Octave 上で LU 分解で連立一次方程式を解く

桂田 祐史

2003 年 8 月 18 日

1 学生の演習から — 解法の巧拙

これは実話なのだが、固有値を逆反復法で求めるプログラム中のループで $y = A^{-1}x$ を計算する方法でかなり拙いことをしてくれて、教師を嘆かせてくれた。

それは `y = inv(A)*x;` とやる方法である。毎回毎回逆行列を再計算させられるとはコンピューターも可愛そうである。せめて

逆行列を使う

```
invA = inv(A); % ループの外
...
y = inv(A) * x;
```

としてもらいたいものだ。

あるいは、「逆行列は計算しない」という教えに忠実に

演算子 \ を使う

```
y = A \ x;
```

とするのが良いかもしれない。行列が疎である場合には、よほど反復回数が多くない限り、上の方法よりも良いはずである。

とは言え、もっとも良い手段は、ループの外で係数行列を LU 分解しておく、次の手順であろう。

LU 分解を使う

```
[L U P] = LU(A); % ループの外
...
y = U \ (L \ (P * x));
```

以上が正しいシナリオのはずだが、これで済まないのが世の中面倒なところである。

2 三角行列係数の方程式の解法

三角行列係数の連立 1 次方程式を解く計算、つまり上三角行列 U や下三角行列 L の逆行列を左からベクトルにかける計算 $x \mapsto L^{-1}x$, $x \mapsto U^{-1}x$ はベクトルの次元 n の 2 乗のオーダー

の計算量で計算できるのは有名な事実である (例えば上三角行列の場合は消去計算抜きの「後退代入」計算で済むから)。

そこで、$y := A^{-1}x$ の計算をする場合に、

```
[L U P] = lu(a);
y = U\(L\(P*x));
```

とする中に現われる二つの \ の計算は高速に出来るはずである。

ところが…私が応援したいと思っている Octave では上三角行列係数の方程式は \ 演算子では高速に解けない。この点 MATLAB はさすがで上下の違いがなく高速に解くことができる。

3 実験による確認

3.1 プログラム

LU 分解の計算時間のテストをする lutest.m

```
% lutest.m
function lutest(n)
  a=rand(n,n);[L U P]=lu(a);x=(1:n)';b=a*x;

  disp('P の掛け算の計算時間');
  tic; c=P*b; toc

  disp('L による割り算の計算時間');
  tic; d=L\c; toc

  disp('U による割り算の計算時間');
  tic; e=U\d; toc

  disp('leftdivu() による割り算の計算時間');
  tic; e2=leftdivu(U,d); toc

  disp(' 精度');
  norm(e-x)
  norm(e2-x)
```

leftdivu.m

```
% leftdivu.m --- 上三角行列 u による割り算 u\z を高速に行う
%    (Octave 用, MATLAB ではかえって遅くなる)
function x=leftdivu(u,z)
  [n,p]=size(u);
  x=zeros(n,1);
  % for k=n:-1:1 とすると動かないので、k=n の処理は別に書く
  x(n)=z(n)/u(n,n);
  for k=n-1:-1:1
    x(k)=(z(k)-u(k,k+1:n)*x(k+1:n))/u(k,k);
  end

```

4 実行結果

Octave での実行結果

```
octave:1> lutest(500)
P の掛け算の計算時間
ans = 0.057695
L による割り算の計算時間
ans = 0.28111
U による割り算の計算時間
ans = 2.5398
leftdivu() による割り算の計算時間
ans = 0.42563
精度
ans =  6.5482e-10
ans =  6.3544e-10
octave:2>
```

$n = 500$ の場合、(\ 演算子を用いて) U で割る演算は L で割る演算の 10 倍近い計算時間がかかっていることが分かる。`leftdivu()` による割り算は、\ 演算子によって L で割る演算の 50% 増し程度の時間で済んでいる (これなら腹が立たないであろう)。

MATLAB での実行結果

```
>> lutest(500)
P の掛け算の計算時間

elapsed_time =

    0.0185

L による割り算の計算時間

elapsed_time =

    0.0390

U による割り算の計算時間

elapsed_time =

    0.0385

leftdivu() による割り算の計算時間

elapsed_time =

    0.1309

精度

ans =

   1.1104e-08


ans =

   1.1090e-08

>>
```

MATLAB の場合は L, U による割り算の実行時間にはほとんど差がない。むしろ `leftdivu()` による割り算の方が遅くなっている (これは \ 演算子が高度に最適化されていると考えるべきであろう)。