

高速4倍精度演算パッケージ ASLQUADの概要

緒方 隆盛・久保 克維・武井 利文

要 旨

高速4倍精度演算パッケージASLQUADは、従来の倍精度数値シミュレーションでの数値誤差を軽減し、より信頼性の高い計算を可能にします。ASLQUADでは、NEC製スーパーコンピュータSXシリーズの高い演算処理能力を最大限に引き出しつつ、ユーザプログラムの移植性の向上や機能の豊富さも追求しています。本稿では、ASLQUADの特長や機能について概説し、固有値計算を例に高精度演算の有用性について述べます。

キーワード

●4倍精度演算 ●多倍長演算 ●高精度演算 ●科学技術計算ライブラリ ●数値誤差

1. はじめに

スーパーコンピュータを利用する上で直面することが想定される問題の1つとして、数値丸め誤差の蓄積による精度喪失が挙げられます。例えば、数値誤差に敏感な収束計算や計算結果の精度評価では、数値誤差をいかに軽減するかが問題となります。従来の大規模シミュレーション開発では、これらの問題を克服するために、使用している数値計算アルゴリズムの精度改善だけでなく、ハードウェアやコンパイラでサポートしている4倍精度演算が利用されてきました。

こうした流れの中で、高速な高精度計算に対する期待が高まっています。しかし、ソフトウェア的な制御を行わずに、与えられたハードウェアやコンパイラをそのまま使っただけでは、高精度計算の性能を左右するデータの内部表現形式、演算方式、保証する精度などを柔軟に選択することができず、また、きめ細かな性能最適化を十分に行えないこともあるため、高速かつ高精度な演算に対する期待に十分応えることは困難です。

NECでは、ソフトウェアの観点から高速かつ高精度な演算

をサポートするために、高速4倍精度演算パッケージASLQUADの開発に取り組んでいます。本稿では、スーパーコンピュータSXシリーズで利用可能なASLQUADの特長と機能を概説し、ASLQUADを使った固有値計算を例に高精度演算の有用性について述べます。

2. ASLQUADの特長

ASLQUADは、高い精度が要求される数値シミュレーションプログラムの作成を強力に支援する開発ツールです。ASLQUADの利用によって、多倍長計算のアルゴリズムの詳細に煩わされることなく、高速かつ高精度な数値シミュレーションプログラムを作成することができ、プログラム開発の生産性が大幅に改善されます。

ASLQUADは、4倍精度の基本演算機能群、数値計算機能を集めたサブルーチンライブラリ機能群、プログラミング支援のための補助機能群で構成されます（図1）。これらの機能群は、Fortran90で書かれたユーザプログラムへの組み込みが容易にできるよう設計されており、ノード内で並列化されたプ

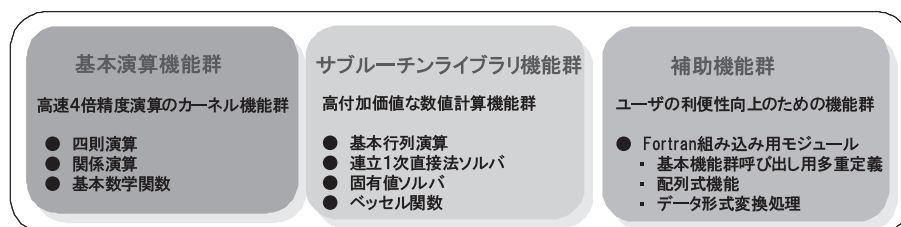


図1 高速4倍精度演算パッケージASLQUADの機能群

プログラムだけでなく、MPIプログラムからの利用も可能です。

SXシリーズのベクトル演算レジスタは、64ビット幅となっています。このベクトル演算レジスタを使って高速な4倍精度演算を実現するため、ASLQUAD内部の4倍精度データ形式には、多倍長計算分野で知られているdouble-double形式を採用しています¹⁻³⁾。double-double形式は、4倍精度データ（128ビット）を2つの倍精度データ（64ビット）で表すデータ形式です。図2にdouble-double形式のデータフォーマットを示します。

double-double形式を使った演算は、1回の4倍精度演算を複数回の倍精度演算でエミュレートすることによって、実現されます。例として、double-double演算の加算(C=A+B)を以下に示します。

```
subroutine A_add_B (Ahi, Alo, Bhi, Blo, Chi, Clo)
real(kind=8) Ahi, Alo, Bhi, Blo, Chi, Clo, z, r, zz
z = Ahi + Bhi
r = Ahi - z
zz = ((r + Bhi) + (Ahi - (r + z))) + Alo + Blo
Chi = z + zz
Clo = (z - Chi) + zz
end subroutine
```

ここで、(Ahi, Bhi, Chi)は、それぞれ(A, B, C)の上位精度部分を表し、(Alo, Blo, Clo)は下位精度部分を表します。その他の四則演算や数学関数についても同様に、1つの4倍精度演算を複数の倍精度演算でエミュレートします。

この手法を応用した多倍長計算ソフトウェアは、Web上でも入手可能です^{4) 5)}。しかし、それらには、double-double演算のための関数呼び出しがコンパイラによる最適化を妨げてしまう欠点があります。そのため、従来の多倍長計算ソフトウェアでは、SXシリーズの強力なベクトルパイプライン処理を活かせません。この欠点を克服するため、ASLQUADは、

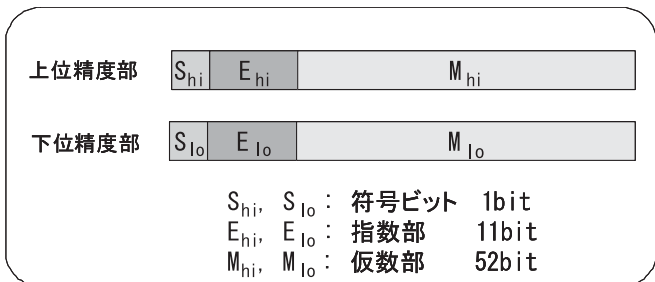


図2 double-double型のデータ形式

FORTRAN90/SXの多段インライン展開機能を利用して関数呼び出しを大幅に削減できるように設計されています。また、double-double演算の四則演算とそれらを組み合わせた初等数学関数では、演算回数や条件分岐の簡略化などの最適化を図っています。

2.1 基本演算機能群

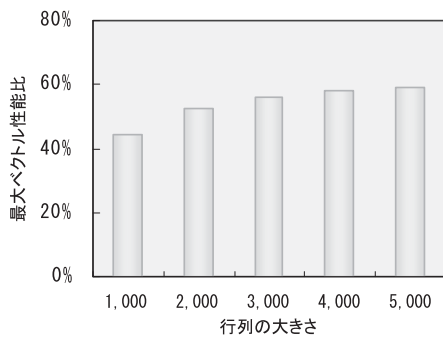
基本演算機能群は、一般的な商用コンパイラでサポートしている4倍精度の四則演算や組み込み関数に相当する機能です。最新版R2.0の基本演算機能群では、double-double形式の演算を使った以下の機能をサポートしています。

- 1) 四則演算/べき乗算 (+, -, *, /, **)
- 2) 関係演算 (>, <, ==, <=, >=, /=)
- 3) 型変換 (INT, DBLE, QEXT)
- 4) 総和/内積 (SUM, DOT_PRODUCT)
- 5) 絶対値/最大値/最小値 (ABS/MAX/MIN)
- 6) 平方根/逆平方根/立方根 (SQRT, RSQRT, CBRT)
- 7) 三角関数 (SIN, COS, TAN)
- 8) 逆三角関数 (ASIN, ACOS, ATAN)
- 9) 指数関数 (EXP, EXP2, EXP10)
- 10) 対数関数 (LOG, LOG2, LOG10)
- 11) 双曲線関数 (SINH, COSH, TANH)

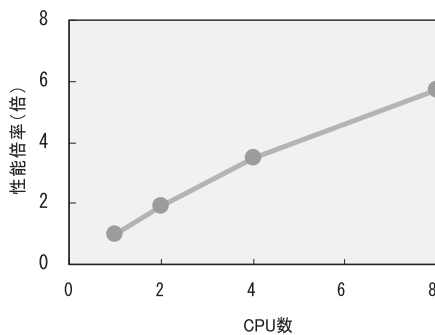
2.2 サブルーチンライブラリ機能群

サブルーチンライブラリ機能群は、基本演算機能群を組み合わせ合わせた高付加価値な数値計算機能群です。各機能は、様々な数値計算手法の中から数値誤差の影響を受けやすいものやそのベースとなるものを厳選し、ベクトル化、並列化を意識して、高度に最適化しています。最新版R2.0では、以下の機能をサポートしています。

- (1) 基本行列演算
 - ・行列積
- (2) 連立1次直接法ソルバ (密行列用)
 - ・LU分解/LU分解後の求解
 - ・行列式/逆行列計算
- (3) 実対称固有値ソルバ/一般化固有値ソルバ
 - ・全固有値/全固有ベクトル計算
 - ・区間指定ありの固有値/固有ベクトル計算



(a) 最大ベクトル性能比(1CPUあたり)



(b) ノード内並列性能倍率(行列の大きさ5,000×5,000)

図3 ASLQUADの実対称一般化固有値ソルバの性能

(4) ベッセル関数

- ・第1種/第2種整数次ベッセル関数

図3は、ASLQUADの一般化固有値ソルバ(実対称密行列用)をSX-8上で利用したときの(a)最大ベクトル性能比¹と(b)CPU数に対するノード内並列化性能倍率²を表すグラフです。1CPUでの最大ベクトル性能比は、おおむね50%まで引き出されており、並列化性能倍率は8CPUで約6倍を実現しています。

図3はSX-8上での測定結果ですが、現在、SX-9については最適化中です。SX-9でも、SX-8と同等の最大ベクトル性能比と並列化性能倍率を見込んでいます。

2.3 補助機能群

補助機能群は、データの互換性やプログラムの移植性を補うための機能群です。この機能群は、基本演算機能群を容易に利用するためのFortranモジュールとして提供しています。Fortranモジュールの内部では、double-double型データを構造体で定義しており、Fortran90の利用者定義演算子(及び利用者定義関数)を使って、基本演算機能群の各機能の呼び出しを多重定義しています。基本演算機能群でサポートしている機能であれば、Fortran90の配列式構文として利用することも可能です。また、Fortranモジュール内部では、SXシリーズでサポートしている4倍精度データ形式とdouble-double形式との間で形式変換を行う機能も提供しています。

これらの機能によって、ユーザはFortranモジュールをプログラムに引用するだけで、Fortran90の文法に近いコーディングで基本演算機能群の利用が可能になります。以下に、ASLQUADを使ったFortran90プログラムのイメージを示します。“TYPE(quad)”の行は、モジュール内で定義されたdouble-double型を表す構造体の配列宣言を意味します。

```
USE ASLQUAD                                !モジュール
TYPE(quad), dimension(256,256) :: A, B, C
~
A=1q0; B=2q0; C=3q0                        !配列式
do K = 1, 256
  do J = 1, 256
    do I = 1, 256
      C(I, J)=C(I, J)+A(I, K)*B(K, J)
    enddo; enddo; enddo
~
END
```

3. ASLQUADを利用した固有値計算の精度

高精度演算が必要になる一例として、大規模固有値計算について紹介します。従来、固有値計算は、構造解析や振動解

¹ 最大ベクトル性能比: double-double演算内部の倍精度浮動小数点演算性能をベクトルユニット内の加算演算器と乗算演算器のみによる浮動小数点演算処理能力(理論ピーク値)で割ったもの。

² 性能倍率: 全体の処理時間を1CPUで実行した時の処理時間で割ったもの。

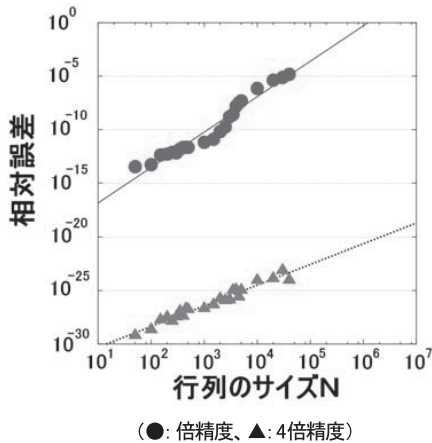


図4 Frank行列の固有値の相対誤差

析など、様々な研究分野で必要とされる計算ですが、解くべき行列の性質によっては、倍精度では精度の良い解が得られない場合があります。図4に、Frank行列と呼ばれる実対称行列の固有値について、倍精度と4倍精度で計算した場合の相対誤差を示します。ここで、Frank行列は、各成分 A_{ij} が

$$A_{ij} = N - \max(i, j) + 1$$

で定義される行列であり、固有値の厳密解 E_i は、

$$E_i = \frac{1}{\sqrt{1 - \cos \frac{(2i-1)\pi}{2N+1}}}$$

で表されることが知られています⁶⁾。また、4倍精度の固有値計算はASLQUADの固有値ソルバ(ASLQ_DCSMAN)を使い、倍精度についてはNEC製科学技術計算ライブラリASLの固有値ソルバ(DCSMAN)を使いました。なお、各精度での固有値の計算にはHouseholder変換と無平方根QR法を用いました。

行列サイズNが100,000を超えると、倍精度演算での相対誤差は、 $O(10^{-5})$ 以下になりました。すなわち、行列サイズNが100,000を超える行列の固有値を、有効桁数5桁以上で求めたい場合は、倍精度より高い精度(例えば、4倍精度)で計算しなければならないことになります。また、この倍精度の計算結果を用いて計算を続けた場合、数値誤差がさらに蓄積され、プログラム全体としての計算精度はさらに劣化する可能性があります。この結果は、Frank行列についての結果ですが、Frank行列のみならず比較的悪条件の大規模密行列について精

度良く固有値を求めたい場合は、4倍精度演算が有効です。

4. むすび

本稿では、高速4倍精度演算パッケージASLQUADの特長と機能の概要を述べました。また、大規模固有値計算の例では相対誤差の比較から4倍精度演算の有用性を述べました。本パッケージの長所は、double-double型の4倍精度演算をFORTRAN90/SXの多段インライン展開機能を用いて最適化を行い、NEC製スーパーコンピュータSXシリーズ向けに高度な最適化を図っていることです。

今後も、SXシリーズのハードウェア性能を高精度演算にも生かし、同シリーズの利用価値を高めるため、高精度演算の機能拡充、サブルーチンライブラリの性能チューニング、8倍精度演算サポートなどを行っていく予定です。

参考文献

- 1) Dekker, T.J.: A floating-point technique for extending the available precision, Numerische Mathematik, Vol.18, pp.224-242, 1971
- 2) Knuth, D. E.: The art of computer programming Vol.2 Seminumerical algorithms, 3rd Edition, Addison- Wesley, 1998
- 3) 小武守恒、藤井昭宏、長谷川秀彦、西田 晃：SSE2を用いた反復解法ライブラリLis 4倍精度演算版の高速化、情報処理学会研究報告, 2006-HPC-108, pp.7-12, 2006
- 4) DDFUN90 ; <http://crd.lbl.gov/~dhbailey/mpdist/index.html>
- 5) QD ; <http://crd.lbl.gov/~dhbailey/mpdist/index.html>
- 6) Frank, Werner L. "Computing Eigenvalues of Computing Eigenvalues of Complex Matrices by Determinant Evaluation and by Methods of Danilewski and Wielandt," Jour. SIAM, 6, 378-392, 1961

執筆者プロフィール

緒方 隆盛
第一コンピュータ事業本部
HPC販売推進本部

久保 克維
第一コンピュータ事業本部
HPC販売推進本部
HPCソリューションマネージャー

武井 利文
第一コンピュータ事業本部
HPC販売推進本部
グループマネージャー