

「計算機実験 I」 実習 3 (2023-06-14/21)

本日 13:00 までに ITC-LMS の「実習 3 (2023/06/14) 出席・アンケート」あるいは「実習 3 (2023/06/21) 出席・アンケート」に回答してください

- 自習課題: C 言語における行列の扱いと LAPACK を用いた計算
 1. ベクトルや行列を扱うためのユーティリティ関数が (cmatrix.h) に用意されている。サンプルプログラム matrix_example.c や行列・行列積を計算するプログラム multiply.c の中身を見て、その使い方を確認せよ
 2. LU 分解のサンプルプログラム (lu_decomp.c) をコンパイル・実行せよ。コンパイル時に LAPACK と BLAS をリンク (-llapack -lblas) する必要がある (ハンドブック 2.14.2 節)

```
$ cc lu_decomp.c -o lu_decomp -llapack
$ ./lu_decomp input1.dat
```
- 自習課題: 疑似乱数 (演習時間中に補足説明あり)
 1. random.c は、Mersenne-Twister 乱数発生器 (mersenne_twister.h) により、(0, 1) の範囲で一様分布する実数乱数を生成するプログラムである。時系列を図示してみよ。種 (seed) を変えて何度か乱数を生成し比較してみよ
- 自習課題: C 言語での複素数の扱い (演習時間中に補足説明あり)
 1. complex.c は、プログラム中で複素数を扱う例である (ハンドブック 2.6 節)。複素数の絶対値や偏角、複素共役を計算するプログラムを作成し、その動作を確認せよ
- 自習課題: ECCS への SSH リモートアクセス
 1. ハンドブック 1.2 節 (リモートログインとファイル転送) を読み、SSH の概要を理解する
 2. 「計算機実験のための環境整備 - ECCS への SSH リモートアクセス」に従い、SSH 鍵の生成、登録、ログインを行う
 3. 「計算機実験のための環境整備 - 知の物理学研究センターワークステーション (ai) への SSH リモートアクセス — 2. ai に SSH 公開鍵を登録 (初回のみ)」に従い、ITC-LMS の「計算機実験 SSH 公開鍵登録フォーム」から公開鍵の登録を行う
- レポート課題*1*2
 1. lu_decomp.c を参考にして、LU 分解を用いて行列の行列式を計算するプログラムを作成せよ。 $n \times n$ の Vandermonde 行列 ($v_{ij} = x_j^{i-1}$) ($x_1 \cdots x_n$ は実数) の行列式を計算し、厳密な値 $\prod_{1 \leq i < j \leq n} (x_j - x_i)$ と比較せよ。ピボット選択で行を入れ替えると、行列式の符号が反転することに注意せよ。なお、この課題では、LU 分解 (LAPACK の dgetrf 関数、Python の scipy.linalg.lu 関数など) 以外のライブラリ関数は使ってはならない
 2. LU 分解を用いて Dirichlet 型の境界条件のもとでの二次元 Laplace 方程式の解を求めるプログラムを作成せよ。適当な境界条件 [例えば $u(0, y) = \sin(2\pi y)$, $u(1, y) = \sin(\pi y)$, $u(x, 0) = u(x, 1) = 0$] や電荷分布を仮定して解を計算し、Gnuplot の splot コマンド (あるいは Matplotlib の plot_surface など) を用いて解をプロットせよ。また、メッシュ数を増やすと、解の形や計算時間がどのように変化するか調べよ (計算時間の測り方については、ハンドブック 1.1.7 節参照)

*1 7/5 締切のレポート No.2 では、今回のレポート課題から 2 問を選んで回答

*2 言語の指定がない課題については、Python や Julia などでもプログラムを作成してもよい。ただし、その場合でも収束の様子などの解析はきちんと行うこと

3. 8 行 8 列のランダムな実対称行列、複素エルミート行列、実直交行列、複素ユニタリー行列を生成し、さらに、実際に生成された行列がそれらの性質を満たしているかチェックするプログラムを作成・実行せよ。なお、この課題では、QR 分解や特異値分解などのライブラリ関数は使ってはならない
4. 非線形連立方程式

$$f(x, y) = x^2 + y^2 - 1 = 0$$
$$g(x, y) = x^2(2 + x) - y^2(2 - x) = 0$$

を多次元のニュートン法 (講義 1 スライド p.20) を用いて解け。ヤコビ行列の逆行列をかける代わりに、LU 分解を用いて線形連立方程式を解くこと

5. Laplace 方程式の境界値問題を Gauss-Seidel 法、SOR 法で解くプログラムを作成し、計算結果や計算速度を LU 分解・Jacobi 法と比較せよ。また、収束までの回数を Jacobi 法と比較せよ。特に SOR 法の場合、パラメータ ω の選び方により、どのように収束回数が変化するか観察し、最適な ω の値について考察せよ
6. 行列・行列積の計算を行うサンプルプログラム `multiply.c` と、BLAS ライブラリの `dgemm` 関数を使ってそれと等価な計算を行う `multiply_dgemm.c` の速度を比較せよ^{*3}。行列サイズによっては 100 倍以上もの性能差が出ることがある。BLAS で使われている最適化手法について調べてみよ

^{*3} ai では、OS 付属の BLAS、LAPACK ではなく、Intel 製の MKL (Math Kernel Library) に含まれる BLAS や LAPACK を利用するのがよい。MKL を使うには、GNU C コンパイラ (`cc`, `gcc`) の代わりに Intel C コンパイラ (`icx` あるいは `icc`) を使い、`-llapack -lblas` の代わりに `-mkl` を指定してリンクする。例: `icx -O3 multiply_dgemm.c -mkl`