

「大規模シミュレーション向け基盤ソフトウェアの開発」

西田 晃

中央大学理工学研究所／  
東京大学大学院情報理工学系研究科

## 1. 研究概要

近年、科学技術の様々な分野において、数値シミュレーションは理論・実験に代わる有力な手法として、その重要性が高まっている。マイクロからマクロに至る様々な現象を扱う上で、大規模な計算資源の利用を前提としたスケーラブルな並列数値演算ライブラリの果たす役割は大きい。しかしながら、わが国においても現在多数の科学技術アプリケーションが開発されているものの、それらの基礎となる数値演算ライブラリについては、専用のルーチンを独自に開発するか、またはベンダが提供するライブラリを利用することが多く、知的共有資産としての汎用のライブラリが開発される例は少ない。

高速ネットワーク環境の発展に伴い、今後は大規模アプリケーションについても、PC から演算サーバ、スーパーコンピュータに至る多様な資源をネットワーク上で共有し、オープンな環境で研究開発を行う機会が増えてくるものと予想される。このような環境のもとで、多重化するメモリ階層を考慮したスケーラブルな並列ライブラリの必要性は増大している。そこで本研究では、従来それぞれの分野において別個に進められてきた並列アルゴリズムや実装に関する研究をもとに、理論的な背景を考慮した標準的なソフトウェア基盤を整備することを目指し、以下の内容について研究を行なった。

まず、大規模ソフトウェア開発の方法論に関して、数値アプリケーション開発の観点から国内外の研究動向について詳細な調査研究を実施するとともに、国内における科学技術ソフトウェア開発上の問題点を解決するための方策の提案とその具体化を主要目的のひとつとした。さらにこの結果をもとに、我が国において開発されているアプリケーションの調査及び関連分野の研究者との協力により、国内の科学技術アプリケーション開発において必要と考えられる共通基盤ソフトウェアの内容とその構成の詳細を決定した。実際の研究開発においては、仕様の策定、プロトタイプ実装、評価用ハードウェアの選定、性能評価、公開及び配布など、各段階で詳細な検討が必要であり、十分な人的、物的資源を用意する必要がある。本研究では、線形系の反復解法、高速関数変換の二分野を想定しており、分野によって最適と思われる研究者への協力要請も検討するとともに、常に最新の開発環境を提供できるよう、多様なアーキテクチャから構成される計算機群を導入した。

計算機的能力は、今後も飛躍的に発展していくものと期待され、それに伴って、計算機を利用した数値シミュレーションはより広い分野で活用されるようになっていくものと予想される。本研究は、従来国外の研究成果に依存してきた大規模並列アプリケーション向けのソフトウェア基盤を国内で実現する上で、さきがけとなるものであり、関連応用分野の基盤を提供するとともに、今後の研究のモデルケースになるものと期待される。本研究の成果物は、国内外での長期間の利用に耐えるものとして設計され、またその内容についても、十分な理論的検討が行われた完成度の高いものとなっている。このような性格の研究は国内においても例が少なく、我が国における知的財産の形成に資するものと考えられる。

## 2. 研究実施内容

### 2.1 実装手法(東京大学/中央大学 西田グループ)

#### (1)研究実施内容及び成果

##### 概要

実装手法に関しては、今後普及すると思われるハードウェア技術やその利用形態を的確に予測するとともに、それらの前提の下で高い性能を発揮することのできる移植性の高いソフトウェアを設計、開発していく必要がある。このため、本研究では十分な調査研究に基づき、多様なアーキテクチャから構成される計算機群を開発環境として導入した。具体的には、開発環境として共有メモリ型並列計算機(SGI Altix 3700)、分散メモリ型並列計算機(Cray XT3)、小規模 PC クラスタ、及びベクトル計算機(NEC SX-6i)等を導入し、可搬性を備えたライブラリの開発を行なうとともに、国内外の計算機ハードウェア技術に関する詳細な調査を行った。消費電力の増大、ネットワークの高性能化に伴い、今後は計算の高並列化により、性能の向上を目指す傾向が顕著になってくるものと予想される。このような背景から、本グループでは平成16年度より IBM Watson 研究所、及びニイウス株式会社との間で Blue Gene/L を利用した数万プロセッサレベルでの高並列な環境下での数値ライブラリの実装技術について研究を開始し、開発アプリケーションの移植、性能評価を実施した。また平成18年度からは、地球シミュレータ共同プロジェクトの一環として、ベクトルプロセッサを用いた大規模並列計算環境への対応を行なった。

ライブラリの設計手法に関しては、国内外での調査結果も踏まえ、可搬性を備えるとともに利用者が効率的に処理を記述できるよう、オブジェクト指向型のインタフェースを採用することとし、ライブラリの記述手法、記述言語の双方について研究を進めた。この成果は各分野で反映されており、反復解法ライブラリについては完全なオブジェクト指向型のインタフェースを実現するとともに、高速関数変換においては性能の自動最適化機能を備えた FFT ライブラリを実装した。いずれも C 言語で記述するとともに Fortran インタフェースを備えており、必要に応じてより高級な言語によるインタフェースを付加するとともに、自動最適化機能についても充実させていくことが可能な仕様となっている。

また、より抽象度の高いインタフェースとして、数値ライブラリの利用を容易にするためのスクリプト言語 SILC (Simple Interface for Library Collections)を開発し、2件の特許出願を行なった。

##### SILC (Simple Interface for Library Collections)

行列計算ライブラリは科学技術計算プログラムの作成に不可欠な構成要素であり、その重要性から多くのライブラリが開発されている。これらのライブラリは一般に、個々のライブラリが提供するインタフェース(API)を通じて利用される。例えば、代表的な行列計算の1つである連立一次方程式 $Ax = b$ の求解の場合、ユーザはまず行列Aとベクトルbをライブラリ固有のデータ構造で準備する。次に、特定のライブラリ関数(求解ルーチン)を所定の引数の順序に従って呼び出す。このようなライブラリ固有のデータ構造と関数呼び出しに基づく従来のライブラリ利用法では、作成したユーザプログラムが特定のライブラリに依存する。ライブラリ間のインタフェースには互換性がないことが多いため、他のライブラリが提供する求解ルーチンを利用するにはユーザプログラムの修正が必要である。別の前処理や演算精度を利用する場合も同様である。また、特定の計算環境でのみ利用可能なライブラリが存在するため、ユーザプログラムを他の環境に移す際にライブラリの変更とそれに伴うユーザプログラムの修正が必要になる。しかしながら、ライブラリや計算環境を変更する都度ユーザプログラムを修正することは負担であり、より柔軟なライブラリ利用法が求められている。この要求に応えるために、我々は計算環境に依存しない行列計算ライブラリインタフェースSILC (Simple Interface for Library Collections)を提案した。ライブラリ固有のインタフェースに基づく従来のライブラリ利用法とは異なり、SILCでは(1)入力データの預け入れ、(2)文字列(数式)による計算指示、(3)計算結果の受取りの3つのステップで行列計算ライブラリの機能を利用する。行列やベクトルなどの入力データは、まずユーザプログラムから独立したメモリ空間に転送される。また、数式の形で与えられた計算指示は、適当なライブラリ関数の

呼び出しに翻訳されて、独立したメモリ空間内で実行される。最後に、ユーザプログラムからの要求に応じて計算結果がユーザプログラムのメモリ空間に戻される。例として、SILC のインタフェースを介して行列計算ライブラリLAPACK の連立一次方程式の求解ルーチン呼び出すC プログラムを以下に示す。

```

silc_envelope_t A, b, x;
/* 行列 A とベクトル b の作成*/
SILC_PUT("A", &A);
SILC_PUT("b", &b);
SILC_EXEC("x = A ¥¥ b"); /* 求解*/
SILC_GET(&x, "x");

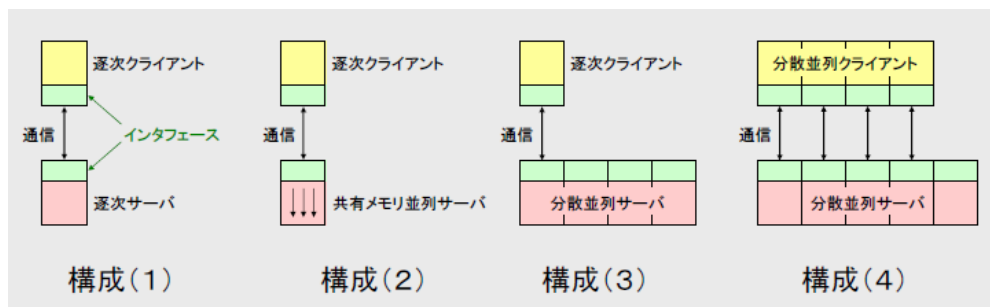
```

このプログラムは、LAPACK のデータ構造を用いて行列A とベクトルb を作成した後、SILC が提供するSILC\_PUT, SILC\_EXEC, SILC\_GET の3つのルーチンを介してLAPACK の求解ルーチン呼び出す。

科学技術計算に用いられる種々の並列環境では、OpenMP やMPI を採用したさまざまな並列行列計算ライブラリが使われている。SILC は、ユーザプログラムと計算環境の間に位置して計算環境の違いを吸収するシステムであり、これらのライブラリをプログラミング言語や計算環境に依らない方法で手軽に利用できるようにする。並列環境については、以下の4種のシステム構成を想定した。

- (1) 逐次クライアント+逐次サーバ
- (2) 逐次クライアント+共有メモリ並列サーバ
- (3) 逐次クライアント+分散並列サーバ
- (4) 分散並列クライアント+分散並列サーバ

ライブラリを利用するユーザプログラムの構成を図に示す。



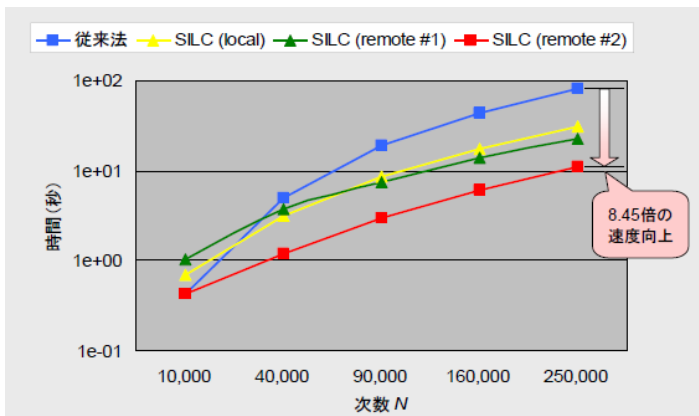
ライブラリはユーザプログラムにリンクされて用いられる。ユーザプログラムはSILC サーバを構成する1つもしくは複数のプロセスに接続してデータ転送と計算指示を行なう。ユーザプログラムからサーバに転送されたデータは、サーバが備えるデータ再分散機構によりライブラリの要求するデータ分散方式に変換され、複数のサーバプロセスに分散して保持される。また、ユーザプログラムに返される計算結果は、データ再分散機構により転送前にユーザプログラムの要求するデータ分散方式に戻される。

実現したシステムを通じて遠隔の高速な分散並列環境を利用することにより、従来法で書かれたユーザプログラムと比較して性能向上が得られることを確かめた。以下に、複数の計算環境の組み合わせについて、2次元拡散方程式の初期値問題を差分法で解いた結果を示す。連立一次方程式の求解には反復解法ライブラリLis の前処理なし共役勾配法を用い、サーバ上で連立一次方程式の求解演算子をLis のMPI 版の呼び出しに翻訳するように設定した。次数をN, 反復回数を $\alpha$ とすると、通信量は $O(N)$ , 計算量は $O(\alpha N)$  であるため、反復回数のかかる問題であれば通信コストを加味しても遠

隔の高速な並列計算機を利用した方が速いことが分かる.

	ユーザプログラム	SILC サーバ
従来法	Xeon4 (1 PE)	—
SILC (local)	Xeon4 (1 PE)	Xeon4 (4 PEs)
SILC (remote #1)	Xeon4 (1 PE)	Xeon8 (8 PEs)
SILC (remote #2)	Xeon4 (1 PE)	Altix (16 PEs)

ホスト名	仕様
Xeon4	IBM eServer xSeries 335 (dual Intel Xeon 2.8 GHz, 1.0 GB RAM) × 4, Red Hat Linux 8.0, LAM/MPI 7.0
Xeon8	Xeon4 と同じ PC クラスターの別の8ノード*
Altix	Intel Itanium2 1.3 GHz × 32基, メモリ 32 GB, Red Hat Linux Advanced Server 2.1, SGI MPI 4.4 (MPT 1.9.1)



### スクリプト化

SILC において, 条件分岐や反復などの制御構文を処理する場合は, ユーザプログラムを作成するユーザがあらかじめ上記制御を含む処理が記述された関数を定義し, ライブラリプログラムシステムに渡しておく必要がある. このため, 制御構文を含む任意の計算処理をユーザプログラムにおいて記述することができない問題があった. そこで本研究では, SILCの命令記述言語を拡張し, 制御構文を加えたスクリプト言語を定義して, これを用いて書かれたユーザプログラムを解析し, 演算処理と制御処理を分離して従来のSILC の枠組みで処理することにより, スクリプト化を実現した. これにより, 以下の共役勾配法の例に示すような平易な記述を対話的に行なうことが可能となった.

```
# 三重対角行列 A とベクトル b を作る
n = 400
A = diag(2.0 * ones(n, 1)) - diag(ones(n-1, 1), 1) - diag(ones(n-1, 1), -1)
b = A * (-ones(n, 1))

# 連立一次方程式 Ax=b を CG 法で解く
rho_old = 1.0
p = zeros(n, 1)
x = zeros(n, 1)
r = b
bnrm2 = 1.0 / norm2(b)
iter = 1
```

```

while (iter <= n) {
  rho = r' * r
  beta = rho / rho_old
  p = r + beta * p
  q = A * p
  alpha = rho / (p' * q)
  r = r - alpha * q
  nrm2 = norm2(r) * bnorm2
  x = x + alpha * p
  if (nrm2 <= 1.0e-12) {
    break
  }
  rho_old = rho
  iter += 1
}

# 解 x をファイルに保存する
save "sol.mtx", x

```

### 関連研究

Trilinos プロジェクトでは、行列計算ライブラリを C++ のクラスライブラリとして統合するための枠組みを提案し、多数の行列計算ライブラリを開発している。各ライブラリの API は、(1)共通の行列とベクトルのデータ構造を用いること、(2)ライブラリの開発言語として C++ を採用し、解法のインタフェースを規定する共通の抽象クラスを継承することの2点において統一されている。また、(3)共通のディレクトリ構造とコンパイル方法を採用し、開発者の異なるライブラリを同じ方法で構築可能な Trilinos パッケージとしてまとめている。ただし、API の詳細はライブラリ毎に大きく異なる。例えば Trilinos に含まれる連立一次方程式の直接解法パッケージと反復解法パッケージの API には互換性がないため、一方から他方に切り替えるにはユーザプログラムの修正が必要である。一方、SILC では計算を数式で指示するので、互換性のないライブラリであっても同じ方法で利用できる。

遠隔の計算環境を利用するという点で分散型 SILC と関連の深いシステムに Ninf-G がある。

Ninf-G はグリッド環境において遠隔手続き呼び出し(RPC)を実現するためのミドルウェアであり、MPI ベースの行列計算ライブラリの呼び出しをサポートしている。ただし、RPC を実行するプロセスは1つ(逐次プログラム、または分散並列プログラムのプロセスの1つ)でなければならない。また、入力データ(ライブラリ関数の引数)はライブラリ側のプロセスの1つにのみ転送される。他のプロセスへの入力データの分散と計算結果の収集についてはユーザが IDL ファイル内に記述しなければならない。一方、SILC では複数プロセスからのライブラリ呼び出しをサポートしている。また、SILC がサポートする格納形式を入出力データの授受に使う場合はユーザがデータ分散とデータ収集の処理を記述する必要はない。

スクリプト化に関する先行研究としては、Matlab を挙げるができる。Matlab はライブラリプログラムを簡便に実行するためのスクリプト言語を提供するものであるが、演算が逐次処理に限られる問題点があった。本研究では設計の段階から並列処理を想定しており、スクリプトにおいても容易に並列計算を行なうことができる。なお、Matlab の並列化についても、MIT, Interactive Supercomputing 社による Star-P など、複数の実装が提案されており、活発な研究開発が行なわれている。

### (2)研究成果の今後期待される効果

本グループのこれまでの成果として、計算機科学分野で得られたハードウェア技術、ソフトウェア技術に関する知見を迅速に科学技術計算に応用するためのソフトウェア実装手法を確立したこと、今後予想される高並列な計算環境での使用に耐える基盤ソフトウェア群を実際に開発、早期に無償で公開

し、ユーザからのフィードバックをもとにソフトウェアを改良、発展させていくサイクルを確立したこと、などを挙げることができる。

## 2.2 反復解法(東京大学/中央大学 西田グループ)

### (1)研究実施内容及び成果

反復解法研究グループでは、多数の公開シンポジウムを開催するとともに、各メンバーとも精力的に研究を進め、共役勾配法、共役残差法系の線形反復解法、固有値解法とその前処理手法について、多くの国際・国内会議で成果を発表した。また、これらの解法を実装したオブジェクト指向型の並列反復解法ライブラリ Lis (A Library of Iterative Solvers for Linear Systems)を公開し、ユーザの要望等を取り入れつつ、版を重ねている。今後もアルゴリズムの充実を図るとともに、より大規模な計算機環境での利用実績を増やしていきたいと考えている。

現在までの理論的な研究成果については、以下を挙げることができる。

### 固有値解法

物性物理や量子化学などの物質科学分野では、密度汎関数法や第一原理計算を用いた大規模なシミュレーションが必要となることが多い。これらの計算では、高速な固有値解法に対する需要は大きく、両分野の研究者が密接に協力することにより、実用性の高い汎用ソフトウェアを開発することができるものと期待される。このような計算では、大規模問題において少数の固有値・固有ベクトルを効率的に求める必要があり、実際の応用に耐え得る高速かつスケーラブルな固有値計算アルゴリズムの設計が求められている。大規模疎行列の固有値を数値的に求める方法としては、いくつかの解法を考えることができ、Lanczos 法やその非対称問題への拡張である Arnoldi 法、あるいは量子化学計算で利用されることの多い Davidson 法や、その一種である Jacobi-Davidson 法などの反復解法を挙げることができる。本グループでは、一般化固有値問題

$$Ax = \lambda Bx$$

の最小固有値、あるいはこれと同値な問題

$$Bx = \mu Ax, \mu = 1 / \lambda$$

の最大固有値を求める問題が Rayleigh 商

$$\mu(x) = x^T Bx / x^T Ax$$

の極値問題に帰着して共役勾配法により解けることに着目し、代数的マルチグリッド法などの適切な前処理と組み合わせることにより、Lanczos 法系の解法と比較してより高速に固有値を計算できることを示した。

### 連立一次方程式解法

流体シミュレーションや構造解析など、偏微分方程式の数値解が必要となる分野においては、離散化の過程で大規模な連立一次方程式を扱うことから、効率的なアルゴリズムに対する需要は大きい。本研究では、これらの応用分野から派生する連立一次方程式の反復解法及びその前処理手法を中心に、極めて高並列な環境での使用にも耐え得るスケーラブルかつ信頼性の高いアルゴリズムの設計及び実装を行ってきた。

近年、大規模連立一次方程式の反復解法として、マルチレベルな解法が多く研究されており、そのような手法のひとつとして、代数的マルチグリッド (AMG) 法を挙げることができる。一般にマルチグリッド法は

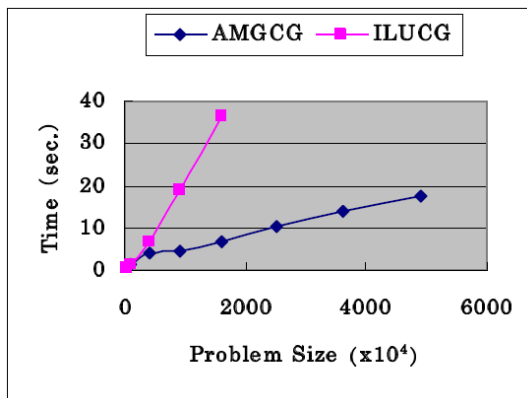
- ・ 並列性に富む
- ・ 逐次計算の場合、問題行列の大きさを  $n \times n$  とすると、うまく機能した場合収束までの計算量は  $O(n)$  である

といった性質を持つ。AMG 法は幾何的マルチグリッド法と同様の原理にもとづいているが、問題行列

のみを利用して誤差の高周波数成分の取り除かれた粗いレベルの行列を生成し、幾何的マルチグリッド法と同等の計算量で高速に問題を解くことができ、

- ・ 不規則な疎行列に対しても適用可能である
- ・ 異方性問題に対しても有効である

といった利点を持つ。我々は、AMG 法のうち最も有力な解法のひとつである smoothed aggregation に基づく algebraic multigrid 法を前処理として用いた共役勾配法 (SA-AMGCG 法) の効率的な並列実装手法を平成14年より提案するとともに、ICCG 法 (Localized ILU 前処理付共役勾配法) との比較から、大規模な問題になるほどより高速に解けることを示した。AMG 法は汎用の前処理手法としては現時点で最も有効なアルゴリズムであり、スケーラビリティについても優れた特性を持つ。本グループではこれを反復解法ライブラリ Lis 上に実装し、高並列環境上でその性能を検証した。以下に平成18年12月に開催された Juelich 総合研究機構でのワークショップに際して実施した IBM Blue Gene 上の 1024ノードでの実行結果を示す。



共役勾配法などのクリロフ部分空間法の収束性は丸め誤差に大きく影響されるが、高精度演算を用いた場合、計算時間が多くかかってしまう。我々は、倍精度浮動小数を2個用いた double-double 型 4倍精度演算の SIMD 命令を用いた高速化手法を提案し、Fortran コンパイラを用いた完全4倍精度演算に比べて約5.8倍の性能向上を得た。また、これを用いた倍精度と4倍精度の混合型反復法を提案し、4倍精度のみの計算と比較して最良で2.8倍の性能向上を得た。

また、理論的成果として、対称行列用の共役残差 (CR) 法を非対称行列用へ拡張した解法である双共役残差 (BiCR) 法の提案を挙げることができる。これについては、双共役勾配 (BiCG) 法よりも多くの場合に滑らかな収束性を示すことを示すとともに、積型反復解法にこれを応用した GPBiCR 法を提案した。

これらの解法はいずれも Lis 上に実装されており、ユーザが利用できるようになっている。

## (2)研究成果の今後期待される効果

大規模科学技術シミュレーションにおいて大きな比重を占める差分法や有限要素法をはじめ、反復解法の適用分野は大変広い。反復解法ライブラリ Lis は、計算機科学、計算科学両分野の知見を反映したソフトウェア基盤として、十分な波及効果が期待できる。現時点においても国内外の様々な分野の研究者から多数のフィードバックを得ており、新たな機能を追加していくことにより、さらなる展開が期待できるものと考えている。

## 2.3 高速関数変換(東京大学 須田グループ)

### (1)研究実施内容及び成果

気象予測や地球流体解析など、球面上の流体方程式を扱う必要のある分野では、スペクトル法などの高速積分変換を用いた高精度な解法が利用されている。本研究では、これらの分野の研究者との協力により、アルゴリズムレベルにおいて並列度の高い解法の研究及び設計を行うとともに、高性能な並列高速フーリエ変換ライブラリを中心に、実際の計算環境において高い実効性能を得ることのできる高速関数変換ソフトウェアの開発を行なった。

#### 高速フーリエ変換

離散フーリエ変換とその高速アルゴリズムである高速フーリエ変換 (FFT) は、大規模な科学技術計算からマルチメディア関連の圧縮／伸長まで非常に多くの分野で用いられている。FFT のアルゴリズムが発見されてから多くの改良が行なわれているが、プロセッサアーキテクチャの急激な進歩に伴い、それに適した新しいFFT カーネルの開発が必要となっている。

キャッシュを有効利用する従来のアルゴリズムにおいては、キャッシュ内で行われているFFT アルゴリズムにStockham FFT 等bit-reverse 処理を要しないアルゴリズムが用いられることが多いが、本研究では、このbit-reverse 処理を主記憶アクセス時間に隠蔽することで、in-place アルゴリズムを用いることができるような手法を提案し、実際にはほぼ隠蔽可能であることを確認した。また、Intel IA-64, IBM POWER などのように、科学技術計算において使用されることの多いアーキテクチャでは、積和演算ユニットを2つ持ち、1サイクルあたり4つの浮動小数演算が実行できるプロセッサが増加する傾向にある。積和演算命令は乗算結果に加減算を行う複合命令であり、積和演算命令をサポートするプロセッサの場合は乗算だけ、または加減算だけを行いたい場合にも積和演算ユニットを使用する。このため、積和演算ユニットを効率よく使用するためには、なるべく乗算と加減算を組み合わせる積和演算命令として実行することが不可欠である。本研究では積和演算命令数が最小となる8 基底FFT カーネルにおいて、必要なひねり係数のテーブルが小さく、ロードする必要があるひねり係数の数も少ないという利点を持つ新しい8 基底FFT カーネルを提案した。これらの成果は開発ライブラリに反映され、スーパースカラ型のプロセッサを対象に、性能を重視し、かつ自動最適化機能を備えた FFT ライブラリである FFTSS を平成17年9月より公開した。また、気象予測や地球流体解析など、球面上の流体方程式を扱う必要のある分野では、大量のデータを用いた大規模な FFT 計算が必要となることが多い。本研究では、平成18年度より地球シミュレータ共同プロジェクトの一環として、ベクトル計算機への対応を行ない、倍精度2次元 FFT において 16TFLOPS を超える性能が得られることを実機上で確認した。

#### 高速多重極子展開法

気象予報・気候予測などの分野では、球面上の球座標において直交関数系となる球面調和関数を用いることが多く、緯度・経度格子上の関数値と球面調和関数展開の係数との間の変換である球面調和関数変換の高速化は重要な意味を持っている。球面調和関数は東西方向の三角関数と南北方向のルジャンドル陪関数との積に分解できることから、前者はFFTで処理することができる。本研究では、後者を高速に解くための一般化高速多重極子展開法を提案し、 $O(N^2 \log N)$  での安定な計算が可能になることを示した。

#### (2)研究成果の今後期待される効果

本グループで開発したライブラリは、ベンダにより提供されているものと比較しても多くの場合に優れた性能を示しており、科学技術分野や産業界に対する波及効果は大きい。現在アーキテクチャに応じて個別に作成されている複数の FFT カーネルの統合を進め、ライブラリとしての完成度を高めていくことで、実アプリケーションでの利用がさらに進むものと期待される。



### 3. 研究実施体制

#### 実装手法研究グループ

- ① 研究分担グループ長：西田 晃（東京大学大学院情報理工学系研究科 リサーチフェロー / 中央大学理工学研究所 客員研究員）
- ② 研究項目：実装手法に関する研究及びソフトウェアの開発を担当

#### 反復解法研究グループ

- ① 研究分担グループ長：西田 晃（東京大学大学院情報理工学系研究科 リサーチフェロー / 中央大学理工学研究所 客員研究員）
- ② 研究項目：並列反復解法に関する研究及びソフトウェアの開発を担当

#### 反復解法研究グループ

- ① 研究分担グループ長：須田 礼仁（東京大学大学院情報理工学系研究科コンピュータ科学専攻 准教授）
- ② 研究項目：高速関数変換手法に関する研究及びソフトウェアの開発を担当

### 4. 主な研究成果

#### (1) 論文発表

A. Nukada, D. Takahashi, R. Suda, and A. Nishida. High Performance FFT on SGI Altix 3700, In *Proceedings of the Third International Conference on High Performance Computing and Communications (HPCC 2007)*, Lecture Notes in Computer Science 4782, pp. 396-407, Springer, 2007.

A. Nukada, Y. Hourai, A. Nishida and Y. Akiyama. High Performance 3D Convolution for Protein Docking on IBM Blue Gene, In *Proceedings of the Fifth International Symposium on Parallel and Distributed Processing and Applications (ISPA07)*, Lecture Notes in Computer Science 4742, pp. 958-969, Springer, 2007.

額田彰, 西田晃, 「地球シミュレータを用いた高性能2次元 FFT」, 2007年先進的計算基盤システムシンポジウム論文集, pp.137-144, 情報処理学会, 2007.

小武守恒, 藤井昭宏, 長谷川秀彦, 西田晃, 「倍精度と4倍精度の混合型反復法の提案」, 2007年ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集, pp. 9-16, 情報処理学会, 2007.

A. Nishida. Building Cost Effective High Performance Computing Environment via PCI Express. In *Proceedings of the 2006 International Conference on Parallel Processing Workshops (ICPPW2006)*, pp. 519-526, IEEE, 2006.

T. Kajiyama, A. Nukada, H. Hasegawa, R. Suda, and A. Nishida, SILC: A Flexible and Environment Independent Interface for Matrix Computation Libraries, In *Proceedings of the Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM2005)*, Lecture Notes in Computer Science 3911, pp. 928-935, Springer, 2006.

A. Nukada. FFTSS: a High Performance Fast Fourier Transform Library, In Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2006), Vol. III, pp.980-983, IEEE, May 14-19, 2006, Toulouse.

A. Fujii, A. Nishida, and Y. Oyanagi, Evaluation of Parallel Aggregate Creation Orders : Smoothed Aggregation Algebraic Multigrid Method, In IFIP International Federation for Information Processing, Vol. 172, pp.99-122, Springer, 2005.

H. Kotakemori, H. Hasegawa, T. Kajiyama, A. Nukada, R. Suda, and A. Nishida, Performance Evaluation of Parallel Sparse Matrix-Vector Products on SGI Altix3700, In Proceedings of the First International Workshop on OpenMP (IWOMP2005), Lecture Notes in Computer Science, Springer, in press.

(2) 口頭発表

A. Nishida, H. Kotakemori, A. Fujii, and A. Nukada, Development of Scalable Software Infrastructure on Blue Gene Systems, In Report on the Juelich Blue Gene/L Scaling Workshop 2006 (FZJ-ZAM-IB-2007-02), John von Neumann Institute for Computing, Research Centre Juelich, February, 2007.

西田晃, 「SSI: 大規模シミュレーション向け基盤ソフトウェアの概要」, IPSJ SIG Notes, 2004(38), pp.25-30, 2004.

A. Nishida. SSI: Development of Simulation Software Infrastructure. Eleventh SIAM Conference on Parallel Processing for Scientific Computing, February 25-27, 2004, San Francisco.