

「計算機実験 I」 実習課題 (2020/06/02 更新)

[対角化] 以降についてはまだ更新の可能性あり

課題番号のあとに [応用] とあるのは少し高度な課題

[数値誤差・アルゴリズム基礎]

1. C 言語における倍精度実数 (double)、単精度実数 (float) の有効桁数、最大値、(正の) 最小値を確認するプログラムを作成せよ。また、IEEE 規格において、零や無限大がどのように表現されるか調べよ
2. フィボナッチ数列 ($a_{n+2} = a_{n+1} + a_n$ ($n \geq 0$), $a_0 = 0, a_1 = 1$) を計算するプログラムを作成し、 $a_{20}, a_{30}, a_{40}, a_{50}, a_{60}$ を求めよ。桁あふれに注意すること。結果を MATLAB で計算した値と比較し、L^AT_EX の tabular 環境を使って表にまとめよ
3. $f(x) = \cos x$ について、 $x = 0.3\pi$ における $f'(x)$ の値を数値微分により計算するプログラムを作成せよ。数値微分の刻みを $h = 1, 1/2, 1/4, 1/8, \dots$ と減少させていった時、誤差がどのように振る舞うか図示せよ。一次近似 (前進差分) と二次近似 (中心差分) における誤差の振る舞いの違いを調べよ。同様に、 $x = 0.3\pi$ における $f''(x)$ の値を数値微分により計算し、その誤差を評価せよ
4. $f(x) = \tanh x + 0.2x + 0.3 = 0$ の解を Newton 法および二分法により求めよ。反復にしたがって、値がどのように真の解 (MATLAB で求めよ) に近づいていくか図示せよ。また、初期値による収束の違いを調べ、その理由について考察せよ
5. 黄金分割法において、左右の領域の幅の比が適当な値から出発した時、どのように黄金比に近づくのか考察せよ。また、実際にプログラムを走らせて考察が正しいかどうか確認せよ
6. [応用] 代数方程式の解をすべてもとめる方法について調べよ。Durand-Kerner-Aberth 法を用いて、代数方程式の全ての解を求めるプログラムを作成せよ。方程式の次数を増やすにつれ、収束までにかかる時間がどのように増えるか調べよ

[常微分方程式]

7. 空気による摩擦のあるバネの問題を考える。壁にバネが繋がれ、バネの先には質量 m の物体が繋がっている。床との摩擦は考えないものとする。バネの伸びる方向に x 座標を取り、自然長の位置を原点とすると、物体の運動方程式は以下のように与えられる。

$$m \frac{d^2x}{dt^2} = -kx - \kappa \frac{dx}{dt}$$

ここで、 k はバネ定数、 κ は摩擦の比例定数とする。Euler 法を使い $x(t)$ を $t = 30$ まで計算せよ。その際、刻み幅 h の大きさを変化させ、解の変わる様子を確認せよ。ただし、 $k = 2, \kappa = 0.2, m = 1$ 、初期条件は $x(0) = 10, x'(0) = 0$ とする。さらに、3 次の Runge-Kutta 法、4 次の Runge-Kutta 法を用いて同様の計算を行い、精度の向上の様子を調べよ

8. 古典調和振動子 $H = \frac{1}{2}(p^2 + q^2)$ をオイラー法、4 次の Runge-Kutta 法により解き、全エネルギー時間変化の様子を観察せよ。次に、逆オイラー法、リープ・フロッグ法を用いて計算を行い、全エネルギーのゆらぎの刻み幅 h 依存性を調べよ
9. Numerov 法とシューティング法を用いて、一次元井戸型ポテンシャル中の粒子のシュレディンガー方程式の固有エネルギーと固有関数の組をいくつか求めよ

10. [応用] 方程式によっては、刻み幅を小さくしても、なかなか精度が上がらないものがある。一つの例として、“硬い方程式”知られている。“硬い方程式”とは何か、これを精度良く解くためにはどうすれば良いか調べよ。また、具体的な問題について計算を行ってみよ

[連立一次方程式]

11. 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
12. 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¹
13. 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 分解で線形連立方程式を解くこと

14. ヤコビ法を用いて、二次元の Laplace 方程式の解を求めるプログラムを作成せよ
15. [応用] ガウス・ザイデル法、SOR 法を用いて、二次元の Laplace 方程式の解を求めるプログラムを作成せよ。収束までの回数を比較せよ。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増えるか観察し、最適な ω の値について考察せよ
16. [応用] 共分散行列 Σ で与えられる n 次元ガウス分布に従う乱数を生成する方法を考えよ。プログラムを作成して共分散を評価し、正しい乱数が生成されているか確認せよ

[対角化]

17. `cmatrix.h` を利用し、(何らかの意味で) 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
18. 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²。行列サイズによっては数十倍もの性能差が出ることがあるが、BLAS で使われている最適化手法について調べてみよ
19. ハウスホルダー法による対角化のサンプルプログラム `diag.c` に、得られた固有ベクトルが互いに正規直交していることや得られた固有値の精度を確認するコードを追加し実行せよ。また、複素エルミート行列に対して同様のプログラムを作成・実行せよ
20. 適当なポテンシャルの形を仮定した一次元シュレディンガー方程式の固有エネルギーと固有関数の組をハウスホルダー法により求めよ。同じ方程式を課題 9 の方法で解き、解を比較せよ
21. 対角成分は (n, n) 成分のみが 1 でそれ以外は全て 2、副対角成分 $(i, i \pm 1)$ は全て -1 の $n \times n$ 三重対角行列を考える。その固有値は、

$$\lambda_k = 2(1 - \cos(\pi(2k - 1)/(2n + 1))) \quad (k = 1, \dots, n)$$

¹ 計算時間の測り方: ハンドブック 1.1.6 節

² 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 (`icc`) を使い、`-llapack -lblas` の代わりに `-mkl` を指定してリンクする。例: `icc -O3 multiply_dgemm.c -mkl`

で与えられる。べき乗法を用いて、最大固有値を計算するプログラムを作成し、その結果を理論式と比較せよ³

22. ファイル `measurement1.dat` に、ある実験で得られたデータが収められている。1カラム目は x 、2カラム目は y 、3カラム目は y の誤差の値である。最小二乗法によりデータを多項式でフィッティングするプログラムを作成せよ。多項式の最大次数を大きくしていくとフィッティング結果はどのように変化するか？何次の多項式を使うのが最も良いと考えられるか考察せよ。さらに、同様の解析をファイル `measurement2.dat` に対して行ってみよ
23. Lanczos 法により固有値を計算するプログラムを作成せよ。Ritz 値が、繰り返しに従ってどのように振る舞うか図示してみよ。収束の速さをべき乗法と比較せよ
24. `convert2matrix.py` は、JPEG や PNG などの形式の画像ファイルをグレースケールに変換し、行列の形で書き出す Python スクリプトである。⁴ これを用いて画像ファイルを行列形式に変換した後、SVD で圧縮してみよ。どの程度まで圧縮可能か？

[その他]

25. [応用] 「計算機実験のための環境整備」 (<https://utphys-comp.github.io>) で抜けているところ (特に、Windows 環境でのコンパイラの利用、SSH アクセスなど) を作成
26. [応用] 「計算機実験ハンドブック」の 1.3 節に Emacs 以外のエディタの説明 (vi (vim)、nano) を追加
27. [応用] 「計算機実験ハンドブック」の第 2 章「C 言語入門」の C++ 言語版を追加

³式では添字は 1 から始まっているが、C 言語では 0 から始まることに注意

⁴実行には、Pillow ライブラリが必要である。ceenv では `pip install pillow` (Python2)、あるいは `pip3 install pillow` (Python3) する必要がある。