計算機実験 I (L1) — 講義・実習の概要

藤堂眞治
wistaria@phys.s.u-tokyo.ac.jp

2019/04/10

講義・実習の目的

- 理論・実験を問わず、学部〜大学院〜で必要となる現代的かつ普遍的な計算機の素養を身につける
- UNIX 環境に慣れる (シェル、ファイル操作、エディタ)
- ネットワークの活用 (リモートログイン、共同作業)
- プログラムの作成 (C 言語、コンパイラ、プログラム実行)
- 基本的な数値計算アルゴリズム・数値計算の常識を学ぶ
- 科学技術文書作成に慣れる (LaTeX, グラフ作成)

身に付けて欲しいこと

- ツールとしてないものは自分で作る (物理の伝統)
- すでにあるものは積極的に再利用する (車輪の再発明をしない)
- 数学公式と数値計算アルゴリズムは別物
- 刻み幅・近似度合いを変えて何度か計算を行う
- グラフ化して目で見てみる
- 計算量 (コスト) のスケーリング (次数) に気をつける
- 記録に残す・再現性を確保する

講義・実習内容

- UNIX 操作・ネットワーク
- プログラミング: C 言語、数値計算ライブラリの利用
- ツール: エディタ、コンパイラ、LaTeX、gnuplot
- 数値計算の基礎
- 常微分方程式の解法
- 連立一次方程式の解法
- 行列の対角化
- 線形回帰

講義と実習

- 講義
  - ツールなどの基礎知識: 実習の準備
  - アルゴリズムの基本概念: 実習の準備
  - より高度なアルゴリズムの紹介
- 実習 (2 グループに分けて)
  - 練習課題: 各自で取り組む
  - レポート課題: 個別レポートとして提出
  - 応用課題: 余裕のある人は積極的に取り組む
- スタッフ computer@exa.phys.s.u-tokyo.ac.jp
  - 講義: 藤堂
  - 実習: 鈴木助教、斉藤助教
  - 実習 TA: 曹 (藤堂研 M1)、森下 (藤堂研 M1)

評価方法・レポート

- 評価
  - 出席 (講義・実習)
  - レポート
- レポート
  - 各自が LaTeX で作成の上提出 (計 2 回)
  - 提出方法・締切については、後日指示

講義資料

- 「計算機実験」ハンドブック (配布済)
  - UNIX 入門
  - gnuplot 入門
  - C 言語入門
  - LaTeX 入門
- 講義資料、実習資料、追加資料、参考書
  - ITC-LMS で配布・提示 https://itc-lms.ecc.u-tokyo.ac.jp/lms/course?idnumber=201905150910F01
  - 公開資料については、http://exa.phys.s.u-tokyo.ac.jp/ja/lectures/2019s-computer1 でも配布

質問がある場合には、、、

1. ITC-LMS の掲示板を見る
2. ハンドブック、講義資料を確認
3. まわりの人に質問してみる
4. ネットで検索
5. 計算機実験担当者 (computer@exa.phys.s.u-tokyo.ac.jp) に相談

メールで質問するときに注意すべきこと

- (メールの) 標題をきちんとつける、きちんと名乗る
- 実行環境を明示する
- 問題を再現する手順を明記する
- 関連するファイル (C や LaTeX のソースコード等) を添付する
- エラーメッセージを添付する

実習・自習環境

- 情報基盤センター大演習室 (iMac 端末)
- 計算機端末室
  - 理学部 4 号館 1215 室 (iMac 16 台)
- 知の物理学研究センターワークステーション (準備中)
  - SSH でリモートログインして使用する (ハンドブック 2.2 節)
  - あらかじめ公開鍵の登録が必要 (実習 EX0 準備練習 4,5)
- MateriApps LIVE! (USB メモリで配布)
  - Mac, Windows PC 上で動作する仮想 UNIX 環境
  - C コンパイラ、gnuplot、エディタなど一式揃っている
  - インストール方法は USB メモリ内の README.html、setup.pdf を参照のこと
- C 言語・LaTeX・gnuplot のインストールに関する情報
  - https://github.com/todo-group/ComputerExperiments/wiki/InstallTeX

SSH によるリモートログイン

- UNIX (Mac も含む) では、SSH (Secure Shell) を使うことで、別のコンピュータに遠隔ログインして作業することができる
  例)
  `$ ssh -X remote.phys.s.u-tokyo.ac.jp -l` *username*

- 二種類の認証方式
  - パスワード認証: ハンドブック 2.2 節ではこちらを説明
  - 公開鍵認証方式: よりセキュリティーの高い方法
    近年はこちらが主流
    参考: https://hwb.ecc.u-tokyo.ac.jp/wp/information-2/coding/crypt/

SSH の公開鍵認証

- あらかじめクライアント (接続元) 側で、「秘密鍵」と「公開鍵」のペアを生成し、「公開鍵」をサーバ (接続先) に置いておく
  - 生成には `ssh-keygen` コマンドを使う (準備 EX0-3)
  - クライアント側に「秘密鍵」、サーバ側に「公開鍵」の両者が揃ってはじめて、クライアントからサーバにリモートログインできる
  - たとえ「公開鍵」が盗まれてしまっても、それだけではリモートログインできないので安心
  - 「秘密鍵」は絶対に人に見られてはならない
- 鍵の場所
  - 秘密鍵: `$HOME/.ssh/id_rsa` に生成される
  - 公開鍵: `$HOME/.ssh/id_rsa.pub` に生成される ⇒ サーバの `$HOME/.ssh/authorized_keys` にコピーする

MATLAB の利用

- MATLAB は、数値計算、数式処理、統計、機械学習、画像処理、信号処理などのライブラリが充実したソフトウェア (cf. Mathematica)
  - 高機能な電卓としての利用
  - C 言語でアルゴリズムやプログラムの基本構造を学んだ上で MATLAB を活用することで、すばやく簡単にプログラム作成可能に
- 2019 年 4 月から全学で MATLAB の利用が可能に
  - 東京大学の正規学生・教職員はどこにいても MATLAB 利用可
  - PC・ Web ブラウザ・タブレット・スマホなどにインストール可 (台数制限なし)
- インストール・基本的な使い方については「MATLAB Quick-Start」を参照のこと
  https://elf-c.he.u-tokyo.ac.jp/courses/375

数値誤差の原因

- 丸め誤差: 無理数や 10 進数を有限のビットの 2 進数で表現することによる誤差 (例: 0.1 が 0.0999999999998 になる)
- 打ち切り誤差: テイラー展開による近似を有限項で打ち切ることによる誤差 (例: 数値微分)
- 桁落ち: 非常に近い数の引き算により生じる
- 情報落ち: 非常に大きな数に小さな数を足し込む場合に生じる (例: 数値積分や常微分方程式の初期値問題で刻み幅を小さくしすぎると生じる)
- オーバーフロー (桁あふれ): 表現できる値を超えてしまう

桁落ち

- 2 次方程式 $ax^2 + bx + c = 0$ の解の公式

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$b^2 \gg |ac|$ の時、桁落ちが生じる

- 例) $2.718282x^2 - 684.4566x + 0.3161592 = 0$ の解を 7 桁の精度で計算してみる (伊理・藤野 1985)

$$\sqrt{D} = \sqrt{(684.4566)^2 - 4 \times 2.718282 \times 0.3161592} = 684.4541$$

$$x_+ = \frac{684.4566 + 684.4541}{2 \times 2.718282} = \frac{1368.911}{5.436564} = 251.7970$$

$$x_- = \frac{684.4566 - 684.4541}{2 \times 2.718282} = \frac{0.0025}{5.436564} = 0.0004\underline{98493}$$

桁落ちを防ぐ方法

- b の符号に応じて、一方を求める (この例では x_+)
- 他方は解と係数の関係を使って求める

$$x_- = \frac{c/a}{x_+} = \frac{0.3161592/2.718282}{251.7970} = 0.0004619138$$

- 回避できない例: 重解に近い場合
  $2.718282x^2 - 1.854089x + 0.3161592 = 0$

$$\begin{aligned}\sqrt{D} &= \sqrt{(1.854089)^2 - 4 \times 2.718282 \times 0.3161592} \\ &= 0.00264575 \\ x_\pm &= 1.854089 \pm 0.00264575 = 1.856737, 1.851445\end{aligned}$$

数値微分

- 関数のテイラー展開

$$f(x+h) = f(x) + hf'(x) + h^2 f''(x)/2 + h^3 f'''(x)/6 + \cdots$$

- 数値微分の最低次近似

$$f_1(x,h) \equiv \frac{f(x+h) - f(x)}{h} = f'(x) + hf''(x)/2 + O(h^2)$$

- より高次の近似

$$f_2(x,h) \equiv \frac{f(x+h) - f(x-h)}{2h} = f'(x) + h^2 f'''(x)/6 + O(h^3)$$

- 刻み h を小さくすると打ち切り誤差は減少するが、小さすぎると今度は桁落ちが大きくなる

刻み幅を変えた計算

- 刻み幅を変えて何度か計算を行い、収束の様子をみる
- グラフ化して目で見てみる
- 理論式と比較
  - 計算式の正しさの確認
  - 近似の改良 (収束の加速・補外)
- 桁落ち・情報落ちの影響の有無

ニュートン法

- 反復法により方程式 $f(x) = 0$ の解を求める
- 真の解を x_0、現在の解の候補を $x_n = x_0 + \epsilon$ とすると

$$0 = f(x_0) = f(x_0 + \epsilon - \epsilon) = f(x_n) - f'(x_n)\epsilon + O(\epsilon^2)$$

- 次の解の候補 (反復法、逐次近似法)

$$\epsilon \approx \frac{f(x_n)}{f'(x_n)} \qquad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- 複素変数の複素関数や多変数の場合にも自然に拡張可

ニュートン法の収束

- x_n が x_0 に十分近い時

$$f(x_n) \approx f'(x_0)(x_n - x_0) + f''(x_0)\frac{(x_n - x_0)^2}{2}$$
$$f'(x_n) \approx f'(x_0) + f''(x_0)(x_n - x_0)$$

- ニュートン法で一回反復すると

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \approx x_n - (1 - \frac{f''(x_0)}{f'(x_0)}\frac{(x_n - x_0)}{2})(x_n - x_0)$$
$$(x_{n+1} - x_0) \approx \frac{f''(x_0)}{2f'(x_0)}(x_n - x_0)^2$$

- 一回の反復で誤差が 2 乗で減る (正しい桁数が倍に増える)
  ⇒ 二次収束

多次元の場合

- $f(x) = 0$: d 次元 (非線形) 連立方程式
- x は d 次元のベクトル: $x = (x_1, x_2, \cdots, x_d)$
- $f(x)$ も d 次元のベクトル: $f(x) = (f_1(x), f_2(x), \cdots, f_d(x))$
- 真の解のまわりでの展開 $(x_n = x_0 + \epsilon)$

$$0 = f(x_0) = f(x_0 + \epsilon - \epsilon) = f(x_n) - \frac{\partial f(x_n)}{\partial x} \cdot \epsilon + O(|\epsilon|^2)$$

- ヤコビ行列 $(d \times d)$: $\left(\frac{\partial f(x_n)}{\partial x}\right)_{ij} = \frac{\partial f_i(x_n)}{\partial x_j}$
- 次の解の候補: $x_{n+1} = x_n - \left(\frac{\partial f(x_n)}{\partial x}\right)^{-1} f(x_n)$

ニュートン法による最適化

- x は d 次元のベクトル: $x = (x_1, x_2, \cdots, x_d)$, $g(x)$ はスカラー
- 勾配ベクトル: $(\nabla g(x))_i = \dfrac{\partial g(x)}{\partial x_i}$
- 極小値 (最小値) となる条件: $\nabla g(x) = 0$
- ニュートン法で $f(x)$ を $\nabla g(x)$ で置き換えればよい
- 次の解の候補: $x_{n+1} = x_n - H^{-1}(x_n)\nabla g(x_n)$
- ヘッセ行列 (Hessian): $H_{ij}(x_n) = \dfrac{\partial^2 g}{\partial x_i \partial x_j}(x_n)$

準ニュートン法

- ニュートン法では、ヘッセ行列の計算・保存が必要
- 準ニュートン法: それまでの反復で計算した勾配ベクトルから、ヘッセ行列を近似 (B_n)
- BFGS 法 (Broyden-Fletcher-Goldfarb-Shanno)

$$B_{n+1} = B_n + \frac{y_n y_n^T}{y_n^T s_n} - \frac{B_n s_n (B_n s_n)^T}{s_n^T B_n s_n}$$

- $s_n = x_{n+1} - x_n$、$y_n = \nabla g(x_{n+1}) - \nabla g(x_n)$
- 直接 B_n の逆行列 C_n を更新することも可能

$$C_{n+1} = B_{n+1}^{-1} = C_n + \left(1 + \frac{y_n^T C_n y_n}{y_n^T s_n}\right) \frac{s_n s_n^T}{y_n^T s_n} - \frac{C_n y_n s_n^T + s_n y_n^T C_n^T}{y_n^T s_n}$$

- 他にも、SR1 法、BHHH 法、記憶制限 BFGS 法

計算をいつやめるか？

- 残差による判定

$$|f(x)| < \delta$$

- 誤差による判定

$$|x_{n+1} - x_n| < \epsilon$$

- 解 $x = x_0$ が m 重解の場合、$x = x_0$ のまわりで展開すると

$$f(x) \simeq \alpha(x - x_0)^m$$

残差が δ 程度になったときの誤差は、$\delta^{1/m}$ 程度
逆に $|x - x_0|$ が $\delta^{1/m}$ 以下になると、$f(x)$ の値がそれ以上変化しない $\Rightarrow$ m 重解の精度は計算精度の $1/m$ 桁程度しかない

- 残差による判定と誤差による判定を併用するのがよい

反復計算

- while による反復 (ハンドブック 3.2.3 節)

```
double residual = 1;     /* 残差 */
double error = 1;        /* 誤差 */
double delta = 1.0e-12; // 欲しい精度
while (residual > delta && error > delta) {
  /* ニュートン法の漸化式 */
  /* residual と error を計算 */
}
```

残差と誤差のどちらかが欲しい精度に達したら計算を終了

- break を使う例 (ハンドブック 3.2.4 節)

```
for (;;) {
  /* ニュートン法の漸化式 */
  /* residual と error を計算 */
  if (residual < delta || error < delta) break;
}
```

初期段階における収束の改善

- Newton 法は初期値によっては収束しない
- 発散や振動を抑える方法として「減速」が有効な場合も
- 減速
  - 反復式を少し修正する

$$x_{n+1} = x_n - \mu_n \frac{f(x_n)}{f'(x_n)}$$

  - まずは $\mu_n = 1$ として計算
  - $|f(x_{n+1})| < |f(x_n)|$ が成り立たないようであれば、μ_n を半分にして再計算
  - μ_n が十分に小さくなれば、$|f(x)|$ は必ず減少する