# MBLAS と MLAPACK; 多倍長精度版の BLAS/LAPACK の作成

**中田真秀**

maho@riken.jp

**理化学研究所情報基盤センター**

February 23, 2009

# 大枠

- 自己紹介
- MPACK(MBLAS/MLAPACK) の紹介
- ライブラリ作成の動機
- 現在の状況
- 今後の予定

# 自己紹介

- 中田真秀 (なかたまほ)
- 理化学研究所 情報基盤センター
- 基礎科学特別研究員 (2007 より)
- 量子化学、理論化学
- 博士 (工学)
- http://accc.riken.jp/maho/

# 多倍長精度版の BLAS/LAPACK を開発中

http://mplapack.sourceforge.net/

- BLAS (Basic Linear Algebra Subprograms) **の多倍長精度版** MBLAS。
- LAPACK (Linear Algebra PACKage) **の多倍長精度版** MLAPACK。
- **現在の状況**:MBLAS **全** 76 **ルーチンは完成。** MLAPACK **は** 40 **程度完成** (700 **程度残ってる**)。
- **対角化、コレスキー分解、** LU **分解、など。**

# 量子化学

- 化学を微視的、理論的、第一原理的に研究する。
- 電子と原子核からなる物質; 原子
- 原子と原子の結合; 分子
- 分子の組みかわり; 化学
- 量子論; シュレーディンガー方程式

# 密度行列の方法

- **シュレーディンガー方程式は解くのが難しい。**
- **変数の数をできるだけ落とした方法。**
- **縮約密度行列変分法** [Journal of Chemical Physics, 114, 8282-8292 (2001)]
- **半正定値計画に帰着できる。**

# 精度の必要性

- 量子化学の問題を半正定値計画法を使って解いていた。
- 半正定値計画法ソルバにはSDPA/SDPARAを使った。
- 大抵7-8桁の精度が出てことは事足りる。
- いくつかの問題は精度が足りなかった。
  - 規模の大きな系
  - 電子相関の強い系

# brute force solution:多倍長精度計算

- 半正定値計画:理論的に精度を上げるのが困難。
- double のままで何とかしたい:理論およびライブラリなど整備が必要。
- いずれにせよ、一度は多倍長精度計算でしっかり解かねばならない。

多倍長精度での行列演算の必要性

# SDPA-GMP:GMP バージョン SDPA

- SDPA という半正定値計画法ソルバーがある。
- GMP という多倍長精度演算ライブラリがある。
- BLAS/LAPACK など、行列演算の部分を GMP(C++) に書き換え。
- SDPA-GMP 作って公開した。
  - Journal of Chemical Physics 128, 16, 164113 (2008).
  - http://sdpa.indsys.chuo-u.ac.jp/sdpa/download.html
  - 自由なソフトウェア:GNU General Public License

# 汎用性がある MPACK(MBLAS/MLAPACK) へ

- SDPA-GMP への初期の実装は適当で汎用性がなかった。動けば良かった。
- 多倍長精度の行列演算は研究者のツールとして今後必須になるだろう。
- SDPA-GMP 以外でも恩恵があるように。
- 最適化は汎用パッケージができてからで十分。
- 最近はプロセッサが高速で、湯水のような多倍長精度計算が可能。

# MPACK (MBLAS/MLAPACK) の開発方針

- 多くのマシンで動くように。
- まずは参照実装を目指す。
- FORTRAN から C++へ移行。
- 大体同じインターフェースになるように。
- なるべく double を置き換えるだけで多倍長精度計算ができるようにしたい。
- 多倍長精度計算のライブラリになるべく依存ないよう。
- GMP(GNU Multiple Precision Arithmetic Library) を利用。
- 自由に使えるよう GNU Lesser General Public License を採用。

# GMP を利用

- GMP(GNU Multiple Precision Arithmetic Library) を利用。
- “the fastest bignum library on the planet!” (それでも遅いが)
- C++インターフェースでほぼ普通の実数型 (double) の様に使える。
- IEEE754 には従ってない (多少問題になる)。

# BLAS とは?

- The BLAS (Basic Linear Algebra Subprograms) are routines that provide standard building blocks for performing basic vector and matrix operations.
- The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. [1]
  - Level 1 : ${}^t\boldsymbol{x} \cdot \boldsymbol{x}$, ${}^t\boldsymbol{x} \cdot \boldsymbol{y}$, etc.
  - Level 2 : $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, ${}^t\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, etc.
  - Level 3 : $\alpha\boldsymbol{A}\boldsymbol{B} + \beta\boldsymbol{C}$ etc.

http://www.netlib.org/blas/faq.html [1]

# MBLAS の特徴

- Multiple precision arithmetic BLAS; BLAS の多倍長精度計算版。
- C++、GMP を使ってかかれている。
- インターフェースはほぼ BLAS と同じ。
- ほぼ double を mpf_class に置き換えるだけで多倍長精度計算になる。

# MBLASの命名規則

Prefixの変化 float, double → “R”eal, complex, double complex → “C”complex.

- saxpy, daxpy → Raxpy
- caxpy, zaxpy → Caxpy
- dgemm, sgemm → Rgemm
- cgemm, zgemm → Cgemm

# MBLASの利用例

SDPA-GMP 7.1.2のsdpa_linear.cppより

```
    if (scalar==NULL) {
      scalar = &MONE;
      // scalar is local variable
    }
    // The Point is the first argument is "Transpose".
    Rgemm("Transpose","NoTranspose",retMat.nRow,retMat.nCol,aMat.nCol,
           *scalar,aMat.de_ele,aMat.nCol,bMat.de_ele,bMat.nRow,
           0.0,retMat.de_ele,retMat.nRow);
    break;
  case DenseMatrix::COMPLETION:
    rError("no support for COMPLETION");
    break;
  }

  return _SUCCESS;
```

# MBLAS の作成方法

- Netlib からルーチンを持ってくる。(たとえば http://netlib.org/blas/drotg.f)
- 基本は手で書き換え; 小文字へ、GO TO を適当に変える、など。
- ループのアンローリングはしない。
- 2007 年くらいからぼちぼち
- FORTRAN77 のコードを見て手作業で変換
- いくつかほとんど使われてないルーチンは無視した。

とても時間がかかった; MLAPACK で改善へ

# MBLASのデバッグ

**デバグは本質的に難しい。**

- **とにかくいろんな値を代入。**MBLAS と BLAS **を呼んで同じ値が出るかのみチェック。**

```
for (int k = MIN_K; k < MAX_K; k++) {
 for (int n = MIN_N; n < MAX_N; n++) {
  for (int m = MIN_M; m < MAX_M; m++) {
...
   for (int lda = minlda; lda < MAX_LDA; lda++) {
    for (int ldb = minldb; ldb < MAX_LDB; ldb++) {
     for (int ldc = max(1, m); ldc < MAX_LDC; ldc++) {

      Rgemm(transa, transb, m, n, k, alpha, A, lda, B, ldb, beta, C, ldc);
      dgemm_f77(transa, transb, &m, &n, &k, &alphad, Ad, &lda,
                                            Bd, &ldb, &betad, Cd, &ldc);
...
    diff = vec_diff(C, Cd, MAT_A(ldc, n), 1);
    if (fabs(diff) > EPSILON) {
       printf("#error %lf!!\n", diff);
       errorflag = TRUE;
    }
```

# LAPACKとは?

- LAPACK provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision.

(http://www.netlib.org/lapack/faq.html より)

# MLAPACK の作成方法

700 個くらいあるので少し考えた。

- Netlib からルーチンを持ってくる。たとえば dsyev.f など
- f2c という FORTRAN から C 変換するプログラムに多少手を入れて C に変換。
- sed をヘビーに使う。
- 命名は BLAS と同じ: dsyev → Rsyev など
- あとはひたすら手作業...
  - コンパイルのみ 20-30 ルーチン/day
  - バグ取りのみ 品質確認 1 ルーチン/day
  - emacs を使うべし。

**モチベーション維持と体力勝負**

# MLAPACK 特有の問題

- 収束というのは IEEE754 の実装依存だが、あまり考えてない。
- LAPACK 3.2 が最新だが、IEEE754 依存実装が多く、LAPACK 3.0 くらいがちょうどよい。

# MLAPACK 現在の状況

- バージョン 0.0.9
- 355 個コンパイルは通る;37 個が動く。
- SDPA-GMP7.1.2 は MLAPACK/MBLAS で動く。
- Rsyev.cpp, Rsterf.cpp: 対角化 ok
- Rtrtri.cpp: 逆行列 ok
- Rpotrf.cpp: コレスキー分解 ok
- LAPACK と違ってあまり ilaenv の効果無し。
- 微妙な FORTRAN との文法違いが落とし穴。

# どれくらい速い/遅いか

- SDPA-GMP では、100~1000 倍。
- **あまり真面目には計測してません。**
- QD (C++/Fortran-90 double-double and quad-double package) **版はもう少し速くなるのではないか。**

# 今後の予定

- ちゃんと動く関数を増やす。
- QD (C++/Fortran-90 double-double and quad-double package) 版の作成。
- スレッドベースの高速化。
- BLAS/LAPACK 互換インターフェースの作成。
- 他プラットフォームのサポート。
- ドキュメント整備。
- BLAS/LAPACK とのベンチマーク。
- 手伝っていただけるとありがたい。