計算機実験II (L1) — 対角化と量子力学

藤堂眞治
wistaria@phys.s.u-tokyo.ac.jp

2019/9/27

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)
- 物理学における具体的な問題を通して実践的な知識と経験を身につける

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
- 問題の解き方は一通りではない
- いろいろな手法を組み合わせて使う

講義・実習内容

- 問題解決型: 計算機実験 I で身に付けた知識をもとに、より高度な数値計算手法・アルゴリズムを学び、物理学における具体的な問題への応用を通して実践的な知識と経験を身につける
  - 数値対角化と量子力学
  - モンテカルロ法・分子動力学と統計物理
  - 最適化問題
- 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 M2)、森下 (藤堂研 M1)

二重井戸ポテンシャル中の粒子

- 時間依存しないシュレディンガー方程式

$$\Big[-\frac{d^2}{dx^2}+V(x)\Big]\psi(x)=E\psi(x)$$

($\hbar^2/2m=1$ となるように単位をとった)

- 二重井戸ポテンシャル

$$V(x)=\begin{cases}\infty & x<0,\ x>1\\ 0 & 0<x<a,\ b<x<1\\ v & a<x<b\end{cases}$$

ただし、$0<a<b<1$ とする

- 境界条件: $\psi(0)=\psi(1)=0$、$0<x<1$ で $\psi(x)$ とその導関数が連続

シュレディンガー方程式の解法

- シューティング
  - 計算機実験 I (L2)
  - シューティングに用いる積分法: 2 階常微分方程式の 2 次元 1 階連立微分方程式への書き換え、オイラー法とその改良、Numerov 法
- ハミルトニアンの対角化
  - 計算機実験 I (L4)
  - 対角化手法: ハウスホルダー法 (LAPACK)、べき乗法、Lanczos 法
- 変分法: 変分関数のパラメータの最適化
- その他の方法: 手で解けるところはあらかじめ解いて次元を減らす
- それぞれのコスト (=計算時間・メモリ) は?

準備: 微分方程式の書き換え

- 2 階の常微分方程式の一般形

$$\frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = r(x)$$

- $y_1 \equiv y$, $y_2 \equiv \frac{dy}{dx}$ とおくと

$$\begin{cases} \frac{dy_1}{dx} &= y_2 \\ \frac{dy_2}{dx} &= r(x) - p(x)y_2 - q(x)y_1 \end{cases}$$

- さらに $\boldsymbol{y} \equiv (y_1, y_2)$, $\boldsymbol{f}(x, \boldsymbol{y}) \equiv (y_2, r(x) - p(x)y_2 - q(x)y_1)$

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(x, \boldsymbol{y})$$

- n 階常微分方程式 $\Rightarrow$ n 次元の 1 階常微分方程式

初期値問題の解法 (Euler 法)

- h を微小量として微分を差分で近似する (前進差分)

$$\frac{dy}{dt} \approx \frac{y(t+h) - y(t)}{h} = f(t, y)$$

- $t = 0$ における $y(t)$ の初期値を y_0、$t_n \equiv nh$、y_n を $y(t_n)$ の近似値とおくと、

$$y_{n+1} - y_n = hf(t_n, y_n)$$

- Euler 法
  - y_0 からはじめて、$y_1, y_2, \cdots$ を順次求めていく

高次の Runge-Kutta 法

- 3 次 Runge-Kutta 法

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}k_1) \\ k_3 &= hf(t_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}k_2) \\ y_{n+1} &= y_n + \tfrac{1}{4}k_1 + \tfrac{3}{8}k_2 + \tfrac{3}{8}k_3 \end{aligned}$$

- 4 次 Runge-Kutta 法

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1) \\ k_3 &= hf(t_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_2) \\ k_4 &= hf(t_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4 \end{aligned}$$

- 4 次までは次数と f の計算回数が等しい

計算コストと精度

- 実際の計算では $f(t, y)$ の計算にほとんどのコストがかかる
- 計算回数と計算精度の関係

	1 次 (Euler 法)	2 次 (中点法)	3 次	4 次
計算精度	$O(h)$	$O(h^2)$	$O(h^3)$	$O(h^4)$
計算回数	N	$2N$	$3N$	$4N$

- 高次の Runge-Kutta を使う方が効率的
- どれくらい小さな h が必要となるか、前もっては分からない
- 刻み幅を変えて $(h, h/2, h/4, \dots)$ 計算してみることが大事
  - 誤差の評価
  - 公式の間違いの発見

Numerov 法

- Numerov 法
  - 二階の常微分方程式で一階の項がない場合に使える
  - 4 次の陰解法
  - 方程式が線形の場合は陽解法に書き直せる

- 微分方程式

$$\frac{d^2y}{dx^2} = f(x, y)$$

$y = y(x)$ を $x = x_i$ のまわりでテイラー展開する。
$x_{i\pm1} = x_i \pm h$ での表式は

$$y(x_{i\pm1}) = y(x_i) \pm hy'(x_i) + \frac{h^2}{2}y''(x_i) \pm \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y''''(x_i) + O(h^5)$$

Numerov 法

- 二階微分の差分近似 ($y_i \equiv y(x_i)$ 等と書く)

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = y_i'' + \frac{h^2}{12} y_i'''' + O(h^4)$$

一方で、微分方程式より

$$y_i'''' = \left.\frac{d^2 f}{dx^2}\right|_{x=x_i} = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2)$$

組み合わせると

$$y_{i+1} = 2y_i - y_{i-1} + \frac{h^2}{12}(f_{i+1} + 10f_i + f_{i-1}) + O(h^6)$$

Numerov 法

- 方程式が線形の場合、$f(x,y) = -a(x)y(x)$ を代入すると

$$y_{i+1} = 2y_i - y_{i-1} - \frac{h^2}{12}(a_{i+1}y_{i+1} + 10a_i y_i + a_{i-1}y_{i-1}) + O(h^6)$$

y_{i+1} を左辺に集めると、陽解法となる

$$y_{i+1} = \frac{2(1 - \frac{5h^2}{12}a_i)y_i - (1 + \frac{h^2}{12}a_{i-1})y_{i-1}}{1 + \frac{h^2}{12}a_{i+1}} + O(h^6)$$

常微分方程式の解法の利用

- $x_i = h \times i$ $(h = 1/n)$、$x_0 = 0$、$x_n = 1$ とする
- $\psi(x_0) = 0$、$\psi(x_1) = 1$ を仮定 $(\psi'(x_0) = 1/h$ と与えたことに相当)
- $E = 0$ とおく
- 初期条件のもとで、$x = x_n$ まで積分
- $\psi(x_n)$ の符号がかわるまで、E を少しずつ増やす
- 符号が変わったら、E の区間を半分ずつに狭めていき、$\psi(x_n) = 0$ となる E (固有エネルギー) と $\psi(x)$ (波動関数) を得る

二分法

- 反復法により一次元の方程式 $f(x) = 0$ の解を求める
- 導関数を使わず関数値のみを利用 (c.f. ニュートン法)
- 初期条件として、$f(a) \times f(b) < 0$ を満たす 2 点の組 $(a < b)$ で解をはさみ込み、領域を狭めていく
- a と b の中点 $x = (a + b)/2$ を考える
  - $|f(x)|$ が十分小さい場合: x が解
  - $f(a) \times f(x) < 0$ の場合: $[a, x]$ を新しい領域にとる
  - $f(x) \times f(b) < 0$ の場合: $[x, b]$ を新しい領域にとる
- 領域 $[a, b]$ の幅が十分小さくなったら終了
- 反復のたびに領域の幅は半分になる
- 全ての解を得られる保証はない
- 二分法の例: bisection.c

対角化

- シュレディンガー方程式の行列表示
- ハウスホルダー法 (LAPACK)
- べき乗法
- Lanczos 法
- ハウスホルダー法によるプログラムの例: double_well.c

シュレディンガー方程式の行列表示

- シュレディンガー方程式

$$[-\frac{d^2}{dx^2} + V(x)]\psi(x) = E\psi(x)$$

- 連立差分方程式を行列の形で表す $(\psi(x_0) = \psi(x_n) = 0)$

$$\begin{pmatrix} \frac{2}{h^2} + V(x_1) & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} + V(x_2) & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{2}{h^2} + V(x_3) & -\frac{1}{h^2} & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} + V(x_{n-1}) \end{pmatrix} \begin{pmatrix} \psi(x_1) \\ \psi(x_2) \\ \psi(x_3) \\ \vdots \\ \psi(x_{n-1}) \end{pmatrix} = \cdots$$

- $(n-1) \times (n-1)$ の疎行列の固有値問題
  - 固有値: 固有エネルギー
  - 固有ベクトル: 波動関数

行列の数値対角化

- 一般的に次元が 5 以上の行列の固有値は、あらかじめ定まる有限回の手続きでは求まらない
- 必ず何らかの反復法 (+収束判定) が必要となる
- 密行列向きの方法
  - Jacobi 法
  - Givens 変換・Householder 法 (三重対角化) + QR 法など
- 疎行列向きの方法
  - べき乗法
  - Lanczos 法 (三重対角化) + QR 法など
- 固有ベクトル
  - 逆反復法

基本方針

- やってはいけない方法: 特性方程式

$$|\lambda E - A| = 0$$

  の係数を求めて、代数方程式として解く
  - 数値的に不安定 (代数方程式の解は係数の誤差に対して敏感)
  - 計算コスト大 [$\sim O(N!)$]
- スタンダードな方法: 行列を次々に直交変換して、対角行列 (あるいは三重対角行列) に近づけていく

$$A \to U_1^T A U_1 \to U_2^T (U_1^T A U_1) U_2 \to U_3^T (U_2^T (U_1^T A U_1) U_2) U_3 \to \cdots$$

- 固有値は変換された行列の固有値、固有ベクトルは変換後の行列の固有ベクトルに左から $U_1 U_2 U_3 \cdots$ を掛けたもの

LAPACK の対角化ルーチン

- 様々な対角化ルーチンが準備されている
  - 倍精度実対称行列の対角化 dsyev http://www.netlib.org/lapack/explore-html/dd/d4c/dsyev_8f.html
  - Fortran による関数宣言

```
subroutine dsyev(character JOBZ, character UPLO,
  integer N, double precision, dimension(lda, *) A,
  integer LDA, double precision, dimension(*) W,
  double precision, dimension(*) WORK,
  integer LWORK, integer INFO)^^I^^I
```

- 他にも dsyevd、dsyevr、dsyevx などがある
  3 重対角化までは同じ。3 重対角行列の対角化が異なる
- 単精度版の ssyev、複素 (エルミート行列) 版の zheev など
- dsyev の使用例: diag.c

cmatrix.h ライブラリ

- Column-major 形式の二次元配列の確保 (`alloc_dmatrix`)、開放 (`free_dmatrix`)、出力 (`print_dmatrix`)、読み込み (`read_dmatrix`) を行うためのユーティリティ関数、(i,j) 成分にアクセスするためのマクロ (`mat_elem`) 他を準備
- ソースコード: cmatrix.h
- 使用例

```
#include "cmatrix.h"
...
double **mat;
mat = alloc_dmatrix(m, n);
mat_elem(mat, 1, 3) = 5.0;
...
free_dmatrix(mat);
```

- サンプルコード: matrix_example.c

要素アクセス・先頭アドレス

- 行列の (i,j) 成分を a[j][i] に格納することにする (column-major)
  - cmatrix.h ではマクロ (mat_elem) を準備

```
#define mat_elem(mat, i, j) (mat)[j][i]
```

  - このマクロを使うと、例えば (i,j) 成分への代入は以下のように書ける

```
mat_elem(a, i, j) = 1;
```

- LAPACK にベクトルや行列の最初の要素へのポインタを渡す
  - ベクトルの最初の要素 (0) へのポインタ: &v[0]
  - 行列の最初の要素 (0,0) へのポインタ: &a[0][0]
  - cmatrix.h にマクロ (vec_ptr、mat_ptr) が準備されているのでそれぞれ、vec_ptr(v)、mat_ptr(a) と書ける

反復法

- 疎行列の場合、行列ベクトル積は高速に行える
- Givens 変換、Householder 変換などを行うと疎行列性が失われる
- 行列ベクトル積のみを用いる反復法が効果的
  - べき乗法
  - Lanczos 法

べき乗法 (Power Method)

- 適当なベクトル v_1 から出発する
- v_1 が最大固有ベクトル ξ_1 と直交していないとすると

$$v_1 = c_1\xi_1 + c_2\xi_2 + c_3\xi_3 + \cdots + c_n\xi_n$$

と展開できる ($c_1 \neq 0$)。この両辺に A を次々掛けて行くと

$$v_2 = Av_1 = c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + c_3\lambda_3\xi_3 + \cdots + c_n\lambda_n\xi_n$$
$$v_3 = A^2v_1 = c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + c_3\lambda_3^2\xi_3 + \cdots + c_n\lambda_n^2\xi_n$$
$$\vdots$$
$$v_{k+1} = A^kv_1 = c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + c_3\lambda_3^k\xi_3 + \cdots + c_n\lambda_n^k\xi_n$$
$$= c_1\lambda_1^k\left[\xi_1 + \sum_{i=2}^{n}\frac{c_i}{c_1}\left(\frac{\lambda_i}{\lambda_1}\right)^k\xi_i\right] \approx c_1\lambda_1^k\xi_1$$

べき乗法の収束

- べき乗法による固有値

$$\frac{v_{k+1}^T v_{k+1}}{v_{k+1}^T v_k} = \lambda_1 + O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right)$$

- 誤差の収束

$$\frac{v_{k+1}^T v_{k+1}}{v_{k+1}^T v_k} \approx \lambda_1 + e^{-2k \ln(\lambda_1/\lambda_2)}$$

- $1/\ln(\lambda_1/\lambda_2)$ 程度の反復が必要
- λ_1 と λ_2 が近い場合には、反復回数が非常に多くなる

Rayleigh-Ritz の方法

- $n \times n$ 行列 A について、互いに正規直交するベクトル $v_1, v_2, \cdots, v_m$ $(m < n)$ が張る部分空間の中で「最良の」固有ベクトルを求めたい

- $n \times m$ 行列

$$V = (v_1 v_2 \cdots v_m)$$

を定義すると、$V^T V = E_m$ が成り立つ (ただし $VV^T \neq E_n$)

- 部分空間内のベクトルを $w = \sum_i a_i v_i$ と表すと、$\frac{w^T A w}{w^T w}$ が極大値を取る (本当の固有ベクトルにできるだけ平行になる) 条件は、

$$\frac{\partial}{\partial a_i} \frac{w^T A w}{w^T w} \sim \sum_j H_{ij} a_j - \lambda a_i = 0$$

Rayleigh-Ritz の方法

- $m \times m$ 行列
$$H_{ij} = v_i^T A_{ij} v_j$$
に対する固有値問題 $Ha = \lambda a$
- λ: もとの行列の近似固有値 (Ritz 値)
- Va: もとの行列の近似固有ベクトル (Ritz ベクトル)
- 最大固有値に対する良い近似固有値が欲しい場合 $\Rightarrow$
$v_1, v_2, \cdots, v_m$ を最大固有ベクトルになるべく近い (しかし、互いに直交する) ベクトルに選べば良い

Lanczos 法

- 初期 (ランダム) ベクトル v_1 に加えて

$$Av_1, Av_1, \cdots A^{m-1}v_1$$

を正規直交化して $v_1, v_2, \cdots, v_m$ を作る (Krylov 部分空間)

- 部分空間での Ritz 値を固有値の近似値とする
- $A^k v_1$ はどんどん最大固有ベクトルに近づいていくので、$m \ll n$ でも良い近似固有値が得られると期待される

変分法

- 波動関数を互いに直交する正規化された波動関数 (基底関数) の線形結合で近似する (変分波動関数、試行関数)

$$|\psi\rangle = \sum_{p=1}^{N} C_p |\phi_p\rangle \qquad (\langle\phi_p|\phi_q\rangle = \delta_{pq})$$

- エネルギーの期待値

$$E = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\sum_{p,q} C_p^* H_{pq} C_q}{\sum_{p,q} C_p^* \delta_{pq} C_q}$$

$$H_{pq} = \langle\phi_p|H|\phi_q\rangle$$

- E ができるだけ小さくなるよう係数 C_p を最適化 (変分原理)

変分法

- $\delta E = 0$ から

$$\sum_q (H_{pq} - E\delta_{pq}) C_q = 0 \qquad \text{for } {}^\forall p$$

- H_{pq}, δ_{pq} を $N \times N$ 行列と考えると、固有値問題とみなせる

$$HC = EC$$

- H はエルミート行列
- $\{\phi_p\}$ の張る部分空間での最適化 (= Rayleigh-Ritz の方法)
- 変分波動関数と真の波動関数の差が ϵ 程度の時、E と真の固有エネルギーの差は ϵ^2 程度

非直交基底関数による変分法

- 重なり積分

$$S_{pq} = \langle \phi_p | \phi_q \rangle \neq \delta_{pq}$$

- 変分波動関数の正規化条件

$$\langle \psi | \psi \rangle = \sum_{p,q} C_p^* \langle \phi_p | \phi_q \rangle C_q = \sum_{p,q} C_p^* S_{pq} C_q = 1$$

- エネルギー期待値

$$E = \frac{\sum_{p,q} C_p^* H_{pq} C_q}{\sum_{p,q} C_p^* S_{pq} C_q}$$

- $\delta E = 0$ から

$$\sum_q (H_{pq} - E S_{pq}) C_q = 0 \;\Rightarrow\; HC = ESC \text{ (一般化固有値問題)}$$

一般化固有値問題

- 重なり行列 $S_{pq} = \langle \phi_p | \phi_q \rangle$
  - エルミート行列: $S_{pq} = S^*_{qp}$
  - 正定値 ($\{\phi_p\}$ が線形独立の場合):

$$x^\dagger S x = \sum_{pq} \langle \phi_p | \phi_q \rangle x^*_p x_q = || \sum_p x_p | \phi_p \rangle ||^2 > 0$$

- 一般化固有値問題 $\Rightarrow$ 2 回の対角化
  - S を固有値分解: $S = UDU^\dagger$
  - S 固有値は全て正 $\Rightarrow$ $D^{-1/2}$ を定義可
  - $HC = ESC \Rightarrow D^{-1/2} U^\dagger H U D^{-1/2} D^{1/2} U^\dagger C = E D^{1/2} U^\dagger C$
  - $H' = D^{-1/2} U^\dagger H U D^{-1/2}$、$C' D^{1/2} U^\dagger C$ とおくと

$$H' C' = E C' \qquad \text{(通常の) 固有値問題}$$

シュレディンガー方程式の一般解

- 二重井戸ポテンシャル

$$V(x) = \begin{cases} \infty & x < 0,\ x > 1 \\ 0 & 0 < x < a,\ b < x < 1 \\ v & a < x < b \end{cases}$$

- それぞれの領域内では手で解ける
- 領域 1 ($0 < x < a$)、領域 3 ($b < x < 1$) では

$$-\frac{d^2}{dx^2}\psi(x) = E\psi(x)$$

シュレディンガー方程式の一般解

- 領域 1 ($0 < x < a$)、領域 3 ($b < x < 1$) における一般解

$$\psi(x) = A_1 e^{i\sqrt{E}x} + B_1 e^{-i\sqrt{E}x}$$
$$\psi(x) = A_3 e^{i\sqrt{E}x} + B_3 e^{-i\sqrt{E}x}$$

あきらかに $E > 0$ であるので

$$\psi(x) = \alpha_1 \cos(\sqrt{E}x) + \beta_1 \sin(\sqrt{E}x)$$
$$\psi(x) = \alpha_3 \cos(\sqrt{E}x) + \beta_3 \sin(\sqrt{E}x)$$

シュレディンガー方程式の一般解

- 領域 2 ($a < x < b$) における一般解

$$\psi(x) = A_2 e^{i\sqrt{(E-v)}\,x} + B_1 e^{-i\sqrt{(E-v)}\,x}$$

- $E < v$ の場合

$$\psi(x) = \alpha_2 \exp(-\sqrt{(v-E)}\,x) + \beta_2 \exp(\sqrt{(v-E)}\,x)$$

- $E > v$ の場合

$$\psi(x) = \alpha_2 \cos(\sqrt{(E-v)}\,x) + \beta_2 \sin(\sqrt{(E-v)}\,x)$$

シュレディンガー方程式の一般解

- 境界条件 ($E < v$ の場合)

$$\alpha_1 = 0$$

$$\alpha_1 \cos(\sqrt{E}a) + \beta_1 \sin(\sqrt{E}a)$$
$$= \alpha_2 \exp(-\sqrt{(v-E)}\,a) + \beta_2 \exp(\sqrt{(v-E)}\,a)$$
$$-\alpha_1\sqrt{E}\sin(\sqrt{E}a) + \beta_1\sqrt{E}\sin(\sqrt{E}a)$$
$$= -\alpha_2\sqrt{(v-E)}\exp(-\sqrt{(v-E)}\,a) + \beta_2\sqrt{(v-E)}\exp(\sqrt{(v-$$
$$\cdots$$

- β_1, α_2, β_2, α_3, β_3 に関する連立方程式: $Mx = 0$
- 5×5 行列 M は E の (非線形な) 関数
- 非自明な解が存在するための条件: $\det M = 0$

行列式の計算

- ガウスの消去法と LU 分解
- LU 分解による行列式の計算
- LAPACK による LU 分解
- LU 分解のプログラムの例: lu_decomp.c
- 行列式計算のプログラムの例: determinant.c

逆行列の「間違った」求め方

- 線形代数の教科書に載っている公式

$$A^{-1} = \frac{\tilde{A}}{|A|}$$

  $|A|$: A の行列式、$\tilde{A}$: A の余因子行列

- $n \times n$ 行列の行列式を定義通り計算すると、計算量$\sim O(n!)$
- したがって、上の方法で逆行列を計算すると、計算量$\sim O(n!)$
- $n = 100$ の場合: $n! \approx 9.3 \times 10^{157}$

逆行列の「正しい」求め方

- 連立一次方程式 $A\mathbf{x} = \mathbf{e}_j$ を全ての $\mathbf{e}_j$ について解く
- Gauss の消去法による連立一次方程式の解法: 計算量$\sim O(n^3)$
- Gauss の消去法の途中で出てくる下三角行列 (L) と上三角行列 (U) 行列を再利用 (LU 分解) すれば、逆行列全体を求めるための計算量も $O(n^3)$
- A の行列式も $O(n^3)$ で計算可
- $n = 100$ の場合: $n^3 = 10^6 \ll 9.3 \times 10^{157}$

LU 分解

- LU 分解による連立一次方程式の解法
  - 方程式は $A\mathbf{x} = LU\mathbf{x} = \mathbf{b}$ と書ける
  - まず、$L\mathbf{y} = \mathbf{b}$ を解いて、$\mathbf{y}$ を求める (前進代入)
  - 次に、$U\mathbf{x} = \mathbf{y}$ を解いて、$\mathbf{x}$ を求める (後退代入)
- 計算量はガウスの消去法と変わらない
- 一度 LU 分解をしておけば、異なる $\mathbf{b}$ に対する解も簡単に求められる
- 行列式は U の対角成分の積で与えられる (ピボット選択により符号が変わることに注意)

LAPACK による連立一次方程式の求解

- LU 分解を行うサブルーチン dgetrf
  http://www.netlib.org/lapack/explore-html/d3/d6a/dgetrf_8f.html
- Fortran による関数宣言

```
subroutine dgetrf(integer M, integer N,
          double precision, dimension(lda, *) A,
          integer LDA, integer, dimension(*) IPIV,
          integer INFO)
```

- A: 左辺の行列、M,N: 次元、IPIV: 選択されたピボット行のリスト、lda: 通常 M (行数) と同じで良い

LAPACK による連立一次方程式の求解

- C 言語から呼び出すための関数宣言を作成 (ハンドブック 3.6.4 節)

```
void dgetrf_(int *M, int *N, double *A,
             int *LDA, int*IPIV, int *INFO);
```

関数名は全て小文字。関数名の最後に _ (下線) を付ける

- LU 分解の例

```
m = 10;
n = 10;
a = alloc_dmatrix(m, n);
...
dgetrf_(&m, &n, mat_ptr(a), &m, vec_ptr(ipiv), &info);
```

完全なソースコード: lu_decomp.c

まとめ

- 一次元 (一粒子) シュレディンガー方程式の固有値問題
  - シューティングによる方法: 連立微分方程式の積分、二分法
  - 対角化による方法: 差分化、ハウスホルダー法、べき乗法、Lanczos 法
  - 変分法 (線形の場合): 固有値問題、一般化固有値問題
  - 特別な場合に有効な方法: LU 分解
- 計算精度は何で決まるか？
- それぞれのコスト (=計算時間・メモリ) は？