

# スクリプト言語 Xcrypt による 格子 QCD シミュレーションの最適化

齋藤 華<sup>†1,†2</sup> 朴 泰 祐<sup>†1</sup> 金谷 和 至<sup>†2</sup>  
埴 敏 博<sup>†1</sup> 佐藤 三 久<sup>†1</sup>

本研究では、小規模問題に対する格子 QCD シミュレーションのパラメータサーチを最適化する。複数ジョブによりパラメータ空間の探索を行う場合、ジョブ間に依存関係があるため、より多くの人的コストがかかってしまう。ここではスクリプト言語 Xcrypt によって、依存するジョブの実行をコントロールしジョブ実行を自動化することで、バッチスケジュールや複数ジョブの管理による煩雑さを回避し、人的コストの削減を目指す。また、自動化による問題点を確認し、物理シミュレーションが正しく行われるための条件を明確化する。

## 1. はじめに

物質の最小単位は素粒子であり、クォークはその素粒子の一つである。約 1 兆度以上の超高温におけるクォーク物質の性質を解明することは、宇宙の初期進化を解明する上で重要である。グルーオンによって媒介されるクォーク間の力は QCD(量子色力学)で記述されるが、相互作用が極めて強いために、時空を格子で離散化した格子 QCD に基づく数値シミュレーションが、クォーク物質の性質を研究する唯一の系統的方法になっている。最高速のスーパーコンピュータを使った大型計算が進められており、並列化やアルゴリズム開発による計算の高速化・最適化も盛んに研究されている<sup>1)</sup>。

本研究では、大規模計算ではなく小規模な問題の繰り返しに注目する。ここでは、格子 QCD シミュレーションにおいて、パラメータを変えながら小規模ジョブを繰り返し実行す

<sup>†1</sup> 筑波大学システム情報工学研究科

Systems and Information Engineering, University of Tsukuba

<sup>†2</sup> 筑波大学数理物質科学研究所

Graduate school of Pure and Applied Sciences, University of Tsukuba

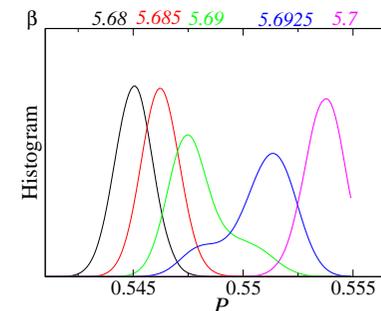


図 1 物理シミュレーションにおいて必要となるヒストグラム

ることによってパラメータ空間を探索するタイプの問題に、スクリプト言語 Xcrypt によるジョブの制御を導入する。依存関係のあるジョブの管理には煩雑さが伴う。Xcrypt を用いて、それを回避する。格子 QCD シミュレーションにおけるパラメータサーチの一例全体を最適化することが最終目標であり、本研究はその第一段階として行う。

## 2. 物理シミュレーションの全体像

### 2.1 有限温度格子 QCD シミュレーションとクォーク物質の相構造

クォーク物質の相転移の解明は重要な研究テーマの一つである<sup>2)</sup>。相転移の性質は相転移点の極近傍でしか見られないため、その解明にはパラメータ空間の探索が必要となる<sup>3)</sup>。具体的な操作を説明する前に、得たいデータの特徴について述べる。

相転移を調べるためのシミュレーションにおいて、必要となるデータは図 1 のようなものである。これはブラケット  $P$  と呼ばれる物理量 (エネルギー密度に相当) の確率分布関数である。一度のジョブでのパラメータ  $\beta$  ( $\beta$  は温度に対応) に対して確率分布関数のピークが一つ得られる。ピークの位置や幅は  $\beta$  によって変化する。ジョブを繰り返し、複数の確率分布関数を求める事で図 1 が得られる。一次相転移点の極近傍ではダブルピーク、それ以外の点ではシングルピークのグラフがそれぞれ得られるので、確率分布関数から 1 次相転移の情報を引き出す上で重要となるのは以下の二つの特徴である：

特徴 1 ダブルピークの兆候が見られる  $\beta$  があること (図 1 の緑や青の  $\beta$ )。

特徴 2 範囲内の全ての  $P$  に対してヒストグラムに 1 個以上の 0 でない値があること。より厳密には、1 次相転移点をカバーする広い  $P$  の範囲で十分な統計精度があること。

## 2.2 シミュレーション全体の流れ

次に相転移点を探すシミュレーションの自動化の具体的な手順について説明する。本シミュレーションの具体的な目的は、予想されている範囲の中で、一次相転移の特徴を求めてパラメータ空間を探索することである。ピークの位置や幅の情報はだまかには予想できるが、正確には予想できない。つまり、上記の特徴を捉えるためにはヒューリスティックな処理を行い、トライ&エラーを繰り返してパラメータ空間を探索しなければならない。これを実際に行うためには図2のようなフローチャートに従ってシミュレーションを行う必要がある。このシミュレーションは計算の中心であるモンテカルロ法部分と、統計量のチェックと $\beta$ のチェックを行いヒストグラムの作成をガイドする条件分岐部分に区別される。モンテカルロ法は一般に、乱数を使って期待値を評価するために用いられる。ここでは特に、期待値計算に用いる確率分布を抽出している。また、条件分岐のうちの分岐1, 2では、各場合に対してパラメータが変更されてモンテカルロ法が繰り返されるような構造となっている。

条件分岐部分について詳しく説明する。条件分岐1はシミュレーションを継続して行うか、又は異なる値の $\beta$ を用いて始めからやり直すかを表している。これは統計精度によって決められるもので、研究の進捗に合わせて変更する可能性がある。また、条件分岐2は一次相転移点が見つけれられたかどうかを表す条件分岐になっており、図3のようなデータを用いて判定する。図3はヒストグラムの指数関数の符号を反転させたものの曲率を表す。曲率の値が負になる領域があればヒストグラムの傾きが増加することを意味する。ヒストグラムの傾きが増加することはダブルピークの兆候があるということでもあるので、この特徴を用いれば、単純な条件で一次相転移点の有無を判断できる。相転移点を十分に特定できた場合はジョブの実行を終了する。一方、特定できない場合には $\beta$ の新しい値を設定して次のジョブを投入する。次のジョブで用いる $\beta$ の値は、ピークの裾が重なるように選ぶ。ピークが正確なガウス分布になっている場合には、この条件を満たす $\beta$ を正確に予想できる。しかし、相転移点の近傍ではピークはガウス分布にはならない(図1の緑や青の線)ので、条件を満たすような $\beta$ を正確に選ぶことは難しくなる。ここでは、上記の方法で予想される $\beta$ を基準に選び、それより狭い間隔としてその半分の値を選択することにする。

これらの条件分岐の判定は計算コストが最も多くかかるモンテカルロ法部分の結果に対して行われるが、独立に処理される。またモンテカルロ法部分は格子QCDシミュレーションに共通して用いられるものであり、アルゴリズムの改良や最適化が盛んに行われている。そのため、これらを同一コードに書き合わせてしまうとコードが複雑になってしまう。コード生成や性能チューニングといった面での生産性を高めるために、モンテカルロ法部分とそ

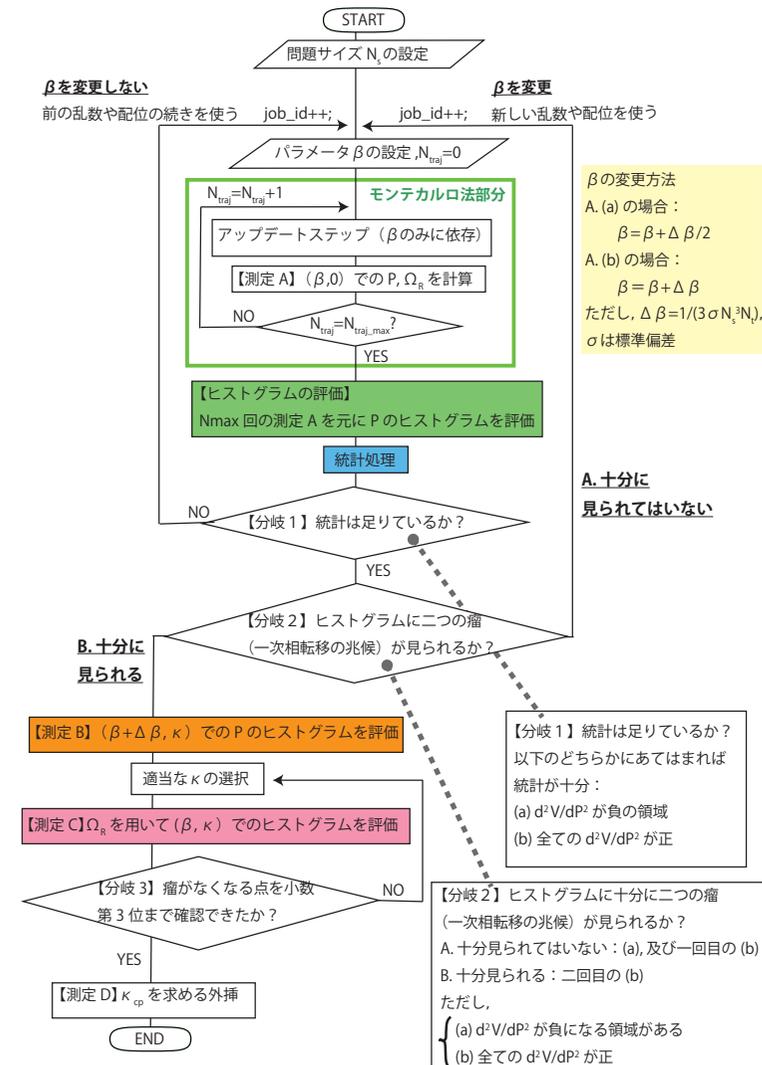


図2 フローチャート

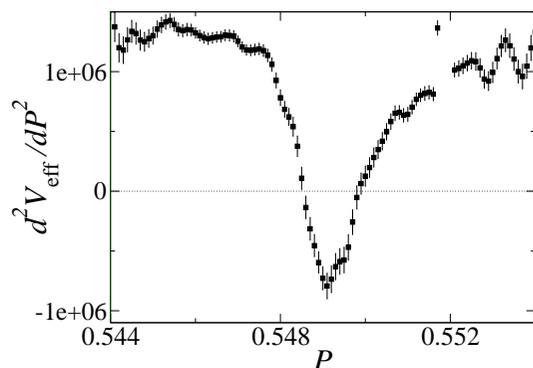


図3 条件分岐2で用いられるデータ

他の条件分岐部分は別々に扱い、シミュレーション全体をスクリプト言語で自動制御できるようにすることが望ましい。

### 3. Xcrypt を用いた格子 QCD シミュレーションの実装

#### 3.1 Xcrypt とその利点

図2のフローチャートでは次のシミュレーションパラメータの値を選ぶ時に前の解析結果を反映する必要がある。このトライ&エラーとなる部分が全体の制御をより難しくしている。また、モンテカルロ法の計算プログラムは一般性があるので、これを独立したプログラムとし、入力パラメータを変えながら実行する形にしたい。一方で、このようなシミュレーションパターンは格子 QCD シミュレーションに限らず、科学技術計算では頻繁に見られるため、アプリケーション側のユーザが容易にプログラムでき、煩雑さを避けられるような言語が求められる。Xcrypt は、そのようなニーズに応えるために提案されている<sup>4)-6)</sup>。

Xcrypt はスーパーコンピュータ上の計算科学分野のシミュレーションを対象とし、同一のプログラムを異なるパラメータで大量実行することを自動化する目的で作られた。その特徴として、バッチ処理の煩雑さからの解放とディレクトリ操作やコーディングの容易さが挙げられる。現在、スーパーコンピュータのジョブ投入はバッチ処理によるものが主流となっている。シミュレーションを繰り返す場合、このバッチ処理を行うスクリプトが大量となり、その管理が煩雑となる。また、スクリプトに可搬性がないためにマシンによって書き換えな

表1 Xcrypt が提供するコマンド

コマンド名	内容
xcrypt <i>myscript.xcr</i>	実行
xcryptdel <i>jobID1 jobID2 ...</i>	実行中のジョブを殺す (finished なジョブは無視)
xcryptdelall	
xcryptcancel <i>jobID1 jobID2 ...</i>	実行中のジョブを殺す (finished なジョブも aborted にする)
xcryptcancelall	
xcryptinvalidate <i>jobID1 jobID2 ...</i>	実行中のジョブを強制的に finished にする (二度と実行されない)
xcryptstat	実行中のジョブ一覧を表示
xcryptclean	実行中のジョブを全て殺し、履歴情報を忘れる

ければならないという不便さもある。Xcrypt はバッチ処理に関するこれらの問題点を解消できるよう設計されている。また、ジョブを繰り返す場合には、ジョブを実行するディレクトリとデータを保存するディレクトリに注意しなければならない。例えば、ディレクトリを混同してインプットファイルを誤って読み込んでしまう可能性もある。Xcrypt の sandbox 機能を用いていることで、ディレクトリの操作が容易に行え、このようなトラブルを避ける事ができる。さらに、Xcrypt は perl をベースに作られている。このため、アプリケーション側のユーザでも容易にコーディングできるような設計となっている。

#### 3.2 Xcrypt によるジョブコントロール

本節では Xcrypt の実行手順について説明する。科学技術計算で見られるトライ&エラーによるシミュレーションは一般にパラメータの決定、ジョブスクリプト生成、ジョブ投入、終了待ちの繰り返しと見なす事が出来る。これらの操作を行うために、Xcrypt ではジョブ実行インターフェイスが以下の三つの部分から構成されている：

- パラメータの決定：**@jobs=prepare(%template)**
- ジョブオブジェクト生成、ジョブ投入：**submit(@jobs)**
- ジョブ終了待ち：**sync(@jobs)**

Xcrypt ファイルは上記の関数を含み、解析部分は perl によって記述される。このようにしてつくられたファイルを表1中の実行コマンドによって実行する。また、Xcrypt が提供するその他のコマンドについても表1に挙げる。

#### 3.3 実 装

本実装ではパラメータの決定、ジョブスクリプト生成、ジョブ投入、終了待ちを組み合わ

```

use base qw (sandbox core);
for($job_id=0; $job_id<$job_num; $job_id++){
    #  $\beta$  の設定 (インプットファイルの生成)
    open Inputfile, "> gaugeprm.001"
        printf Inputfile "%lf\n", $beta;
        ...
    close Inputfile;

    %template = (
        'id' => "job$ job_id",
        'exe0' => './goupd_sx.sh', # exe file
        'copiedfile1' => 'goupd_sx.sh',
        'copiedfile2' => 'gaugeprm.001',
    );
    @jobs=&prepare_submit_sync (%template);

    # ヒストグラムの評価, 統計処理
    # (シミュレーション結果の解析)
    ...

    # 条件分岐 1,2 (次のパラメータの値の決定)
    ...

} # end of job_id
    
```

図 4 Xcrypt コードの例

表 2 実験環境 (CCS flare16)

cpu	Xeon X5550 ( 4-core x 2 socket ) 2.66GHz
メモリ容量	32GB memory
OS	CentOS 5.5 (x86_64)
compiler	Intel compiler 11.1

表 3 シミュレーションパラメータ

問題サイズ	$24^3 \times 4$
$N_{sweep\_max}$	70
$\beta$ の初期値	5.6800

せた以下の関数を用いる：

**@jobs=&prepare\_submit\_sync (%template);**

Xcrypt コードには、この他にジョブ実行に必要な情報である%templateを Xcrypt に渡す操作の部分がある。また、perl で記述する部分として、インプットファイルの生成、次のシミュレーションのインプットパラメータの値の決定を行う箇所もある。具体的には図 4 にあるようにまとめた。ただし、本実装ではフローチャート図 2 の条件分岐 2 までの実装を行った。この実装は特定の領域の全ての  $P$  に対してヒストグラムが 1 個以上の 0 でない値を持つことを確認することが目的である。今回はテストとして  $job\_id < 3$  とした。

## 4. 評 価

### 4.1 評価環境と結果

実験環境を表 2 にまとめる。また、シミュレーションパラメータを表 3 に示す。

結果として、図 5 が得られた。Xcrypt の条件分岐により選択された  $\beta$  は 5.6955, 5.71131 であり、範囲内の全ての  $P$  に対してヒストグラムに 1 個以上の (0 でない) 値を得ることができた。これより、 $\beta$  の統計精度に応じて、次の  $\beta$  の値が正しく設定できていると言える。一次相転移の特徴であるダブルピークの構造はまだ見られていないが、このシミュレーションを繰り返すことにより、その特徴を漏らさずにパラメータ空間全体を探索することができるはずである。したがって、フローチャート図 2 で表されるシステムによりパラメータ空間を自動で探索できることがわかった。

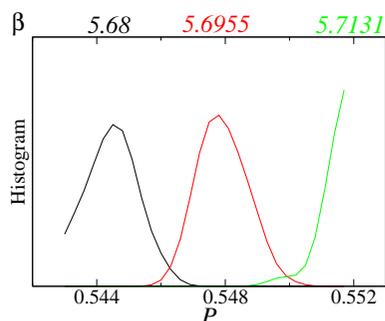


図 5 Xcrypt によって自動で生成されたヒストグラム

#### 4.2 考 察

今回の実装では今後に向けて課題が得られた。まず、相転移点を特定できるように修正しなければならない。今回の実装では相転移点を特定できなかった。図 1 ではダブルピークの兆候 (図の緑と青) が見られているのに対して、図 5 にはそのような形状は見られないことがそれに対応する。これは各  $\beta$  での統計が十分ではなかったためと考えられる。ここではジャックナイフ法によってエラー評価を行っているが、この評価方法ではデータの数が少ないとエラーを過小評価してしまう可能性がある。本来ならば、統計が不十分であった場合、条件分岐 1 において同じ  $\beta$  の値でジョブが繰り返されるはずであった。しかし、エラーが過小評価されたためジョブが繰り返されなかった。つまり、今回はテストとして行ったためデータが少なくエラー評価が正しくなされずに、相転移点が上手く特定できなかったと考えられる。次回の実装では過小評価されないように  $N_{sweep\_max}$  をより大きな値に変更する必要がある。

また、今後の方針として、より現実的なシミュレーションを具体化していきたい。フローチャートには記入されていないが、実際には初期配位に依存しないように始めのデータを捨てる。今回はテストだったのでデータ数 100 のうちの 30 しか捨てておらず、捨てる量が少ない。この影響で相転移点を特定できなかった可能性があるため、この点を十分注意する必要があるだろう。

#### 5. おわりに

格子 QCD シミュレーションにおけるスクリプト言語 Xcrypt によるジョブコントロールを検討した。また、フローチャート図を用いてシミュレーション全体のモデル化を行い、全

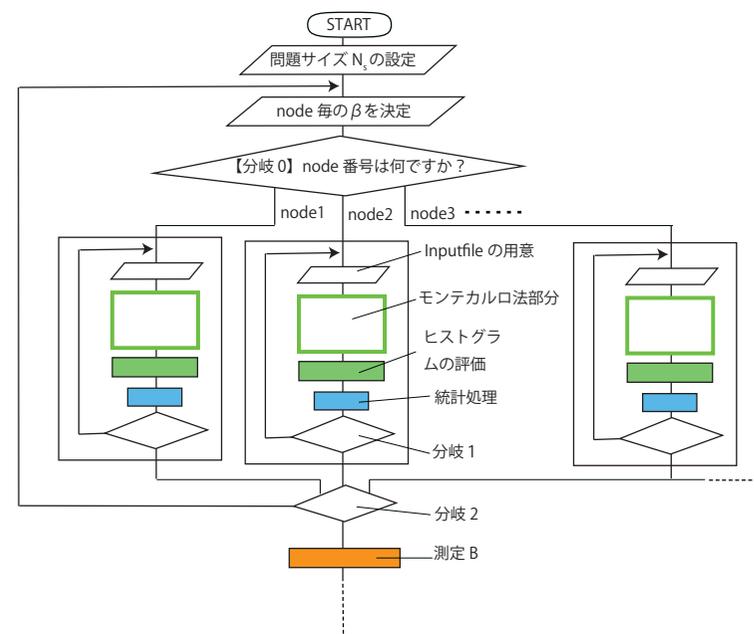


図 6 複数ノードで実行した場合のフローチャート

体を制御するスクリプトを Xcrypt を用いてプログラミングした。さらに、Xcrypt の機能を用いて、ヒューリスティックな操作の一部を自動化することに成功した。

今後の進展として、シミュレーション全体の自動化とモンテカルロ法部分の並列化を検討している。計算の主体はモンテカルロ法のシミュレーションであり、比較的小規模だが、並列性は十分にある。そこでマルチコア構成の計算ノードを用いて計算を加速する。並列化は OpenMP によるものを考えている。さらに、図 6 にあるように複数ノードでモンテカルロ法とそれに伴う解析部分の処理を行い、それらの結果を集計して次のパラメータを決めるといったシステムを検討していきたい。最終的には、筑波大学の T2K-Tsukuba<sup>7)</sup> 上での実装を目標とする。これらの自動化や並列化を実現し、シミュレーション全体を最適化したいと考えている。

## 謝 辞

本研究を進めるにあたり，Xcrypt の使い方，動作確認等をご指導頂いた，京都大学学術情報メディアセンター中島浩教授並びに，平石拓助教に深く感謝致します。

## 参 考 文 献

- 1) Ken-ichi Ishikawa: Recent algorithm and machine development for lattice QCD, [arXiv:hep-lat/0811.1661]
- 2) K. Kanaya : Finite Temperature QCD on the Lattice – Status 2010, [arXiv:hep-lat/1012.4247]
- 3) H. Saito et al. : Phase structure of finite temperature QCD in the heavy quark region, [arXiv:hep-lat/1106.0974]
- 4) <https://www2.cc.u-tokyo.ac.jp/procon2010/Tutorial/Xcrypt.pdf>
- 5) 平石 拓, 安部 達也, 三宅 洋平, 岩下 武史, 中島 浩 : 柔軟かつ直観的な記述が可能なジョブ並列スクリプト言語 Xcrypt, 先進的計算基盤システムシンポジウム (SACSYS2010), P.183-191
- 6) [http://super.para.media.kyoto-u.ac.jp/xcrypt/Xcrypt\\_manual.pdf](http://super.para.media.kyoto-u.ac.jp/xcrypt/Xcrypt_manual.pdf)
- 7) <http://www.ccs.tsukuba.ac.jp/CCS/t2k-tsukuba>