

# LU分解法(2)

---

東京大学情報基盤センター 准教授 埜 敏博

2019年12月17日(火) 2限 10:25 – 12:10



レポートおよびコンテスト課題  
(締切:  
2020年2月3日(月)24時 厳守

# 講義日程(工学部共通科目)

- ~~1.~~ ~~9月24日: ガイダンス~~
- ~~2.~~ ~~10月1日~~
  - ~~● 並列数値処理の基本演算(座学)~~
- ~~3.~~ ~~10月8日: スパコン利用開始~~
  - ~~● ログイン作業、テストプログラム実行~~
- ~~4.~~ ~~10月15日~~
  - ~~● 高性能プログラミング技法の基礎1  
(階層メモリ、ループアンローリング)~~
- ~~5.~~ ~~10月29日~~
  - ~~● 高性能プログラミング技法の基礎2  
(キャッシュブロック化)~~
- ~~6.~~ ~~11月5日~~
  - ~~● 行列ベクトル積の並列化~~
- ~~7.~~ ~~11月12日~~
  - ~~● ベキ乗法の並列化~~
- ~~8.~~ ~~11月26日~~
  - ~~● 行列-行列積の並列化(1)~~
- ~~9.~~ ~~12月3日~~
  - ~~● 行列-行列積の並列化(2)~~
- ~~10.~~ ~~12月10日~~
  - ~~● LU分解法(1)~~
  - ~~● コンテスト課題発表~~
11. 12月17日
  - LU分解法(2)
12. 1月7日
  - LU分解法(3)、非同期通信
13. 1月14日
  - RB-Hお試し、研究紹介他

# LU分解法(中級レベル以上)の演習日程

並列化が難しいので、3週間確保してあります。

## 1. 今週

- 講義(知識、アルゴリズムの理解)
- 並列化の検討

## 2. 来週

- LU分解法の逐次アルゴリズムの説明
- LU分解法の並列化実習(1)

## 3. 再来週

- LU分解法の並列化実習(2)

# 講義の流れ

1. LU分解法の  
逐次アルゴリズム解説
2. 並列化実習のつづき

# LU分解並列化のヒント(2)

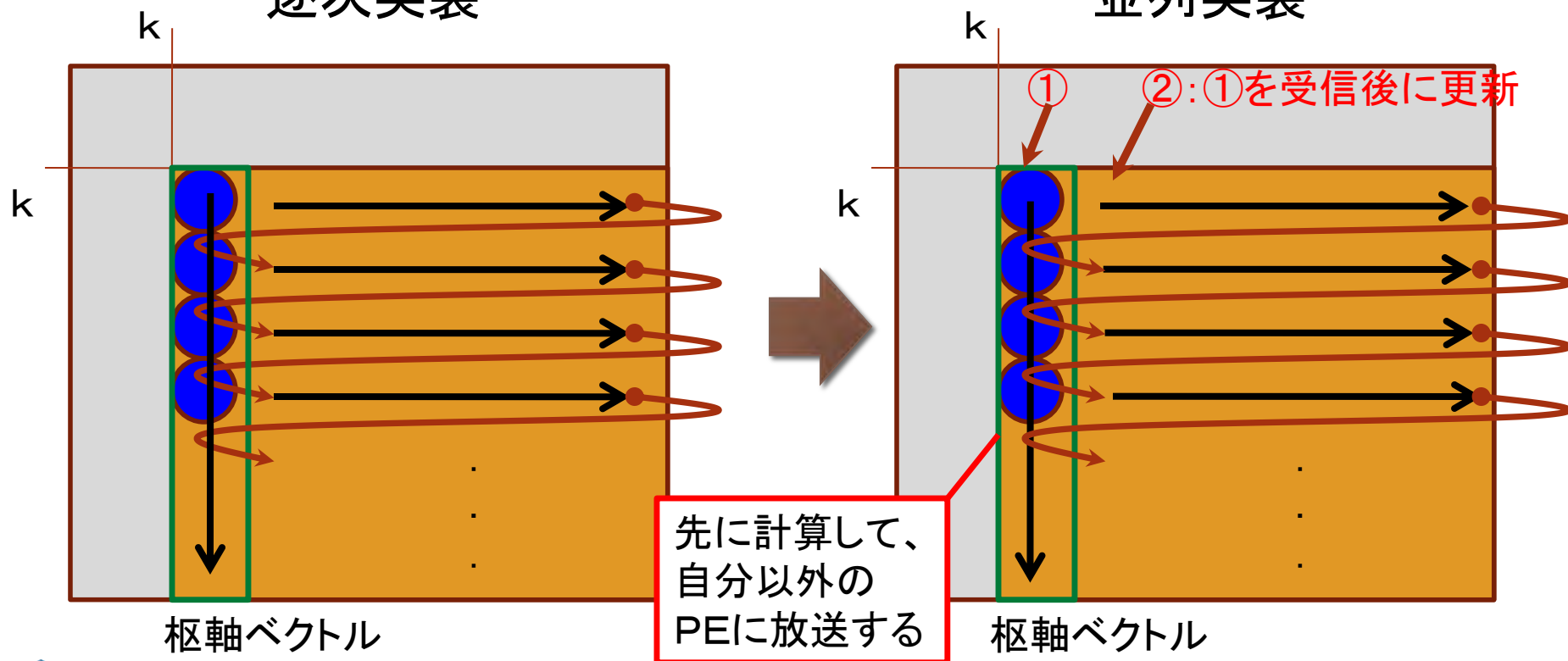
---

# LU分解部分並列化の方針(C言語)

- LU分解部分では、枢軸ベクトルをもつPEが先に計算し(図の①)、それをその他のPEに放送する必要があります。

逐次実装

並列実装



枢軸ベクトル

枢軸ベクトル

# LU分解部分のプログラム解説(C言語)

```
for (k=0; k<n; k++) {  
  
    dtemp = 1.0 / A[k][k];  
    for (i=k+1; i<n; i++) {  
        A[i][k] = A[i][k]*dtemp;  
    }  
  
    for (j=k+1; j<n; j++) {  
        dtemp = A[j][k];  
        for (i=k+1; i<n; i++) {  
            A[j][i] = A[j][i] - A[k][i]*dtemp;  
        }  
    }  
}
```

基本行(k行)の移動ループ

基本行からの係数を計算し、  
枢軸ベクトルを求めている  
部分(①)

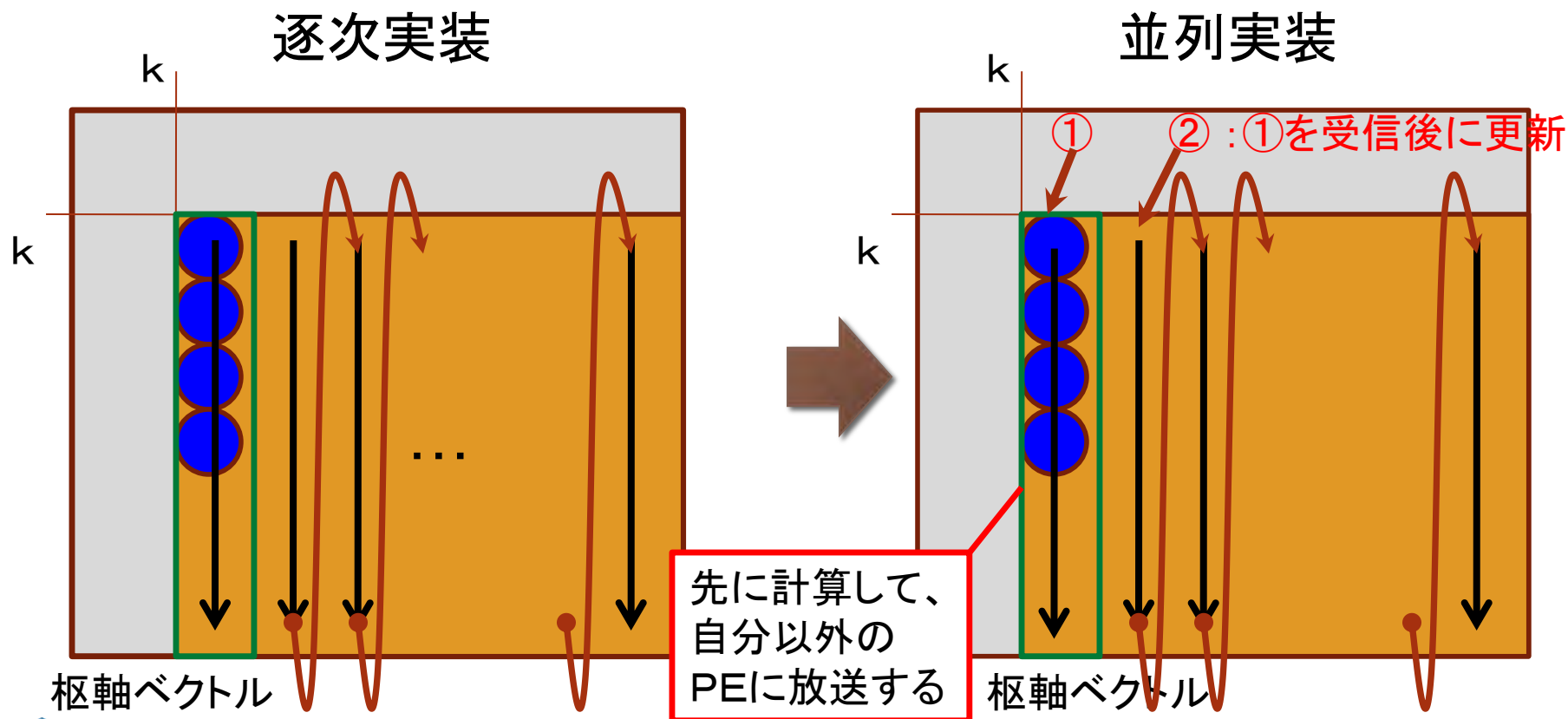
枢軸ベクトルを参照

枢軸ベクトルを参照しつつ、  
消去を行っている部分(②)

基本行を参照

# LU分解部分並列化の方針 (Fortran言語)

- LU分解部分では、枢軸ベクトルをもつPEが先に計算し (図の①)、それをその他のPEに放送する必要があります。





# LU分解部分のプログラム解説 (Fortran言語)

```
do k=1, n
```

基本行(k行)の移動ループ

```
  dtemp = 1.0d0 / A(k, k)
  do i=k+1, n
    A(i, k) = A(i, k)*dtemp
  enddo
```

基本行からの係数を計算し、  
枢軸ベクトルを求めている  
部分(①)

```
  do j=k+1, n
    dtemp = A(k, j)
    do i=k+1, n
      A(i, j) = A(i, j) - dtemp * A(i, k)
    enddo
  enddo
```

基本行を参照

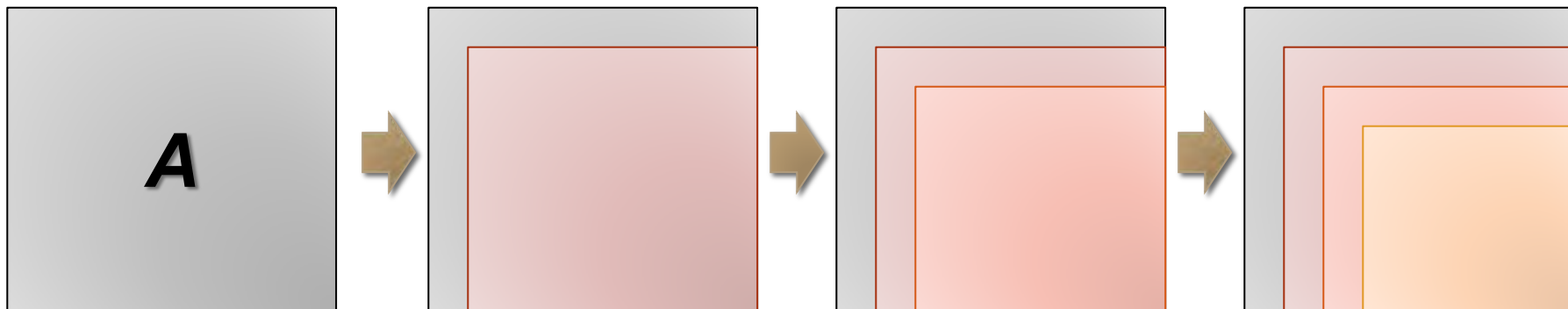
枢軸ベクトルを参照しつつ、  
消去を行っている部分(②)

枢軸ベクトルを参照

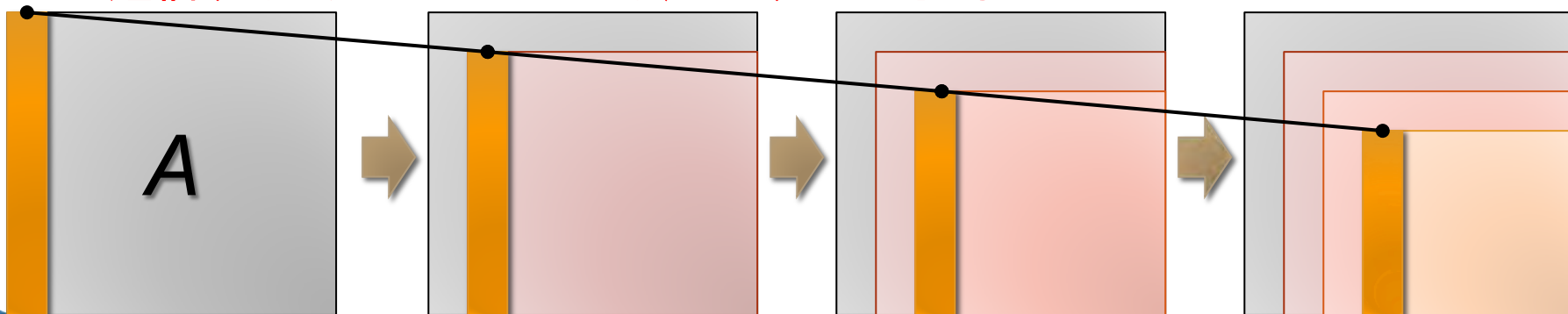
```
enddo
```

# LU分解のアルゴリズムの特徴

- LU分解は、更新範囲が1つずつ小さくなっていく



- 枢軸ベクトルも、1つずつ小さくなっていく
  - 送信するメッセージサイズも、1つずつ小さくなっていく

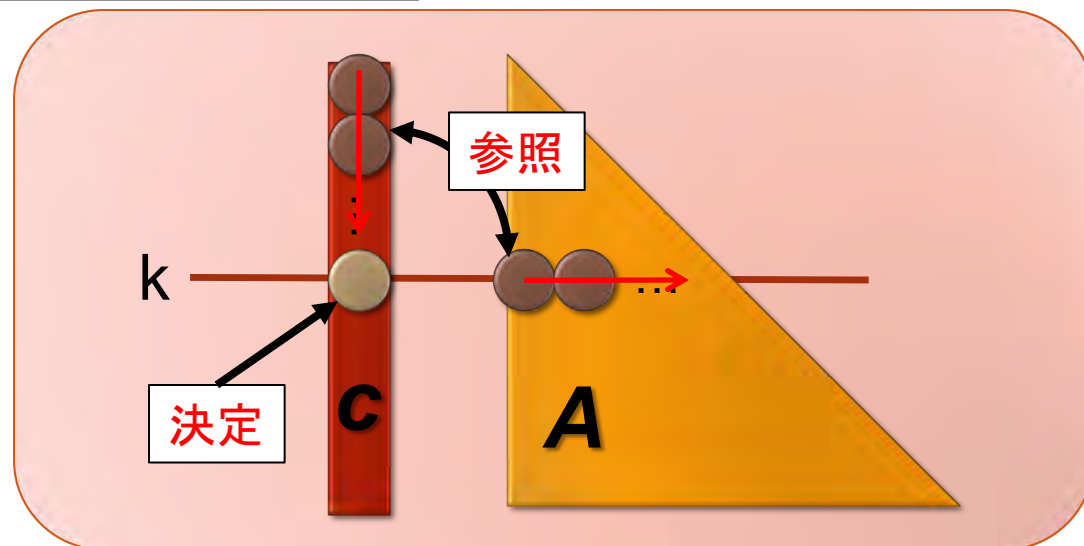


# 前進代入部分のプログラム解説(C言語)

```
for (k=0; k<n; k++) {  
    c[k] = b[k];  
    for (j=0; j<k; j++) {  
        c[k] -= A[k][j]*c[j];  
    }  
}
```

ベクトルcの値を決定する  
要素(k要素)の移動ループ

k要素より前のベクトルcの要素  
を参照して、k要素の値を決定

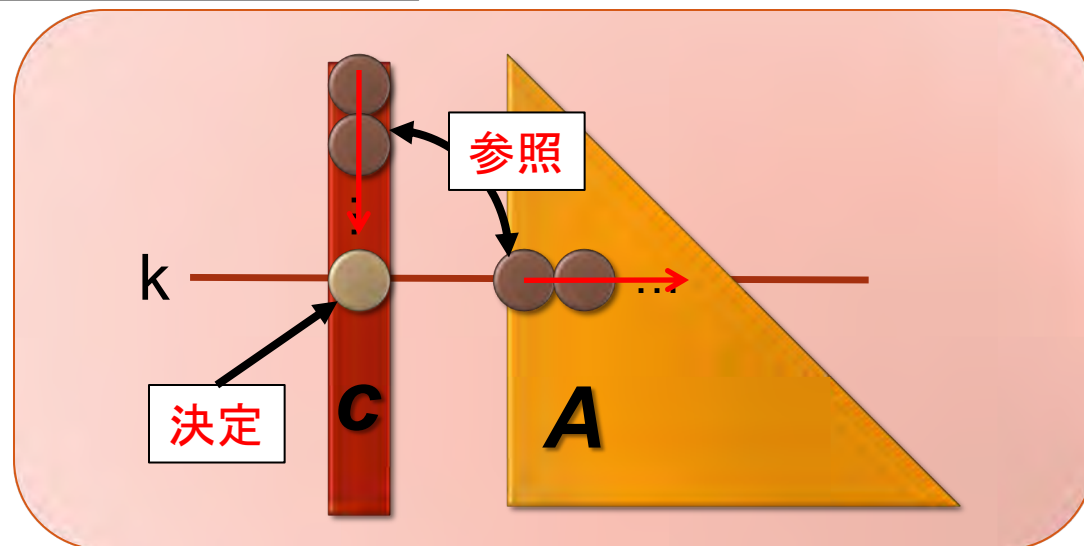


# 前進代入部分のプログラム解説(Fortran言語)

```
do k=1, n
  c(k) = b(k)
  do j=1, k-1
    c(k) = c(k) - A(k, j)*c(j)
  enddo
enddo
```

ベクトルcの値を決定する  
要素(k要素)の移動ループ

k要素より前のベクトルcの要素  
を参照して、k要素の値を決定



# LU分解の並列化方法の確認(再掲)

1. LU分解部分のみ並列化する
2. 行列Aを表示し、逐次の答え(LuAc.dat)と一致しているか確認する
3. 前進代入部分を並列化する
4. ベクトルcを表示し、逐次の答え(LuAc.dat)と一致しているか確認する
5. 後退代入部分を並列化する
6. ベクトルxを表示し、逐次の答え(すべて1)と一致しているか確認する

**鉄則: 一度にすべて並列化しても、まず動かない。  
地道に並列化していくのが完成への早道。**

# 来週へつづく

---

## LU分解(3)