スーパーコンピュータを使おう（5）

石原　卓

Ⅰ．はじめに

連載 [1–4] の 5 回目となる本稿では，スーパーコンピュータを必要とする具体的な問題として，「乱流の直接数値計算」を取り上げ，スーパーコンピュータを用いた大規模並列計算の舞台裏を紹介します。本稿が名古屋大学情報連携基盤センターのスーパーコンピュータ（HPC2500[5] やその次期のマシン）を始めとして，機会があれば国内外のスーパーコンピュータを使ってみたいと考えている学生さんや研究者の皆様の参考になれば幸いです。

以下では，「乱流の直接数値計算」において，スーパーコンピュータを必要とする理由，具体的な問題設定と計算のコアとなる部分，並列プログラムの概要と開発の方針，及び，自分自身の体験について述べたいと思います。

Ⅱ．スーパーコンピュータを必要とする理由

我々の身の回りの空気や水の流れは，非圧縮ナビエ・ストークス方程式という流体の方程式によって非常によく記述されます。その方程式は決定論的であるため，初期条件と境界条件さえ与えれば後の流れの発展は定まることになります。しかし，その方程式は非線形であり，解析的な解を見つけることは困難です。また，計算機を用いて方程式の数値解を求めることは原理的に可能ですが，非線形性の強さを表す無次元パラメータのレイノルズ数 Re が大きくなるにつれ，流れは初期条件のわずかな違いに敏感となり，流れの詳細の数値予測は困難になります。さらに，高 Re の乱れた流れ（乱流）の中には大小さまざまなスケールの変動（渦）が混在し，外力や初期条件，境界条件によって決まる最も多くのエネルギーを保有する大きい渦と，粘性による散逸が卓越する小さい渦のスケール比は Re の 3/4 乗に比例して大きくなることが知られています。そのため，高 Re 乱流の全てのスケールの渦を解像する直接数値計算は Re の 9/4 乗に比例する主記憶を必要します；われわれの身の回りの流れの例として，ティーカップの中の流れは $Re = O(10^4)$，野球のピッチャーが投げるボールのまわりの流れは $Re = O(10^5)$ であるので，これらの流れを直接計算によって解像するためには，おおよそ数 10GByte から数 TByte の主記憶が必要となります。また，数値不安定を避けるため時間刻みを十分に小さくとる必要があります。その時間刻みと大きいスケールの渦の時間スケールの比も Re の 3/4 乗に比例するため，トータルで Re の 3 乗に比例する計算コストがかかることになります。したがって，高 Re の乱流における小スケール渦の動きまで解像する「乱流の直接数値計算」ではスーパーコンピュータが必要不可欠

になります。

Ⅲ．問題設定と計算のコアになる部分

乱流は未解決な問題であり，小スケールの渦がどのような統計法則に従っているのかといった現象論的な理解も未だ十分とはいえません。そのため，高 *Re* の乱流における小スケール渦の普遍統計法則を解明するためには，使える計算機を駆使した，できる限り簡単な問題設定の，できる限り大きい *Re* を目指した「乱流の直接数値計算」が必要になります。その際，数値計算法としてはできる限り高精度かつ高解像度のものを用います。そこで，簡単な問題設定として周期境界条件を採用し，外力のある非圧縮ナビエ・ストークス方程式を，フーリエ・スペクトル法を用いて数値的に解くことになります。周期境界条件を用いる理由は，乱流の細かいスケールの統計は大きいスケールの渦の振る舞いの詳細にはよらないと一般に考えられており，かつ，この条件下では流体の方程式を非常に精度良く数値シミュレーションできるからです。

フーリエ・スペクトル法では，流れ場をフーリエ級数で展開し，そのフーリエ係数の時間発展を数値的に求めることになります。フーリエ・スペクトル法では微分演算を波数空間における代数演算で行えるため差分近似誤差のようなエラーは介入しません。また，非圧縮性を満たすように圧力を求める演算（これは差分法において最も計算コストのかかる部分）も代数的に行うことができて効率的です。なお，方程式中の非線形項を評価するために，一旦，速度等の従属変数を波数空間から実空間に変換して，非線形項を計算し，それをまた波数空間に戻すといった操作が必要になりますが，この波数空間⇔実空間の変換も高速フーリエ変換（FFT）を用いることにより，効率的に実現可能です。

このように，フーリエ・スペクトル法を用いた乱流の直接数値計算においては，3 次元 FFT を用いることにより効率的かつ高精度に流体方程式を解くことが可能です。それでは実際の計算ではどれくらいの頻度で 3 次元 FFT を行うのでしょうか？　非線形項を評価する際に生じるエリアジング誤差を位相シフト法で完全に除去し，かつ時間発展に 4 次のルンゲクッタ法を採用した場合，時間刻み 1 ステップの間に実に 72 回の 3 次元実 FFT（3 次元複素 FFT なら 36 回）を行う必要があります。実際，計算時間の約 8 割以上を 3 次元 FFT のために使用していることになります。よって，効率的に乱流の直接数値計算を行うためには如何に効率的な 3 次元 FFT を実装するかが重要になります。

Ⅳ．並列プログラムの概要と開発の方針

Ⅲで，効率的な 3 次元 FFT の実装が，乱流の直接数値計算において重要であることを述べましたが，大規模な数値計算では非常に大きなメモリを必要とするため，並列計算ではほぼ間違いなく分散メモリを使うことになります。3 次元 FFT は基本的に 1 次元 FFT を 3 方向に施せばよいのですが，同一分散メモリ空間内にない方向に 1 次元 FFT を施すためにはそれに先立ち，一度大掛かりなデータ転送を行っておく必要があります（図 1 参照）。

必要な操作は単純なのですが，実際にプログラムを作成し，その効率を追及するにはハイパ

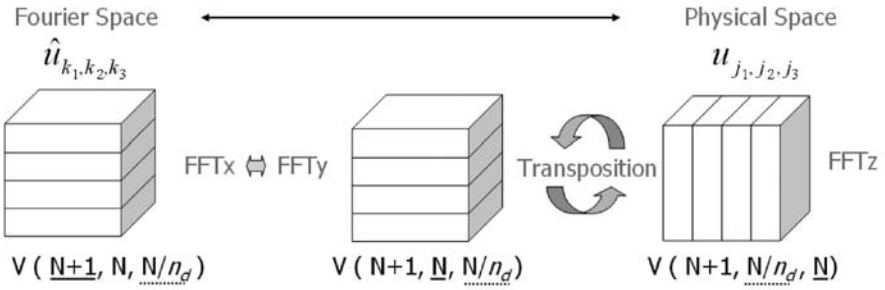

図１　３次元 FFT の並列計算の概要．実線の下線部の軸が FFT 用，破線の下線部の軸が並列計算用，ベクトル型計算機では残りの軸をベクトル化用に使う．

フォーマンスコンピューティングの経験と知識が必要となります。すでに使用するスーパーコンピュータ用に最適化された科学計算ライブラリがあるなら，それを用いればよいのですが，公開されて間もない新しいスーパーコンピュータ用には，そのようなライブラリがまだ用意されていない場合が多く，自力で開発する必要があります。スーパーコンピュータのハード特性と相性の良いアルゴリズムを選択し，そのプログラムを開発し，最適化していく作業は，多くの専門知識を必要とするため簡単ではありません。実際問題としては，MPI＋Fortran 77 や 90 など汎用性が高くかつ結果の検証済のプログラムにて，その部分ごとに計算時間を把握し，いくつかあらかじめできる最適化を行ったのち，プログラミングの経験者やハイパフォーマンスコンピューティングの専門家に相談するのが確実だと思います。

Ⅴ．並列化プログラム開発の体験記

私の手元には多くの乱流の直接数値計算コードがあります。古くは，①名古屋大学の VP2600[6] 用に Fortran77 で記述したもの，②名古屋大学の VPP500[7] 及び VPP5000[8] 用に VPPFortran[9] で記述したもの，それを基盤にして，③ MPI＋Fortran77 で作成しなおし地球シミュレータ [10] 用に最適化したもの，④②を基準にして HPC2500[5] 用に XPFortran[11] で記述したもの，⑤③を基にして，名古屋大学の HPC2500 用に MPI＋Fortran で作り直したもの，などです。以下に，簡単にこれらのプログラムの特徴を述べます。

① VP2600[6] はベクトル型計算機で，プログラムでは単純に（i）最内側ループを長くする，(ii) メモリのバンクコンフリクトを避けるなどに気をつけました。この頃は最適化をあまり意識していませんでしたが，あるループがベクトル化できて，プログラムが数 10 倍速くなったというような体験をしたことがありました。

② VPP500[7] は名古屋大学に最初に導入された並列ベクトル型計算機で，各プロセッサはベクトル型なので，基本的に①と同様なことに気をつけつつ，分散メモリの並列処理をするために，VPPFortran[9] の並列指示行を加えました。基本的にローカル変数のみで可能な演算を spread do の指示行下で行い，グローバル変数を必要とするデータ転送には，spread move を用いるといった感じでした。いずれにせよ高速化の要はベクトル化率を上げることで，spread

move は遅いので極力使わないといった方針のプログラムでした。VPP500 はちょうど並列計算機が世の中に普及し始めたころのマシンで，並列計算を使いこなせる人が周りにいなかったため苦労しましたが，その分，当時世界最大規模の 512 の 3 乗の乱流直接数値計算が実現できたときは感動しました。次いで，名古屋大学大型計算機センターのスーパーコンピュータが VPP500[7] から VPP5000[8] への移行したときには，プログラムの変更の必要がなかったので，移行後直ちに，当時の世界最大規模である 1024 の 3 乗の乱流 DNS が世界初で実現できました。

③地球シミュレータ [10] は 2002 年 3 月に公開されてから，実に 2.5 年にわたり世界 top をキープした並列ベクトル型のスーパーコンピュータです。全体で 640 ノードある各ノードに 8GFlops のベクトル型プロセッサが 8 つと 16GB の共有メモリがあり，640 ノードはすべてクロスバースイッチにより 1 対 1 でつながっています。手持ちの②のプログラムは VPP 用に特化したものだったので，地球シミュレータ用に新たに開発し直す必要がありました。ただし，この開発は，地球シミュレータの性能を正しく評価することが目的のハイパフォーマンスコンピューティング（HPC）の専門家と共同研究で行うことができたので私自身はあまり苦労せず効率的なプログラムを手に入れることができました。HPC の世界では有名なゴードンベル賞をねらってプログラムを開発したので，プログラムでは高速化のためのさまざまな工夫がなされています [12]。工夫その 1 は基数 4 の FFT の採用です：地球シミュレータのひとつのベクトルプロセッサの演算速度が 8GFlops で，ベクトルプロセッサとメインメモリ間の通信速度が 32GByte/sec なので，演算が 8 回できる時間に 8Byte の倍精度変数が 4 つしかロード / ストアができないことになります。通常の 2 を基数とする FFT では割合として 8（ロード / ストア）に対して 10 回の演算しかしないので，プロセッサのデータ待ち状態が生じますが，4 を基数とする FFT を採用すると 16（ロード / ストア）に対して 34 回の演算をするので，プロセッサが休まず稼働することになります。工夫その 2 はノード内の計算量の均等化です：ベクトル化はオートなので，あまり，効率化の工夫の余地がないのですが，演算によってはノード内の 8 つのプロセッサが均等に計算をしないことがあるので，マニュアルマイクロタスクと OpenMP を併用して，計算量の均等化を実現しました。工夫その 3 はノード間通信の最適化です：地球シミュレータでは MPI_put を用いることにより 64MByte を超えるデータは最大の通信速度 11.63GB/sec でノード間通信可能なので，一度に通信するデータサイズが大きくなるように工夫して，最大通信速度を確保しました。以上の主な工夫とその他の細かい工夫の組み合わせで，最終的には 2048^3 の計算を 512 ノードで行い，16.4TFlopes（ピークの 48%）を実現し，ゴードンベル賞の特別賞を獲得することができました [13]。また，4096^3 の計算は 2008 年現在未だに世界最大を維持しています。

④名古屋大学のスーパーコンピュータが VPP5000[8] から HPC2500[5]（ベクトル並列型からスカラ並列型）へ移行するにあたり，新たに HPC2500 用の高速に動作するプログラムが必要になりました。幸い，HPC2500 用ライブラリ（SSL2）のなかに，スカラ計算機向けの高速な 1 次元複素フーリエ変換（DVCFM1）[14] があったため，これを用いた高速な 3 次元 FFT を比較的容易に得ることができました。スカラ計算機を高速に動かすためには，キャッシュ上に

あるデータをできるだけ使いまわして計算することが必要ですが，DVCFM1 内ではスカラ計算機向けに 4-step algorithm をベースにした FFT を用いています。なお，VPPFortran の上位互換である XPFortran[11] では，VPPFortran[9] のときと同様に spread do でループのプロセス並列が可能ですが，そのループの内側で OpenMP を用いたスレッド並列が可能です [3]。割り当てられるプロセッサやメモリの占有状態にもよるのでしょうが，経験ではトータルなプロセッサの数が同じ場合，スレッド並列は 2 か 1（スレッド並列しない）程度が高速でした。なお，XPFortran においても spread move を用いることにより効率的にデータ転送を行うことが可能でした。

⑤地球シミュレータ用に開発した③のプログラムは MPI と Fortran を用いたベクトル並列型のスーパーコンピュータ用に最適化されていたため，名古屋大学の VPP5000 上でも，実用的に使えたのですが，さすがに HPC2500 上では④のプログラムの方が高速でした。そこで，汎用的なプログラムの必要性から，スカラ計算機用 DVCFM1 に，XPFortran ではなく MPI を組み合わせて作成したのがこのプログラムです。これにより 512^3 以下の問題サイズでは④のプログラムと同等の性能が引き出せたのですが，1024^3 以上の問題サイズでは④のプログラムのほうが高速でした。ハード（HPC2500）との相性が MPI より XPFortran の方が良いのでしょうか？（プログラムの使用頻度が低かったこともあり，この原因は追究できていません。）

Ⅵ．おわりに

以上，VP2600 から地球シミュレータや HPC2500 にいたるまで，多くのスーパーコンピュータを相手にプログラミングしてきた体験を交えた話を書いてみましたが，振り返ってみると，新しく導入されたばかりのスーパーコンピュータを用いて，世界最大規模の計算をするという類稀な体験ができたのは幸運でした。名古屋大学の情報連携基盤センター（旧：大型計算機センター）と地球シミュレータの関係者の方々には特に感謝したいと思います。

それにしても，VP2600 と比べ，地球シミュレータは約 10000 倍のスピードであるから，スーパーコンピュータの世界の進化はすごいと思います。乱流は未解決な問題として，従来手のつけようのない難問でしたが，大規模な直接数値計算を行ったことにより，多くのことが明らかになってきました。今後さらにスーパーコンピュータが発達することを考えると，自分の生きている間に乱流の主な問題はすべて解決してしまうのでは！？　という気がしてきます。10 年，20 年後にスーパーコンピュータがさらに速くなることを見越して，いまから研究テーマを暖めておくのも悪くないかもしれません。

参考文献

[1]　石井克哉：スーパーコンピュータを使おう（1），名古屋大学情報連携基盤センターニュース，Vol. 6，No. 3，pp. 263−267，2007.8

[2]　永井亨：スーパーコンピュータを使おう（2），名古屋大学情報連携基盤センターニュース，Vol. 6，No. 4，pp. 355−360，2007.11

[3] 平野靖:スーパーコンピュータを使おう（3），名古屋大学情報連携基盤センターニュース，Vol. 7，No. 1，pp. 42−50，2008.2
[4] 津田知子：スーパーコンピュータを使おう（4），名古屋大学情報連携基盤センターニュース，Vol. 7，No. 2，pp. 216−220，2008.5
[5] 津田知子：新スーパーコンピュータ（HPC2500）利用のしおり，名古屋大学情報連携基盤センターニュース，Vol. 4，No. 2，pp. 104−113，2005.5
[6] 永井亨：ベクトル計算機入門，名古屋大学大型計算機センターニュース，Vol. 23，No. 1，pp. 47−55，1992.2
[7] 長谷川明生，津田知子：新システムについて，名古屋大学大型計算機センターニュース，Vol. 26，No. 4，pp. 305−314，1995.11
[8] 永井亨：新スーパーコンピュータシステムの概要，名古屋大学大型計算機センターニュース，Vol. 30，No. 3，pp. 281−285，1999.9
[9] 永井亨：VPP Fortran 入門，名古屋大学大型計算機センターニュース，Vol. 26，No. 4，pp. 315−339，1995.11
[10] http://www.es.jamstec.go.jp/index.html
[11] 永井亨：XPFortran 入門，名古屋大学情報連携基盤センターニュース，Vol. 5，No. 2，pp. 129−168，2006.5
[12] Yokokawa M, Itakura K, Uno A, Ishihara T, Kaneda Y.: 16.4Tflops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator. Proc. ACM/IEEE Supercomput. Conf., pp. 50−50. 2002
http://www.supercomp.org/sc2002/paperpdfs/pap.pap273.pdf
[13] http://www.supercomp.org/sc2002/news_nrp_conclude.html
[14] FUJITSU SSL II 拡張機能使用手引書 II

（いしはら　たかし：名古屋大学工学部・大学院工学研究科）