静岡理工科大学　理工学部　情報システム学科　卒業論文

「単精度・倍精度と多倍長精度の反復改良法と LU 分解の計算速度比較」

0413046　神尾 亮介　【担当教員：幸谷智紀】

2009 年 2 月 16 日

1. 研究目的

今回、幸谷先生の紹介で、連立 1 次方程式の解き方のひとつである反復改良法を知り、これを学習し卒業研究を通じて理解を深めた。そのためには、線形代数の知識が必要になるが、内容を理解できていなかったので、連立 1 次方程式の基本から復習した。また、反復改良法を解く過程で LU 分解を用いるため、LU 分解も学習した。

最後に、LU 分解を用いた反復改良法を学習した。素朴なアルゴリズムの他に Buttari らの提案する混合精度型のアルゴリズムも学習した。

これらの学習の成果に基づいて、最終目標は、LU 分解と反復改良法のプログラムを作成し、精度が同じ状態でどちらの計算が速いのか実験によって確認するということにした。

2. LU 分解のアルゴリズム

本卒研では、n 次元連立 1 次方程式を $Ax=b$ とおく。この時、A を係数行列、x を解ベクトル、b を定数ベクトルと呼ぶ。また、連立 1 次方程式として、フランク行列 $A=[n-\max(i,j)+1]$ を係数行列に用い、解ベクトル x を $[1,2,\cdots n]^T$ とする。b は Ax として作成した。

今回、反復改良法と比較を行う LU 分解のアルゴリズムは以下のようになっている。

Gauss の消去法の上三角行列にした行列 A の空白部分にも計算したものが実際は残っている。これを用いて、上三角行列 U と下三角行列 L に分解する。

$$\text{行列 } A \quad \Rightarrow \quad LU$$

$$L=\begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a_{32}^{(1)}/a_{22}^{(1)} & 1 \end{bmatrix},\ U=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix}$$

よって、元の連立 1 次方程式は

$$Ax=b \Leftrightarrow (LU)x=b$$

に変換される。なので $L(Ux)=b$ の Ux を y とおいて、

$$\begin{cases} (1)Ly=b \text{を} y \text{について解く。(前進代入)} \\ (2)Ux=y \text{を} x \text{について解く。(後退代入)} \end{cases}$$

として、解ベクトル x が求められる。

3. 多倍長精度の反復改良法アルゴリズム

$$\begin{bmatrix} \text{単精度} \to S\text{桁} \\ \text{倍精度} \to L\text{桁} \end{bmatrix} \quad \text{とする。}$$

1．x_1 を決める。（S 桁で A の LU 分解を行う。）

2．$k=1,2,\cdots$　として以下を反復

3．$r_{k(L)}=b-Ax_{k(L)}$　（L 桁で計算する。）

3.5　$r_{k(S)}:=r_{k(L)}$　（S 桁の残差にする。）

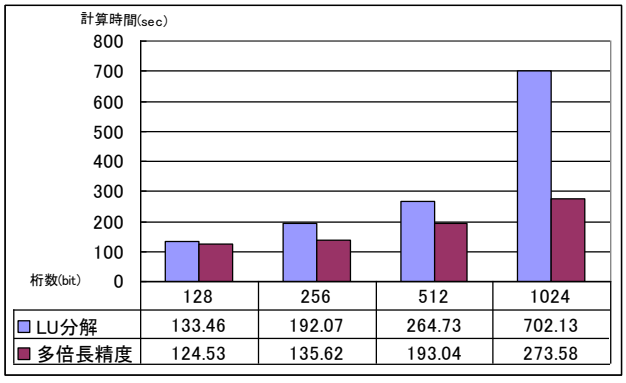
4．$A_{(S)}d_{k(S)}=r_{k(S)}$（S 桁で $d_{k(32)}$ を求める。）

4.5　$d_{k(L)}:=d_{k(S)}$　（S 桁から L 桁へ。）

5．$x_{k+1}=x_{k(L)}+d_{k(L)}$　（L 桁で求める。）

反復改良法では、残差 r_k を高精度で行う必要があるが、Buttari らは 4.の LU 分解の計算は、悪条件でなければ低精度で行うことができ、ループ内の計算を短縮できると指摘している。今回、この Buttari らのアルゴリズムを用いて単・倍精度を多倍長精度に置き換えた上記の反復改良法を自作で作成し、LU 分解と反復改良法の計算速度比較実験を 128,256,512,1024bit の精度で行った。

4. 実験結果



上図は、多倍長精度の LU 分解と反復改良法が次元数 1024 の場合の計算時間結果である。このとき、解の精度はどちらも同じになっていることは確認済みである。他の bit 数も同様に、次元数が上がるほど圧倒的な差で反復改良法が速かった。これにより、Buttari らの提案するアルゴリズムは、元のアルゴリズムより早く計算することが可能であることが証明できた。ただし、反復改良法にとって悪条件な問題の場合に関しては、LU 分解の方が、早くなってしまうこともある。