365	0.391	0.429	0.427	0.000	0.397	0.413	0.360	0.339	0.343	0.302	0.283	0.288	0.244	0.000	0.217	0.221	0.255
0.508	0.483	0.434	0.414	0.425	0.412	0.433	0.431	0.406	0.420	0.412	0.381	0.382	0.360	0.360	0.376	0.357	0.388	0.437	0.434	0.411	0.411	0.399	0.350	0.330	0.335	0.295	0.275	0.279	0.254	0.226	0.220	0.225	0.255
0.493	0.454	0.000	0.414	0.412	0.000	0.412	0.412	0.000	0.401	0.393	0.000	0.370	0.362	0.000	0.355	0.350	0.385	0.457	0.000	0.445	0.412	0.000	0.372	0.349	0.000	0.314	0.291	0.000	0.259	0.244	0.000	0.237	0.256
0.489	0.466	0.428	0.437	0.433	0.412	0.435	0.441	0.425	0.431	0.415	0.383	0.391	0.383	0.359	0.367	0.348	0.380	0.429	0.413	0.403	0.377	0.385	0.343	0.324	0.330	0.290	0.270	0.271	0.236	0.222	0.209	0.223	0.254
0.482	0.460	0.424	0.435	0.431	0.412	0.441	0.465	0.483	0.455	0.422	0.383	0.390	0.382	0.357	0.364	0.345	0.377	0.424	0.408	0.397	0.372	0.381	0.340	0.323	0.329	0.289	0.269	0.269	0.233	0.219	0.207	0.222	0.253
0.472	0.436	0.000	0.409	0.406	0.000	0.425	0.483	0.000	0.473	0.408	0.000	0.367	0.361	0.000	0.345	0.340	0.374	0.444	0.000	0.421	0.395	0.000	0.363	0.343	0.000	0.310	0.285	0.000	0.250	0.232	0.000	0.232	0.253
0.469	0.447	0.412	0.424	0.420	0.401	0.431	0.455	0.473	0.446	0.413	0.375	0.381	0.374	0.349	0.357	0.339	0.370	0.417	0.402	0.391	0.367	0.376	0.336	0.319	0.325	0.286	0.266	0.266	0.231	0.217	0.205	0.219	0.250
0.463	0.442	0.407	0.416	0.412	0.393	0.415	0.422	0.408	0.413	0.398	0.367	0.375	0.368	0.345	0.353	0.335	0.367	0.414	0.399	0.390	0.366	0.374	0.333	0.316	0.322	0.283	0.264	0.265	0.231	0.217	0.205	0.218	0.249
0.454	0.418	0.000	0.383	0.381	0.000	0.383	0.383	0.000	0.375	0.367	0.000	0.347	0.340	0.000	0.334	0.330	0.365	0.434	0.000	0.425	0.394	0.000	0.357	0.336	0.000	0.303	0.281	0.000	0.251	0.236	0.000	0.230	0.248
0.453	0.430	0.387	0.372	0.382	0.370	0.391	0.390	0.367	0.381	0.375	0.347	0.349	0.331	0.329	0.346	0.330	0.360	0.408	0.408	0.387	0.388	0.379	0.333	0.315	0.320	0.282	0.264	0.268	0.244	0.218	0.212	0.217	0.246
0.450	0.434	0.392	0.000	0.360	0.362	0.383	0.382	0.361	0.374	0.368	0.340	0.331	0.000	0.335	0.351	0.330	0.357	0.396	0.397	0.000	0.372	0.389	0.340	0.321	0.326	0.288	0.270	0.275	0.234	0.000	0.209	0.211	0.244
0.443	0.433	0.409	0.375	0.359	0.000	0.359	0.357	0.000	0.349	0.345	0.000	0.329	0.335	0.353	0.354	0.329	0.355	0.391	0.370	0.384	0.378	0.000	0.338	0.318	0.000	0.287	0.267	0.000	0.242	0.220	0.195	0.210	0.244
0.424	0.418	0.406	0.391	0.376	0.355	0.367	0.364	0.345	0.357	0.353	0.334	0.346	0.351	0.354	0.348	0.324	0.358	0.405	0.379	0.371	0.368	0.377	0.342	0.325	0.326	0.292	0.273	0.270	0.237	0.215	0.202	0.222	0.250
0.383	0.379	0.372	0.365	0.357	0.350	0.348	0.345	0.340	0.339	0.335	0.330	0.330	0.330	0.329	0.324	0.311	0.373	0.348	0.336	0.330	0.323	0.316	0.302	0.288	0.276	0.259	0.243	0.228	0.210	0.194	0.185	0.197	0.271
0.409	0.401	0.395	0.391	0.388	0.385	0.380	0.377	0.374	0.370	0.367	0.365	0.360	0.357	0.355	0.358	0.373	0.303	0.310	0.310	0.307	0.303	0.297	0.284	0.271	0.260	0.244	0.229	0.215	0.198	0.184	0.176	0.185	0.238
0.346	0.437	0.428	0.429	0.437	0.457	0.429	0.424	0.444	0.417	0.414	0.434	0.408	0.396	0.391	0.405	0.348	0.310	0.331	0.340	0.343	0.343	0.347	0.322	0.308	0.304	0.277	0.260	0.253	0.227	0.208	0.196	0.202	0.252
0.317	0.397	0.398	0.427	0.434	0.000	0.413	0.408	0.000	0.402	0.400	0.000	0.408	0.397	0.370	0.379	0.336	0.310	0.340	0.359	0.380	0.380	0.000	0.352	0.336	0.000	0.303	0.284	0.000	0.252	0.232	0.210	0.210	0.257
0.302	0.382	0.410	0.000	0.411	0.445	0.403	0.397	0.421	0.391	0.390	0.425	0.387	0.000	0.384	0.371	0.330	0.307	0.343	0.380	0.000	0.386	0.374	0.342	0.326	0.323	0.294	0.276	0.273	0.256	0.000	0.224	0.215	0.257
0.292	0.376	0.401	0.397	0.410	0.412	0.377	0.372	0.395	0.367	0.366	0.395	0.388	0.372	0.378	0.368	0.323	0.303	0.343	0.380	0.386	0.368	0.366	0.337	0.322	0.319	0.290	0.273	0.268	0.245	0.237	0.224	0.215	0.254
0.283	0.383	0.000	0.413	0.399	0.000	0.385	0.381	0.000	0.376	0.374	0.000	0.379	0.389	0.000	0.377	0.316	0.297	0.347	0.000	0.374	0.366	0.000	0.342	0.327	0.000	0.296	0.278	0.000	0.245	0.230	0.000	0.218	0.250
0.268	0.345	0.355	0.360	0.350	0.372	0.343	0.340	0.363	0.336	0.333	0.357	0.333	0.340	0.338	0.342	0.302	0.284	0.322	0.352	0.342	0.337	0.342	0.317	0.303	0.302	0.274	0.258	0.252	0.226	0.211	0.209	0.204	0.240
0.254	0.326	0.333	0.338	0.330	0.349	0.324	0.323	0.343	0.319	0.316	0.336	0.315	0.321	0.318	0.325	0.288	0.271	0.308	0.336	0.326	0.322	0.327	0.304	0.291	0.289	0.263	0.247	0.241	0.216	0.202	0.200	0.195	0.230
0.242	0.327	0.000	0.343	0.335	0.000	0.330	0.329	0.000	0.325	0.322	0.000	0.320	0.326	0.000	0.326	0.276	0.260	0.304	0.000	0.323	0.319	0.000	0.302	0.289	0.000	0.262	0.246	0.000	0.215	0.200	0.000	0.193	0.221
0.226	0.291	0.299	0.302	0.295	0.314	0.290	0.289	0.309	0.286	0.283	0.303	0.282	0.288	0.287	0.292	0.259	0.244	0.277	0.303	0.294	0.290	0.296	0.274	0.263	0.262	0.238	0.224	0.219	0.196	0.182	0.181	0.177	0.208
0.212	0.271	0.278	0.283	0.275	0.291	0.270	0.269	0.285	0.266	0.264	0.281	0.264	0.270	0.267	0.273	0.243	0.229	0.260	0.284	0.276	0.273	0.278	0.258	0.247	0.246	0.224	0.210	0.205	0.184	0.172	0.171	0.166	0.196
0.198	0.268	0.000	0.288	0.279	0.000	0.271	0.268	0.000	0.266	0.265	0.000	0.268	0.275	0.000	0.270	0.228	0.215	0.253	0.000	0.273	0.268	0.000	0.252	0.241	0.000	0.219	0.206	0.000	0.181	0.170	0.000	0.161	0.184
0.182	0.235	0.251	0.244	0.254	0.259	0.236	0.233	0.250	0.231	0.231	0.251	0.244	0.234	0.242	0.237	0.210	0.199	0.226	0.252	0.256	0.245	0.245	0.226	0.216	0.215	0.196	0.184	0.181	0.165	0.159	0.151	0.145	0.169
0.167	0.211	0.227	0.000	0.226	0.244	0.222	0.219	0.232	0.217	0.217	0.236	0.218	0.000	0.219	0.215	0.194	0.184	0.208	0.232	0.000	0.237	0.230	0.211	0.202	0.200	0.182	0.172	0.170	0.159	0.000	0.139	0.132	0.155
0.160	0.200	0.201	0.217	0.220	0.000	0.209	0.207	0.000	0.205	0.205	0.000	0.212	0.209	0.195	0.202	0.185	0.176	0.196	0.210	0.224	0.224	0.000	0.209	0.200	0.000	0.181	0.171	0.000	0.151	0.139	0.125	0.122	0.144
0.176	0.222	0.218	0.221	0.225	0.237	0.223	0.222	0.232	0.219	0.218	0.230	0.217	0.211	0.210	0.222	0.197	0.185	0.202	0.211	0.215	0.215	0.218	0.204	0.195	0.193	0.177	0.166	0.161	0.145	0.132	0.122	0.120	0.138
0.259	0.256	0.255	0.255	0.255	0.256	0.254	0.253	0.253	0.250	0.249	0.248	0.246	0.244	0.244	0.250	0.271	0.238	0.252	0.257	0.257	0.254	0.250	0.240	0.231	0.221	0.208	0.196	0.184	0.169	0.155	0.144	0.138	0.150

Table B.13　Rodded B 体系　z=28.56～42.84cm

0.216	0.213	0.208	0.201	0.194	0.188	0.186	0.184	0.180	0.180	0.179	0.176	0.178	0.179	0.178	0.172	0.157	0.168	0.142	0.129	0.120	0.114	0.108	0.103	0.098	0.093	0.088	0.082	0.077	0.072	0.068	0.067	0.074	0.110
0.213	0.210	0.202	0.193	0.183	0.171	0.176	0.174	0.165	0.170	0.169	0.160	0.167	0.171	0.173	0.169	0.154	0.164	0.177	0.157	0.146	0.136	0.127	0.123	0.117	0.109	0.105	0.099	0.090	0.086	0.082	0.081	0.093	0.108
0.208	0.202	0.189	0.171	0.161	0.000	0.159	0.158	0.000	0.155	0.153	0.000	0.148	0.152	0.162	0.163	0.151	0.160	0.169	0.147	0.132	0.123	0.000	0.111	0.107	0.000	0.095	0.089	0.000	0.077	0.074	0.075	0.089	0.106
0.201	0.193	0.171	0.000	0.152	0.152	0.163	0.162	0.152	0.159	0.157	0.143	0.140	0.000	0.146	0.155	0.146	0.156	0.164	0.138	0.000	0.116	0.124	0.122	0.116	0.108	0.104	0.098	0.089	0.073	0.000	0.071	0.086	0.105
0.194	0.183	0.161	0.152	0.156	0.151	0.161	0.161	0.151	0.158	0.155	0.143	0.144	0.137	0.138	0.148	0.142	0.152	0.159	0.133	0.120	0.130	0.122	0.119	0.114	0.106	0.102	0.096	0.087	0.082	0.068	0.068	0.084	0.104
0.188	0.171	0.000	0.152	0.151	0.000	0.153	0.153	0.000	0.150	0.147	0.000	0.140	0.137	0.000	0.138	0.138	0.149	0.153	0.000	0.133	0.126	0.000	0.116	0.112	0.000	0.100	0.094	0.000	0.080	0.076	0.000	0.082	0.103
0.186	0.176	0.159	0.163	0.161	0.152	0.163	0.166	0.160	0.163	0.157	0.144	0.149	0.147	0.138	0.144	0.137	0.147	0.155	0.130	0.136	0.128	0.120	0.119	0.115	0.109	0.104	0.096	0.087	0.082	0.077	0.068	0.083	0.103
0.184	0.174	0.158	0.162	0.161	0.153	0.166	0.176	0.184	0.174	0.161	0.145	0.149	0.147	0.138	0.143	0.136	0.146	0.153	0.129	0.135	0.128	0.120	0.120	0.122	0.126	0.111	0.098	0.087	0.082	0.077	0.068	0.083	0.103
0.180	0.165	0.000	0.152	0.151	0.000	0.160	0.184	0.000	0.181	0.155	0.000	0.140	0.139	0.000	0.135	0.134	0.145	0.150	0.000	0.131	0.125	0.000	0.119	0.131	0.000	0.120	0.098	0.000	0.080	0.075	0.000	0.082	0.102
0.180	0.170	0.155	0.159	0.158	0.150	0.163	0.174	0.181	0.171	0.158	0.143	0.147	0.145	0.136	0.141	0.135	0.145	0.152	0.128	0.134	0.127	0.120	0.119	0.121	0.125	0.110	0.097	0.087	0.081	0.077	0.068	0.083	0.102
0.179	0.169	0.153	0.157	0.155	0.147	0.157	0.161	0.155	0.158	0.153	0.141	0.145	0.143	0.135	0.140	0.134	0.144	0.152	0.128	0.134	0.126	0.119	0.117	0.114	0.109	0.103	0.096	0.086	0.081	0.077	0.068	0.083	0.102
0.176	0.160	0.000	0.143	0.143	0.000	0.144	0.145	0.000	0.143	0.141	0.000	0.134	0.132	0.000	0.133	0.133	0.144	0.149	0.000	0.131	0.124	0.000	0.114	0.110	0.000	0.099	0.093	0.000	0.080	0.075	0.000	0.082	0.103
0.178	0.167	0.148	0.140	0.144	0.140	0.149	0.149	0.140	0.147	0.145	0.134	0.135	0.129	0.131	0.140	0.135	0.145	0.153	0.129	0.117	0.127	0.120	0.117	0.113	0.105	0.101	0.095	0.087	0.082	0.068	0.068	0.084	0.103
0.179	0.171	0.152	0.000	0.137	0.137	0.147	0.147	0.139	0.145	0.143	0.132	0.129	0.000	0.135	0.145	0.137	0.148	0.157	0.134	0.000	0.114	0.123	0.121	0.116	0.108	0.104	0.098	0.089	0.074	0.000	0.071	0.087	0.105
0.178	0.173	0.162	0.146	0.138	0.000	0.138	0.138	0.000	0.136	0.135	0.000	0.131	0.135	0.145	0.149	0.139	0.150	0.163	0.143	0.131	0.123	0.000	0.113	0.109	0.000	0.098	0.092	0.000	0.080	0.076	0.077	0.091	0.107
0.172	0.169	0.163	0.155	0.148	0.138	0.144	0.143	0.135	0.141	0.140	0.133	0.140	0.145	0.149	0.148	0.140	0.155	0.175	0.160	0.152	0.145	0.137	0.134	0.129	0.120	0.116	0.109	0.100	0.095	0.090	0.089	0.100	0.112
0.157	0.154	0.151	0.146	0.142	0.138	0.137	0.136	0.134	0.135	0.134	0.133	0.135	0.137	0.139	0.140	0.136	0.164	0.154	0.148	0.144	0.140	0.135	0.131	0.125	0.119	0.113	0.107	0.100	0.093	0.088	0.085	0.091	0.124
0.168	0.164	0.160	0.156	0.152	0.149	0.147	0.146	0.145	0.145	0.144	0.144	0.146	0.148	0.150	0.155	0.164	0.135	0.139	0.139	0.138	0.136	0.133	0.127	0.122	0.117	0.111	0.104	0.099	0.091	0.085	0.082	0.087	0.111
0.142	0.177	0.169	0.164	0.159	0.153	0.154	0.153	0.150	0.152	0.152	0.149	0.153	0.157	0.163	0.175	0.154	0.139	0.149	0.153	0.155	0.155	0.157	0.146	0.140	0.139	0.127	0.120	0.117	0.105	0.098	0.093	0.096	0.118
0.129	0.157	0.147	0.138	0.133	0.000	0.130	0.129	0.000	0.128	0.128	0.000	0.129	0.134	0.143	0.160	0.148	0.139	0.153	0.162	0.171	0.172	0.000	0.161	0.154	0.000	0.140	0.132	0.000	0.118	0.109	0.099	0.100	0.122
0.120	0.146	0.132	0.000	0.120	0.133	0.136	0.135	0.131	0.134	0.134	0.131	0.117	0.000	0.131	0.152	0.144	0.138	0.155	0.171	0.000	0.176	0.171	0.158	0.151	0.150	0.137	0.130	0.128	0.120	0.000	0.106	0.103	0.123
0.114	0.136	0.123	0.116	0.130	0.126	0.128	0.128	0.125	0.127	0.126	0.124	0.127	0.114	0.123	0.145	0.140	0.136	0.155	0.172	0.176	0.169	0.169	0.157	0.151	0.150	0.137	0.129	0.127	0.116	0.113	0.107	0.103	0.122
0.108	0.127	0.000	0.124	0.122	0.000	0.120	0.120	0.000	0.120	0.119	0.000	0.120	0.123	0.000	0.137	0.135	0.133	0.157	0.000	0.171	0.169	0.000	0.160	0.154	0.000	0.140	0.132	0.000	0.117	0.110	0.000	0.105	0.120
0.103	0.123	0.111	0.122	0.119	0.116	0.119	0.120	0.119	0.119	0.117	0.114	0.117	0.121	0.113	0.134	0.131	0.127	0.146	0.161	0.158	0.157	0.160	0.150	0.144	0.144	0.132	0.124	0.121	0.109	0.102	0.101	0.099	0.116
0.098	0.117	0.107	0.116	0.114	0.111	0.115	0.122	0.131	0.121	0.114	0.110	0.113	0.116	0.109	0.129	0.125	0.122	0.140	0.154	0.151	0.151	0.154	0.144	0.140	0.139	0.128	0.120	0.117	0.105	0.098	0.097	0.095	0.112
0.093	0.109	0.000	0.108	0.106	0.000	0.109	0.126	0.000	0.125	0.109	0.000	0.105	0.108	0.000	0.120	0.119	0.117	0.139	0.000	0.150	0.150	0.000	0.144	0.139	0.000	0.128	0.119	0.000	0.105	0.098	0.000	0.094	0.108
0.088	0.105	0.095	0.104	0.102	0.100	0.104	0.111	0.120	0.110	0.103	0.099	0.101	0.104	0.098	0.116	0.113	0.111	0.127	0.140	0.137	0.137	0.140	0.132	0.127	0.128	0.117	0.110	0.107	0.096	0.090	0.089	0.087	0.102
0.082	0.099	0.089	0.098	0.096	0.094	0.096	0.098	0.097	0.097	0.095	0.093	0.095	0.098	0.092	0.109	0.107	0.104	0.120	0.132	0.130	0.129	0.132	0.124	0.120	0.119	0.110	0.103	0.101	0.091	0.085	0.084	0.082	0.096
0.077	0.090	0.000	0.089	0.087	0.000	0.087	0.087	0.000	0.087	0.086	0.000	0.087	0.089	0.000	0.100	0.100	0.099	0.117	0.000	0.128	0.127	0.000	0.121	0.117	0.000	0.107	0.101	0.000	0.089	0.083	0.000	0.080	0.091
0.072	0.086	0.077	0.073	0.082	0.080	0.082	0.082	0.080	0.081	0.081	0.080	0.082	0.074	0.080	0.095	0.093	0.091	0.105	0.118	0.120	0.116	0.117	0.109	0.105	0.105	0.096	0.091	0.089	0.081	0.079	0.075	0.072	0.084
0.068	0.082	0.074	0.000	0.068	0.076	0.077	0.077	0.075	0.077	0.077	0.075	0.068	0.000	0.077	0.090	0.088	0.085	0.098	0.109	0.000	0.113	0.110	0.102	0.098	0.098	0.090	0.085	0.083	0.079	0.000	0.069	0.066	0.077
0.067	0.081	0.075	0.071	0.068	0.000	0.068	0.068	0.000	0.068	0.068	0.000	0.068	0.071	0.077	0.089	0.085	0.082	0.093	0.099	0.106	0.107	0.000	0.101	0.097	0.000	0.089	0.084	0.000	0.075	0.069	0.062	0.061	0.072
0.074	0.093	0.089	0.086	0.084	0.082	0.083	0.083	0.082	0.083	0.083	0.082	0.084	0.087	0.091	0.100	0.091	0.087	0.096	0.100	0.103	0.103	0.105	0.099	0.095	0.094	0.087	0.082	0.080	0.072	0.066	0.061	0.060	0.069
0.110	0.108	0.106	0.105	0.104	0.103	0.103	0.103	0.102	0.102	0.102	0.103	0.103	0.105	0.107	0.112	0.124	0.111	0.119	0.122	0.123	0.122	0.120	0.116	0.112	0.108	0.102	0.096	0.091	0.084	0.077	0.072	0.069	0.074

Table B.14　Rodded B 体系　z=0～42.84cm

1.835	1.829	1.816	1.799	1.780	1.761	1.736	1.712	1.689	1.660	1.631	1.602	1.564	1.520	1.459	1.368	1.214	1.277	1.065	0.963	0.907	0.867	0.832	0.786	0.743	0.704	0.656	0.612	0.571	0.523	0.482	0.463	0.507	0.746
1.829	1.822	1.810	1.795	1.777	1.770	1.734	1.710	1.697	1.658	1.629	1.608	1.561	1.515	1.454	1.362	1.207	1.256	1.343	1.201	1.141	1.108	1.110	1.002	0.944	0.935	0.836	0.777	0.760	0.670	0.606	0.574	0.638	0.736
1.816	1.810	1.802	1.801	1.782	0.000	1.737	1.714	0.000	1.662	1.632	0.000	1.563	1.518	1.447	1.354	1.199	1.243	1.314	1.197	1.201	1.160	0.000	1.019	0.953	0.000	0.848	0.785	0.000	0.702	0.638	0.574	0.626	0.731
1.800	1.795	1.801	0.000	1.766	1.747	1.710	1.688	1.677	1.637	1.608	1.589	1.551	0.000	1.445	1.344	1.190	1.236	1.319	1.265	0.000	1.145	1.188	1.040	0.975	0.973	0.864	0.805	0.808	0.681	0.000	0.610	0.630	0.730
1.780	1.777	1.782	1.766	1.737	1.732	1.698	1.678	1.668	1.628	1.597	1.577	1.527	1.492	1.432	1.333	1.181	1.232	1.337	1.282	1.196	1.193	1.147	1.012	0.950	0.951	0.843	0.784	0.783	0.715	0.630	0.614	0.640	0.731
1.761	1.770	0.000	1.747	1.732	0.000	1.703	1.686	0.000	1.636	1.603	0.000	1.525	1.480	0.000	1.328	1.173	1.228	1.390	0.000	1.300	1.193	0.000	1.065	0.996	0.000	0.889	0.821	0.000	0.724	0.681	0.000	0.667	0.731
1.736	1.734	1.737	1.710	1.698	1.703	1.679	1.674	1.678	1.626	1.583	1.555	1.499	1.453	1.404	1.308	1.161	1.212	1.314	1.225	1.188	1.101	1.109	0.991	0.936	0.941	0.832	0.771	0.762	0.667	0.627	0.585	0.632	0.726
1.712	1.710	1.714	1.688	1.678	1.686	1.674	1.696	1.749	1.650	1.580	1.542	1.484	1.438	1.389	1.294	1.149	1.200	1.299	1.209	1.171	1.088	1.096	0.984	0.936	0.951	0.833	0.768	0.756	0.661	0.620	0.579	0.628	0.722
1.689	1.697	0.000	1.677	1.668	0.000	1.678	1.749	0.000	1.704	1.586	0.000	1.478	1.432	0.000	1.286	1.138	1.193	1.347	0.000	1.231	1.142	0.000	1.041	0.996	0.000	0.894	0.809	0.000	0.700	0.650	0.000	0.653	0.721
1.660	1.658	1.661	1.637	1.627	1.636	1.626	1.650	1.704	1.606	1.538	1.500	1.446	1.402	1.355	1.264	1.123	1.175	1.274	1.187	1.150	1.070	1.080	0.970	0.923	0.939	0.823	0.758	0.747	0.653	0.613	0.573	0.621	0.714
1.631	1.629	1.632	1.608	1.597	1.603	1.583	1.580	1.586	1.538	1.498	1.472	1.422	1.380	1.336	1.247	1.109	1.161	1.261	1.177	1.144	1.063	1.072	0.961	0.909	0.914	0.810	0.752	0.743	0.651	0.612	0.572	0.618	0.710
1.601	1.608	0.000	1.589	1.577	0.000	1.554	1.542	0.000	1.500	1.472	0.000	1.406	1.367	0.000	1.235	1.095	1.151	1.308	0.000	1.230	1.133	0.000	1.017	0.955	0.000	0.855	0.791	0.000	0.699	0.658	0.000	0.645	0.708
1.564	1.561	1.563	1.551	1.528	1.525	1.500	1.484	1.478	1.446	1.422	1.406	1.366	1.339	1.290	1.208	1.077	1.130	1.235	1.191	1.113	1.116	1.079	0.956	0.900	0.904	0.802	0.749	0.748	0.684	0.602	0.588	0.614	0.701
1.520	1.515	1.518	0.000	1.492	1.480	1.453	1.438	1.432	1.402	1.380	1.367	1.339	0.000	1.261	1.184	1.058	1.110	1.197	1.159	0.000	1.060	1.105	0.973	0.916	0.917	0.817	0.764	0.768	0.649	0.000	0.582	0.602	0.696
1.459	1.454	1.447	1.445	1.432	0.000	1.404	1.389	0.000	1.355	1.336	0.000	1.290	1.261	1.215	1.151	1.035	1.093	1.179	1.091	1.107	1.077	0.000	0.957	0.900	0.000	0.807	0.751	0.000	0.673	0.613	0.553	0.601	0.697
1.368	1.362	1.353	1.344	1.333	1.328	1.308	1.294	1.286	1.264	1.247	1.235	1.208	1.184	1.151	1.100	1.004	1.090	1.217	1.123	1.088	1.069	1.078	0.982	0.931	0.925	0.833	0.778	0.762	0.674	0.612	0.579	0.636	0.715
1.214	1.207	1.199	1.190	1.181	1.173	1.161	1.149	1.138	1.123	1.109	1.095	1.077	1.058	1.035	1.004	0.951	1.127	1.038	0.996	0.969	0.944	0.918	0.875	0.833	0.795	0.745	0.698	0.654	0.602	0.557	0.532	0.567	0.776
1.277	1.256	1.243	1.236	1.232	1.228	1.212	1.200	1.193	1.175	1.161	1.151	1.130	1.110	1.093	1.090	1.126	0.907	0.920	0.915	0.903	0.885	0.864	0.825	0.787	0.752	0.705	0.661	0.620	0.571	0.530	0.506	0.531	0.682
1.065	1.343	1.314	1.319	1.337	1.390	1.314	1.299	1.347	1.274	1.261	1.308	1.235	1.197	1.179	1.217	1.038	0.920	0.978	0.999	1.006	1.000	1.008	0.934	0.891	0.878	0.800	0.750	0.727	0.651	0.598	0.564	0.581	0.723
0.963	1.201	1.197	1.265	1.282	0.000	1.225	1.209	0.000	1.187	1.177	0.000	1.192	1.159	1.091	1.123	0.996	0.915	0.998	1.052	1.107	1.104	0.000	1.020	0.971	0.000	0.874	0.819	0.000	0.724	0.665	0.603	0.604	0.737
0.907	1.141	1.201	0.000	1.196	1.300	1.187	1.171	1.231	1.150	1.144	1.230	1.114	0.000	1.107	1.088	0.969	0.903	1.005	1.107	0.000	1.121	1.082	0.989	0.941	0.932	0.846	0.796	0.784	0.734	0.000	0.641	0.615	0.737
0.867	1.108	1.160	1.145	1.193	1.193	1.101	1.087	1.142	1.069	1.064	1.133	1.116	1.060	1.077	1.069	0.944	0.885	1.000	1.104	1.121	1.067	1.060	0.974	0.928	0.920	0.836	0.786	0.771	0.704	0.680	0.643	0.617	0.728
0.832	1.111	0.000	1.187	1.147	0.000	1.109	1.096	0.000	1.080	1.072	0.000	1.079	1.105	0.000	1.078	0.917	0.864	1.008	0.000	1.082	1.060	0.000	0.988	0.943	0.000	0.851	0.798	0.000	0.703	0.658	0.000	0.624	0.714
0.785	1.002	1.019	1.040	1.012	1.065	0.991	0.984	1.041	0.970	0.961	1.017	0.955	0.973	0.957	0.982	0.875	0.824	0.934	1.020	0.989	0.974	0.988	0.914	0.875	0.869	0.789	0.741	0.723	0.648	0.605	0.599	0.583	0.687
0.743	0.944	0.953	0.975	0.950	0.996	0.936	0.936	0.996	0.923	0.909	0.955	0.900	0.916	0.900	0.931	0.833	0.787	0.891	0.971	0.941	0.928	0.942	0.875	0.838	0.832	0.757	0.710	0.692	0.620	0.578	0.573	0.559	0.659
0.704	0.935	0.000	0.973	0.951	0.000	0.941	0.951	0.000	0.939	0.914	0.000	0.904	0.917	0.000	0.925	0.795	0.752	0.878	0.000	0.932	0.920	0.000	0.869	0.832	0.000	0.753	0.705	0.000	0.616	0.573	0.000	0.552	0.631
0.656	0.836	0.848	0.864	0.843	0.889	0.832	0.833	0.893	0.823	0.810	0.855	0.802	0.817	0.807	0.833	0.745	0.705	0.800	0.874	0.846	0.836	0.851	0.789	0.757	0.753	0.684	0.642	0.627	0.561	0.522	0.519	0.505	0.595
0.612	0.777	0.785	0.805	0.784	0.821	0.771	0.768	0.809	0.758	0.752	0.791	0.749	0.764	0.751	0.778	0.698	0.661	0.750	0.819	0.796	0.786	0.798	0.741	0.710	0.705	0.642	0.603	0.588	0.528	0.493	0.488	0.475	0.559
0.571	0.760	0.000	0.808	0.783	0.000	0.762	0.756	0.000	0.747	0.743	0.000	0.748	0.768	0.000	0.761	0.654	0.620	0.727	0.000	0.784	0.771	0.000	0.723	0.692	0.000	0.627	0.588	0.000	0.519	0.485	0.000	0.460	0.524
0.523	0.670	0.702	0.682	0.715	0.724	0.667	0.661	0.700	0.653	0.651	0.699	0.684	0.649	0.673	0.674	0.602	0.571	0.651	0.724	0.734	0.704	0.703	0.648	0.620	0.616	0.561	0.528	0.519	0.473	0.455	0.432	0.413	0.483
0.482	0.606	0.638	0.000	0.630	0.681	0.627	0.620	0.650	0.613	0.612	0.658	0.603	0.000	0.613	0.612	0.557	0.529	0.598	0.665	0.000	0.680	0.658	0.605	0.578	0.573	0.522	0.493	0.485	0.455	0.000	0.397	0.377	0.444
0.463	0.574	0.574	0.610	0.614	0.000	0.585	0.579	0.000	0.573	0.572	0.000	0.588	0.582	0.553	0.579	0.532	0.506	0.564	0.603	0.641	0.643	0.000	0.599	0.573	0.000	0.519	0.488	0.000	0.432	0.397	0.356	0.349	0.410
0.507	0.638	0.626	0.630	0.640	0.667	0.633	0.628	0.653	0.621	0.618	0.645	0.614	0.602	0.601	0.636	0.566	0.530	0.581	0.604	0.616	0.617	0.624	0.583	0.559	0.552	0.505	0.475	0.460	0.413	0.377	0.349	0.344	0.395
0.746	0.736	0.731	0.730	0.731	0.731	0.726	0.722	0.721	0.714	0.710	0.708	0.701	0.696	0.697	0.715	0.776	0.682	0.723	0.737	0.737	0.728	0.714	0.687	0.659	0.631	0.595	0.559	0.524	0.483	0.443	0.410	0.395	0.426

C. OpenCL による開発

C.1.　GPU 開発環境

GPU を一般的な計算機として用いるには、専用の開発環境(フレームワーク)が必要である。3.4 節で述べたように、本研究では OpenCL を用いているが、OpenCL 以外にも GPGPU 用のフレームワークは存在する。OpenCL 開発の実際の手順については本項で述べるが、その前に各フレームワークの簡単な説明を行う。

CUDA は計算機シミュレーションの分野で比較的古くから使われており、信頼性の高い枯れたフレームワークだと言える。GPU ベンダーである NVIDIA が開発しており、Nvidia 製の GPU 上で動作する。情報量も多いため、開発は行い易い。ただし、NVIDIA 製 GPU 以外では動作せず、CUDA によって記述されたコードを利用し続けられるかは NVIDIA 一社次第で変わる可能性がある。

OpenCL は標準化団体 Khronos によって策定されているフレームワークである。GPU だけでなく CPU やメニーコア(Xeon Phi)、さらには FPGA などでも OpenCL を用いた開発が行える。CUDA に比べると普及度は低い。また、API は比較的煩雑で低水準であり、プログラミングは若干し辛い。ソースコード自体は CPU でも GPU でも共通でコンパイル出来るが、高速化のためにはデバイス毎の最適化が必要になる。利点としては、GPU が無い場合でも CPU 上で実行できることが挙げられる。また、現状(2016 年)時点では殆ど唯一の標準フレームワークである。

以上 2 つのフレームワークより簡単に GPU プログラミングが出来るように、いくつか新しいフレームワークが登場している。OpenACC(http://www.softek.co.jp/SPG/Pgi/OpenACC/)がその 1 つで、OpenMP のようにソースコード上に特殊な文を挿入することで並列化を指定する。現在は商用コンパイラでしかサポートされていない。C++AMP(https://msdn.microsoft.com/ja-jp/library/hh265137.aspx)は Microsoft と AMD によって開発されているフレームワークで、並列計算用に C++を拡張している。

2016 年現在、GPGPU や並列計算用のフレームワークは、長年使われて枯れているものが少なく、新しいものが複数乱立している。変化の激しい業界でもあるため、よく調べてからどのフレームワークを選択するか決めるべきであろう。

C.2.　開発環境の導入

OpenCL の開発環境は、各ハードウェアベンダーから提供されている。2014 年 1 月時点では、以下の URL において配布されている。

- AMD
  http://developer.amd.com/tools-and-sdks/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/

- Intel
  http://software.intel.com/en-us/vcsource/tools/opencl-sdk
- Nvidia
  https://developer.nvidia.com/cuda-downloads

Linux 用と Windows 用がそれぞれ存在するので、OS の種類に応じてパッケージをダウンロードし、システムにインストールする。AMD や NVIDIA の GPU 上で OpenCL を実行する場合は、同じベンダーのデバイスドライバがシステムにインストールされている必要があるので注意すること。

C.3. コンパイル方法

OpenCL の API は、C 言語から呼び出せるようにヘッダファイルとライブラリファイル(*.so や*.lib)が用意されている。これらのファイルがおかれるパスは提供ベンダーによって異なるため、詳しくはフレームワークのマニュアルを参照されたい。

GCC であれば、ソースコードのファイル sample.c をコンパイルして実行ファイル sample を作成したい場合、以下のコマンドでコンパイルできる。

```
gcc -c -I(ヘッダファイルのあるディレクトリのパス) sample.c
```

リンクは次のように実行する。

```
gcc -o sample sample.o -L(ライブラリのあるディレクトリのパス) -lOpenCL
```

Visual Studio などの IDE を用いている場合は、IDE の設定に追加のヘッダファイル・ライブラリの探索パスを追加する。

また、C++を用いる場合も C 言語の API を直接呼び出せる。また、Khronos Group が提供する C++用のバインディングを利用することが出来る(http://www.khronos.org/registry/cl/)。

C.4. OpenCL の実行フロー

OpenCL は、API を通して「オブジェクト」を取得・作成し、オブジェクトに対する操作を行う設計となっている。OpenCL に演算処理を行わせたい場合、主に以下のオブジェクトを作成しなければならない。

- プラットフォーム

  OpenCL フレームワーク全体を示す。OpenCL を利用する場合、まずプラットフォームを取得する。

- デバイス

  演算を行わせるデバイスを示す。システムに存在する CPU、GPU などをデバイスとして扱う。

- プログラム・カーネル

  OpenCL C のソースコードからプログラムを作成する。プログラムは利用対象のデバイスに合わせてカーネルに変換される。

- コンテキスト

  デバイス、メモリ領域、カーネル等をまとめた実行環境の単位。デバイスやメモリ操作はコンテキスト単位で行われる。

- コマンドキュー

  OpenCL では、メモリのコピー、カーネルの実行などはコマンドを発行し、キューに積むことで行われる。

- バッファ

  デバイスからアクセスが可能なメモリ領域。グローバルメモリかコンスタントメモリのメモリ領域としてカーネルに渡される。

デバイスを柔軟に制御できるよう、様々な API とオブジェクトが OpenCL には用意されているが、基本的な流れは Fig. C.1 のようになる。

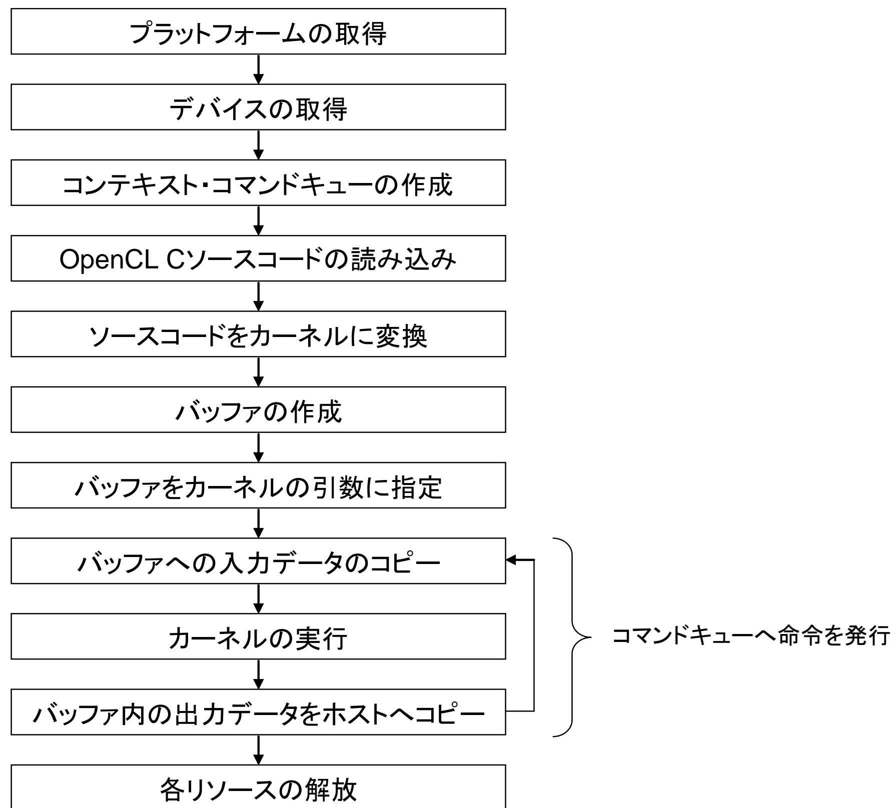

Fig. C.1 OpenCL のデバイス準備から終了までの実行フロー

C.5. カーネルの例

単純な演算処理について、OpenCL C によるカーネルの記述方法を示す。

長さ 100 の単精度浮動小数点配列について、それぞれの要素で和をとるような処理は C 言語では次のようになる。

```
float a[100], b[100], c[100];
…
for(int i = 0; i < 100; i++){
    c[i] = a[i] + b[i];
}
```

この for ループの処理を 100 個のワークアイテムに割り当て、分割する場合、カーネルは次のようになる。このカーネルは、実行する時に並列化数を 100 と指定されて実行される。

```
__kernel void addArray(__global float *a, __global float *b, __global float *c){
    int i = get_global_id(0);
```

```
        c[i] = a[i] + b[i];
    }
```

カーネル関数には、引数としてグローバルメモリへのアドレスが渡され、配列として利用する。自分自身のインデックス値(ID)は get_global_id 関数によって得られる。

公刊論文リスト

[1] T. Okubo, T. Endo, A. Yamamoto, "A Multi-level Parallel Computation of Reactor Cores using GPU for Loading Pattern Optimization," *Proc. PHYSOR2014*, Kyoto, Japan, Sep. 28 - Oct. 3, (2014).

[2] T. Okubo, T. Endo, A. Yamamoto, "Efficient Execution of Monte Carlo Simulation Based on Pseudo-Scattering using GPU," 2015 ANS Annual Meeting, San Antonio, TX, June 7 – 11, (2015).

[3] 大久保卓哉, 遠藤知弘, 山本章夫, "仮想散乱に基づく GPU を利用したモンテカルロ計算の高速化," 日本原子力学会 2015 秋の大会, 静岡大学, (2015).