

LU分解法 (2)

東京大学情報基盤センター 准教授 片桐孝洋

2015年6月30日(火) 10:25-12:10

スパコンプログラミング(1)、(I)



東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

講義日程（工学部共通科目）

レポートおよびコンテスト課題
(締切:
2015年8月3日(月)24時 厳守

~~1. 4月14日: ガイダンス~~

~~2. 4月21日~~

- ~~● 並列数値処理の基本演算(座学)~~

~~3. 4月28日: 座学のみ~~

- ~~● ソフトウェア自動チューニング~~
- ~~● 非同期通信~~

~~4. 5月12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~~

~~5. 5月19日~~

- ~~● 高性能演算技法1
(ループアンローリング)~~

~~6. 6月2日(8:30-10:15)~~

- ~~● 高性能演算技法2
(キャッシュブロック化)~~

~~7. 6月2日(10:25-12:10)~~

- ~~● 行列ベクトル積の並列化~~

~~8. 6月9日(8:30-10:15)~~

~~★大演習室2~~

- ~~● ベキ乗法の並列化~~

~~9. 6月9日(10:25-12:10)~~

- ~~● 行列・行列積の並列化(1)~~

~~10. 6月16日~~

- ~~● 行列・行列積の並列化(2)~~

~~11. 6月23日~~

- ~~● LU分解法(1)~~
- ~~● コンテスト課題発表~~

10. 6月30日

- LU分解法(2)

11. 7月7日

- LU分解法(3)

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）

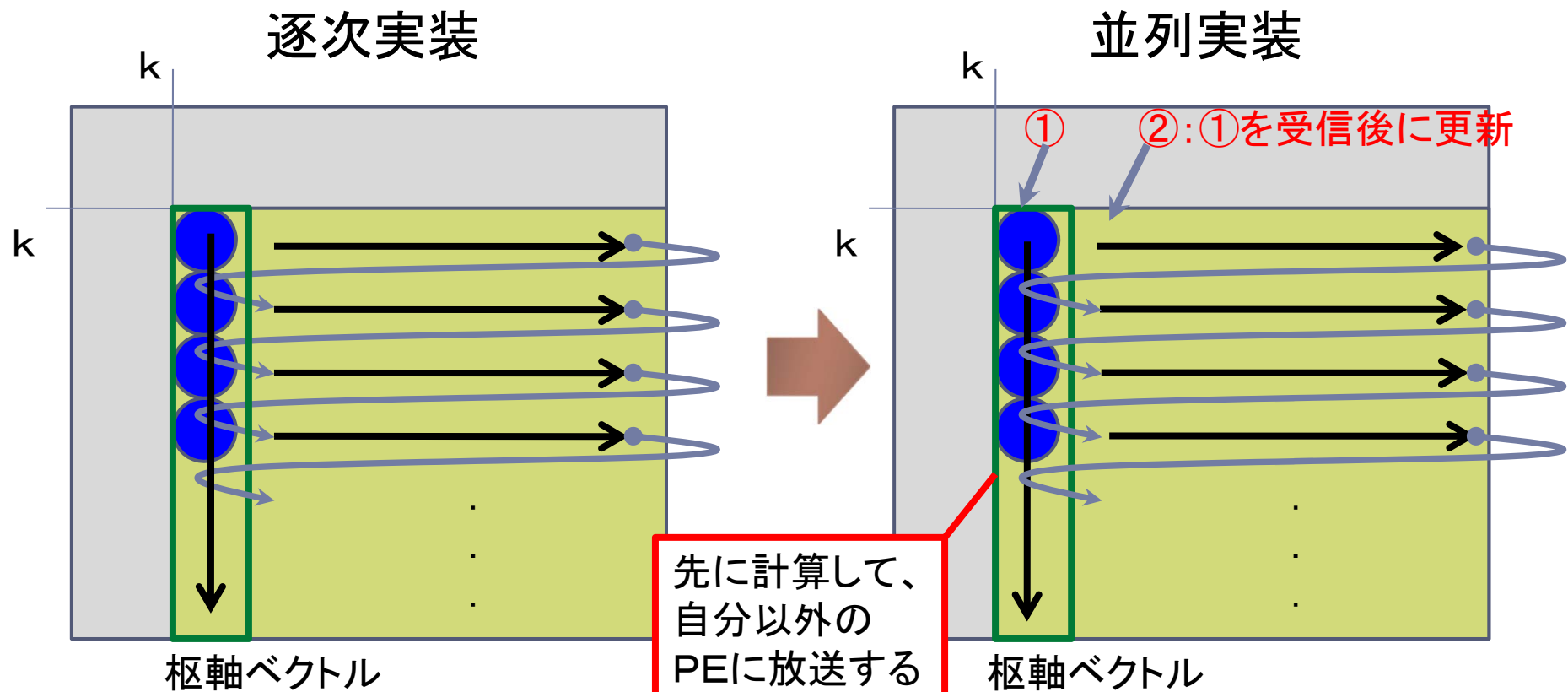
講義の流れ

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。



LU分解部分のプログラム解説 (C言語)

```
for (k=0; k<n; k++) {
```

基本行(k行)の移動ループ

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

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)

```
    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;  
        }  
    }  
}
```

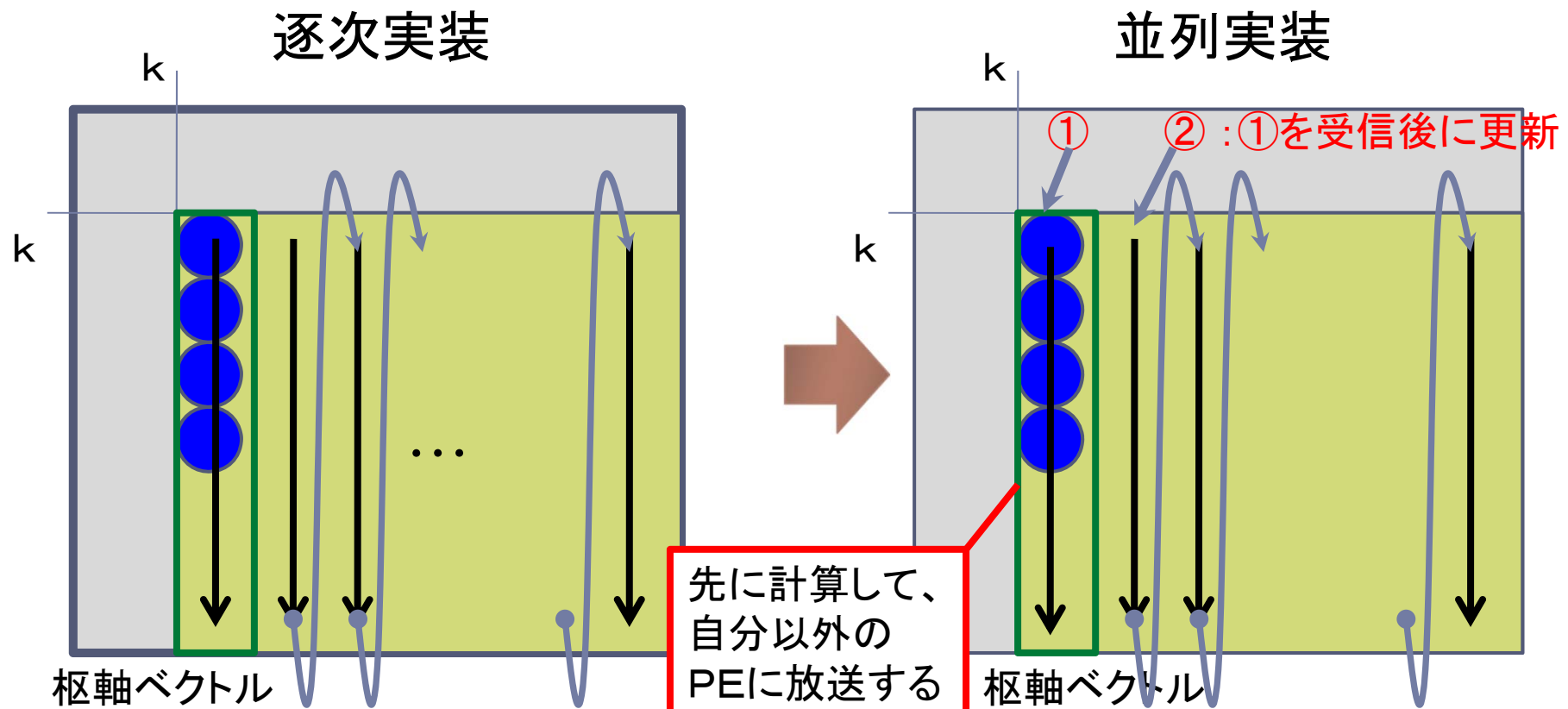
枢軸ベクトルを参照

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

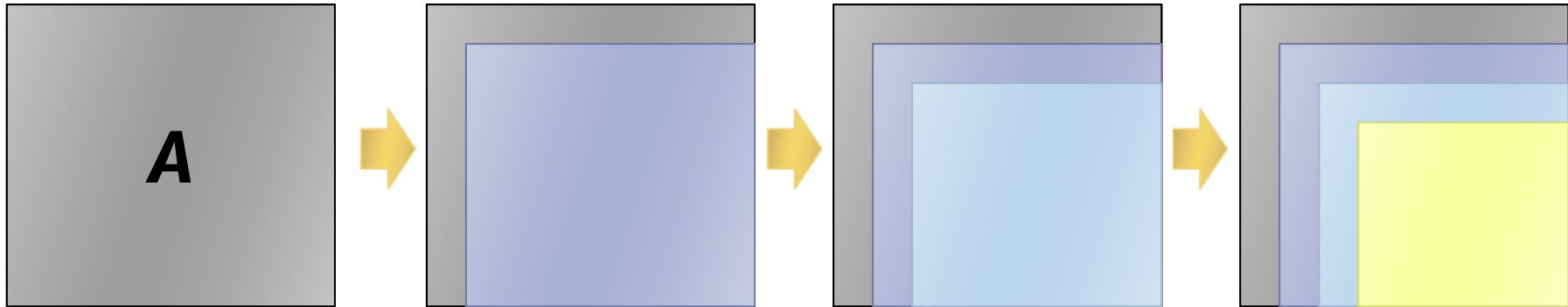
枢軸ベクトルを参照しつつ、
消去を行っている部分(②)

枢軸ベクトルを参照

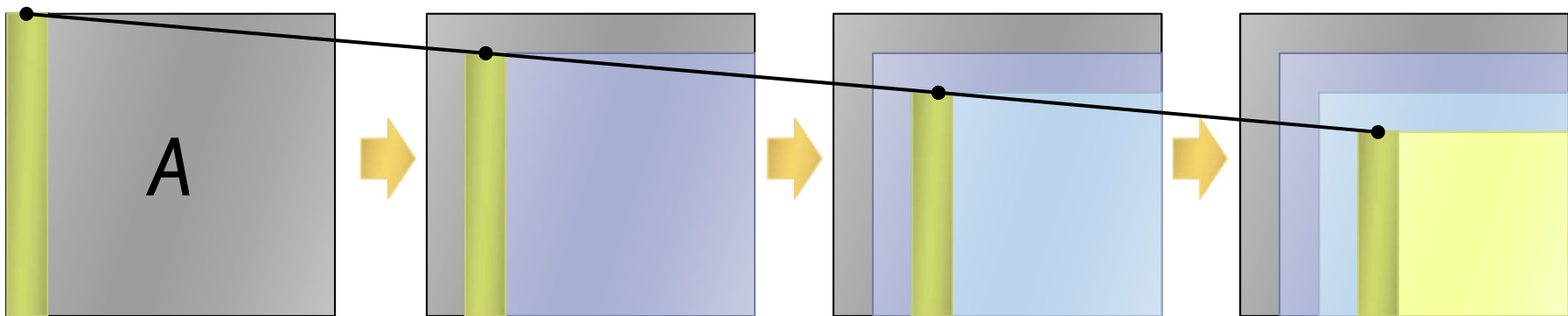
```
enddo
```

LU分解のアルゴリズムの特徴

- ▶ LU分解は、更新範囲が1つづ小さくなっていく



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

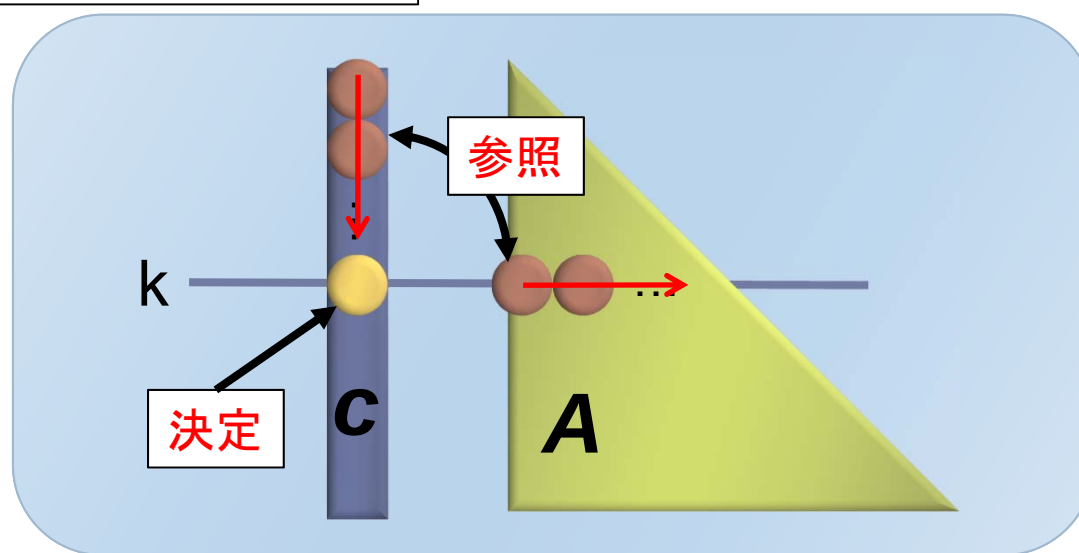


前進代入部分のプログラム解説（C言語）

```
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

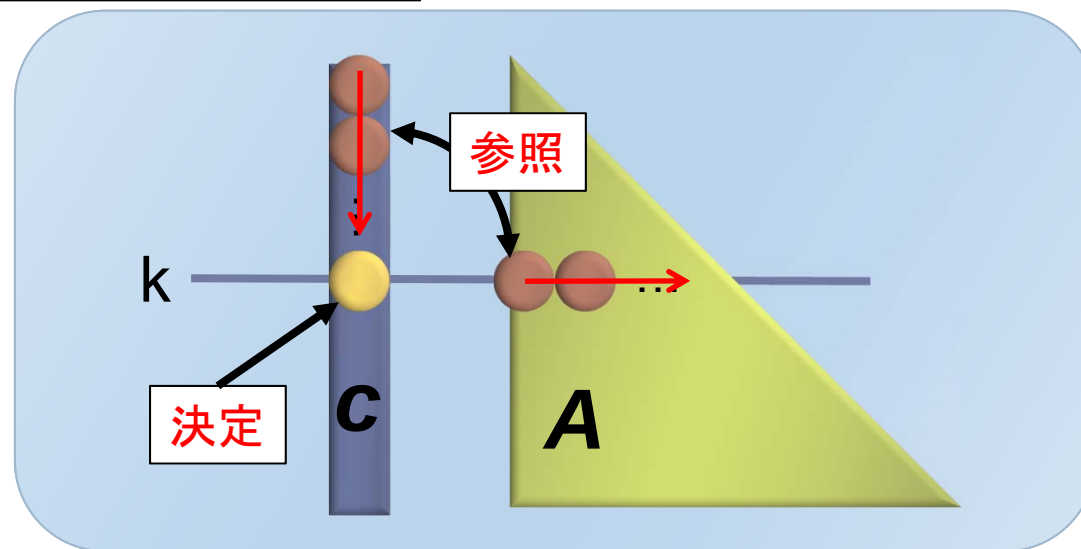


前進代入部分のプログラム解説 (Fortran言語)

```
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. 行列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. 行列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)