応用数学工学特論

山本有作

2006 年 5 月 25 日

1　先週の復習

先週の講義では，連立方程式 $A\boldsymbol{x}=\boldsymbol{b}$ をガウスの消去法で解くことと，A をＬＵ分解して，$LU\boldsymbol{x}=\boldsymbol{b}$ として解くことは等価である，ということとＬＵ分解の一意性を証明した．

2　LU 分解のブロック化

LU 分解を行う際，行列の乗算をそのままでやるのではなく，小行列と小行列の乗算の形式にすると，効率的にキャッシュを利用できるため，高速化が可能となる．また行列をブロック化する時は，一般的に，行列乗算 $C := C - AB$ における 3 個のブロック A, B, C が，すべてキャッシュに収まるようにブロックサイズを設定すると最も効率的になる．

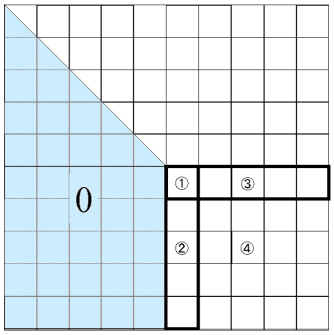


図 1　LU 分解のブロック化

♠ LU 分解プログラムの一例

ガウスの消去法における考え方を，そのままブロック単位に適用したプログラムの一例を紹介する．

---

do K=1, N
　　A_{KK} = $L_{KK}U_{KK}$　　・・・　の領域の計算
　　do I = K+1, N
　　　　L_{IK} = $A_{IK}U_{KK}^{-1}$　　・・・　の領域の計算
　　end do
　　do J = K+1, N
　　　　U_{KJ} = $L_{KK}^{-1}A_{KJ}$　　・・・　の領域の計算
　　end do
　　do I = K+1, N
　　　　do J = K+1, N
　　　　　　A_{IJ} = A_{IJ} - $L_{IK}U_{KJ}$　　・・・　の領域の計算　　→ 演算量の大部分を占める
　　　　end do
　　end do
end do

♣ A'_{IK} = A'_{IK} - $L_{IK}U_{KK}$
　　= A_{IK} - $A_{IK}U_{KK}^{-1}L_{KK}^{-1}A_{KK}$
　　= 0

このプログラムでは　の部分の演算が，全体の演算量の大部分を占める．この時，この演算は行列乗算であるため BLAS3 が適用できる．つまりこのプログラムによりかなりの高速化が見込まれる．実際，Opteron(1.6GHz) 上での測定結果によると，行列をそのまま LU 分解した場合は，プロセッサのピーク性能の 10 %程度，ブロック化して計算した場合は，ピーク性能の 65 %程度を実現できる．

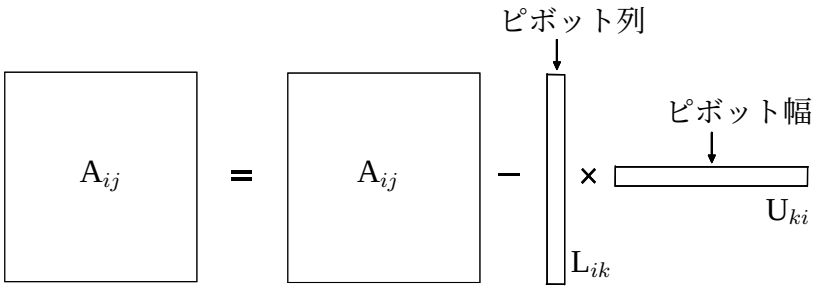


図 2　演算のイメージ:行列の場合

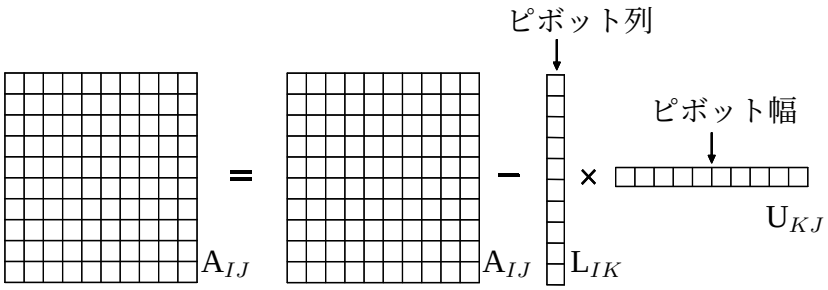


図 3　演算のイメージ：ブロックの場合

3　共有メモリ型計算機上での並列化

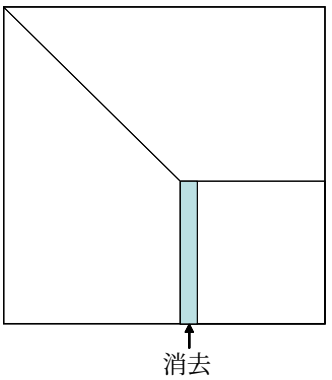
本章では，共有メモリ型計算機における行列計算の並列化について説明する．

- ブロック化アルゴリズムでは，演算の大部分が BLAS3

⇓

並列化された BLAS3 を使えばよい

- ブロック化されていないアルゴリズム




```
do k=1, n
    do i=k+1,n
        l_ik = a_ik / a_kk
        do j=k+1, n
            a_ij = a_ij − l_ik a_kj
        end do
    end do
end do
```

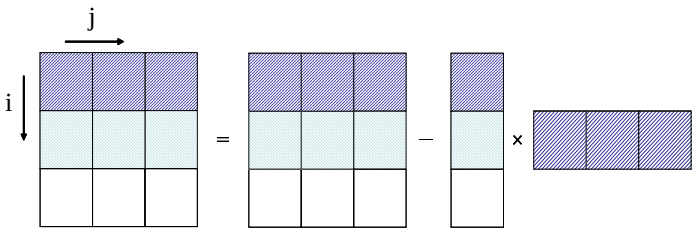


図 4　3 重の do loop（i，j，k）

4　分散メモリ型計算機上での並列化

4.1　データ分割の種類

ブロック分割 CPU が N 台のとき各 CPU に連続した $1/N$ のデータ領域を割り当てる．ガウスの消去法などの計算では最後のほうは右にあるブロックしか使われなくなるため CPU の負荷が均等でなくなる．この分割方法は FFT など全ての最初から最後まで一貫して全てのブロックが均等に使われる演算に有効である．

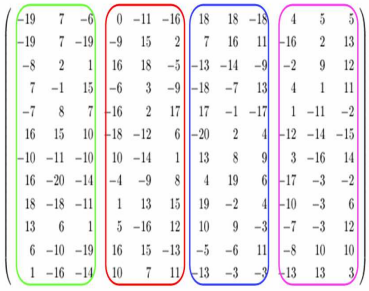


図 5: ブロック列分割

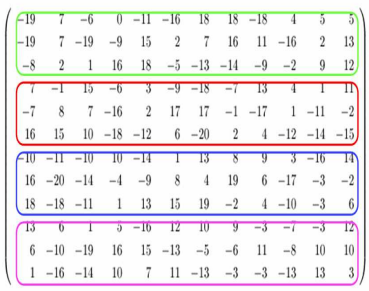


図 6: ブロック行分割

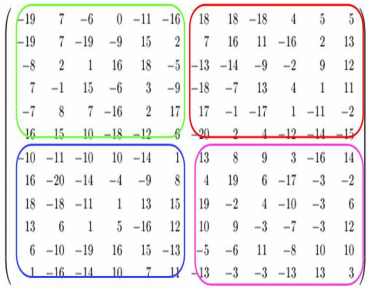
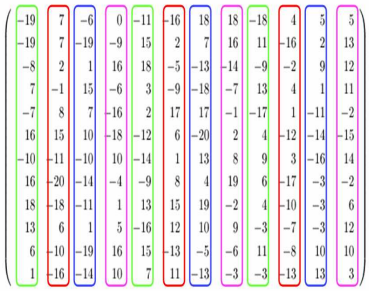


図 7: ブロック双方向分割

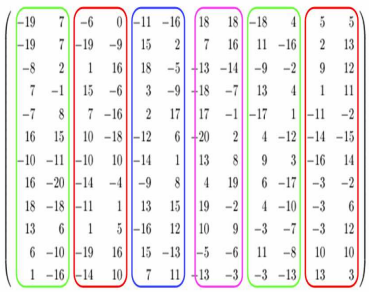
サイクリック分割 一列ずつ各プロセスに循環して割り当てる方法．負荷の均衡は保たれるがブロック化すると 1 ブロックに複数の CPU が割り当てられるため効率的でない．



負荷の均等性	◎
フロック化アルゴリズムとの適合性	×

図 8: サイクリック列分割

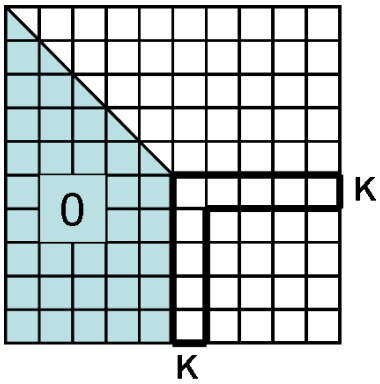
ブロックサイクリック分割 何列かまとめて 1 つのブロックとしてブロック単位でサイクリック分割を行う方法．ある程度負荷分散もできる上に 1 ブロックには 1 CPU しか割り当てられないので効率もよい．



負荷の均等性	○
フロック化アルゴリズムとの適合性	○

図 9: ブロックサイクリック列分割

4.2 ブロックサイクリック分割による並列化



第 K 段の処理

- 第 K 列を持つプロセッサ (MOD(K-1,P)) は
  $A_{KK} = L_{KK}U_{KK}$ と LU 分解
- 第 K 列担当のプロセッサは
  $L_{IK} = A_{IK}U_{KK}^{-1}(I = K+1, \cdots, N)$ を計算する
- 第 K 列担当のプロセッサは
  $L_{IK}(I = K, \cdots, N)$ を全プロセッサに送信（ブロードキャスト）
- 各プロセッサは自分の担当する列について
  $U_{KJ} = L_{KK}^{-1}A_{KJ}$ を計算する
  L_{KK}^{-1} は第 K 列を持っているプロセッサから受信するブロック
  A_{KJ} はあらかじめ自分で持っているブロック
- 各プロセッサは自分の担当する列について
  $A_{IJ} = A_{IJ} - L_{IK}U_{KJ}$ を計算