応用数理実験

固有値問題 (2) 実対称行列の三重対角化

桂田 祐史

2018 年 8 月 22 日

1 今回の課題

(1) Householder 変換または Lanczos 法により実対称行列を三重対角行列に相似変換するプログラムを作成せよ (相似変換のための行列は求めなくとも構わない)。

(2) 実対称三重対角行列に対して (その特性を十分生かして)、冪(べき)乗法、逆反復法により、絶対値が最大の固有値と絶対値最小の固有値を求めるプログラムを作成し、(1) で作ったプログラムと結合せよ。

(3) 三重対角行列に対して、二分法により固有値を求めるプログラムを作成せよ。

三重対角化のプログラムのテスト用のデータとしては、例えば

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

を用いよ[1]。これを講義で説明した Householder 法[2]で三重対角化すると

$$P^{-1}AP = \begin{pmatrix} 1.0000000000 & -1.7320508076 & 0 & 0 \\ -1.7320508076 & 7.6666666667 & 1.2472191289 & 0 \\ 0 & 1.2472191289 & 0.9761904762 & -0.1237179148 \\ 0 & 0 & -0.1237179148 & 0.3571428571 \end{pmatrix}$$

となる。さらに固有値は

$$\begin{aligned} \lambda_1 &= 8.290859369381589606621222410409759227\cdots, \\ \lambda_2 &= 1, \\ \lambda_3 &= 0.426022047760461836484914938273277876 12\cdots, \\ \lambda_4 &= 0.283118582857948556893862651316962896 25\cdots, \end{aligned}$$

となる。

プログラムの正しさに自信が持てたら大きな行列で実験してみること。

[1] この行列も前回と同様、一松信著「数値解析」朝倉書店 (1982) から採った。

[2] Lanczos 法を用いた場合も、相似変換のための直交行列の第 1 列ベクトルを $\mathbf{e}_1 = {}^t(1, 0, \cdots, 0)$ に取ると、(符合の違いを無視すれば) 同じ結果になるはずである。

2　参考 — 三重対角行列に対する Gauss の消去法

2.1　C 言語版

関数 trid(), trilu(), trisol()

```
/* trid-lu.c -- 三項方程式を Gauss の消去法で解く */

/* 三項方程式 (係数行列が三重対角である連立一次方程式のこと) を解く
 *   入力
 *     n: 未知数の個数
 *     al,ad,au: 連立一次方程式の係数行列
 *       (al: 対角線の下側、 ad: 対角線、 au: 対角線の上側)
 *        al[i] = A_{i,i-1}, ad[i] = A_{i,i}, au[i] = A_{i,i+1},
 *        al[0], au[n-1] は意味がない)
 *        b: 連立一次方程式の右辺の既知ベクトル
 *   出力
 *     al,ad,au: 入力した係数行列を LU 分解したもの
 *     b: 連立一次方程式の解
 *   能書き
 *     一度 call すると係数行列を LU 分解したものが返されるので、
 *     以後は同じ係数行列に関する連立一次方程式を解くために、
 *     サブルーチン trisol が使える。
 *   注意
 *     ピボットの選択をしていないので、係数行列が正定値である
 *     などの適切な条件がない場合は結果が保証できない。
 */
trid(n,al,ad,au,b)
int n;
double al[],ad[],au[],b[];
{
  trilu(n,al,ad,au);
  trisol(n,al,ad,au,b);
}

/* 三重対角行列の LU 分解 (pivoting なし) */
trilu(n,al,ad,au)
int n;
double al[],ad[],au[];
{
  int i, nm1 = n - 1;
  /* 前進消去 (forward elimination) */
  for (i = 0; i < nm1; i++) ad[i + 1] -= au[i] * al[i + 1] / ad[i];
}

/* LU 分解済みの三重対角行列を係数に持つ三項方程式を解く */
trisol(n,al,ad,au,b)
int n;
double al[],ad[],au[],b[];
{
  int i, nm1 = n - 1;
  /* 前進消去 (forward elimination) */
  for (i = 0; i < nm1; i++) b[i + 1] -= b[i] * al[i + 1] / ad[i];
  /* 後退代入 (backward substitution) */
  b[nm1] /= ad[nm1];
  for (i = n - 2; i >= 0; i--) b[i] = (b[i] - au[i] * b[i + 1]) / ad[i];
```

```
}
```

関数 trid() の使用例

```
/*
 * test-lu.c --- trid() のテスト
 *   コンパイル＆リンク: cc -o test-lu test-lu.c trid-lu.c
 */

#define NDIM 100
#include <stdio.h>
#include <math.h>

main()
{
  int i, n, nm1;
  double al[NDIM],ad[NDIM],au[NDIM],b[NDIM],x[NDIM];
  n = 10; nm1 = n - 1;
  /* A */
  ad[0] = 2.0; au[0] = -1.0;
  for (i = 1; i < nm1; i++) {
    al[i] = -1.0; ad[i] = 2.0; au[i] = - 1.0;
  }
  al[nm1] = - 1.0; ad[nm1] = 2.0;
  /* x */
  for (i = 0; i < n; i++)
    x[i] = i;
  /* 右辺 */
  b[0] = ad[0] * x[0] + au[0] * x[1];
  for (i = 1; i < nm1; i++)
    b[i] = al[i] * x[i-1] + ad[i] * x[i] + au[i] * x[i+1];
  b[nm1] = al[nm1] * x[nm1-1] + ad[nm1] * x[nm1];
  /* 解く */
  trid(n, al, ad, au, b);
  /* */
  for (i = 0; i < n; i++)
    printf("%f\n", b[i]);
}
```

2.2 Fortran 版

```
* trid-lu.f -- 三項方程式を Gauss の消去法で解く
*****************************************************************
*       三項方程式 (係数行列が三重対角である連立一次方程式のこと) を解く
*         入力
*           n: 未知数の個数
*           al,ad,au: 連立一次方程式の係数行列 A
*             (al: 対角線の下側、 ad: 対角線、 au: 対角線の上側、
*              al(i) = A_{i,i-1}, ad(i) = A_{i,i}, au(i) = A_{i,i+1},
*              al(1), au(n) は意味がない)
*           b: 連立一次方程式の右辺の既知ベクトル
*         出力
*           al,ad,au: 入力した係数行列 A を LU 分解したもの
*           b: 連立一次方程式の解
*         能書き
```

```
*           一度 call すると以後は同じ係数行列に関する連立一次方程式を
*           解くためにサブルーチン trisol が使える。
*         注意
*           ピボットの選択をしていないので、係数行列が正定値である
*           などの適切な条件がない場合は結果が保証できない。
      subroutine trid(n,al,ad,au,b)
      integer n
      real al(*),ad(*),au(*),b(*)
      call trilu(n,al,ad,au)
      call trisol(n,al,ad,au,b)
      end
*****************************************************************
*       三重対角行列の LU 分解 (pivoting なし)
      subroutine trilu(n,al,ad,au)
      integer n
      real al(*),ad(*),au(*)
      integer i
c     前進消去 (forward elimination)
      do i=1,n-1
         ad(i+1) = ad(i+1) - au(i) * al(i+1) / ad(i)
      end do
      end
*****************************************************************
*       LU 分解済みの三重対角行列を係数に持つ三項方程式を解く
      subroutine trisol(n,al,ad,au,b)
      integer n
      real al(*),ad(*),au(*),b(*)
      integer i
c     前進消去 (forward elimination)
      do i=1,n-1
         b(i+1) = b(i+1) - b(i) * al(i+1) / ad(i)
      end do
c     後退代入 (backward substitution)
      b(n) = b(n) / ad(n)
      do i=n-1,1,-1
         b(i) = (b(i) - au(i) * b(i+1)) / ad(i)
      end do
      end
```