

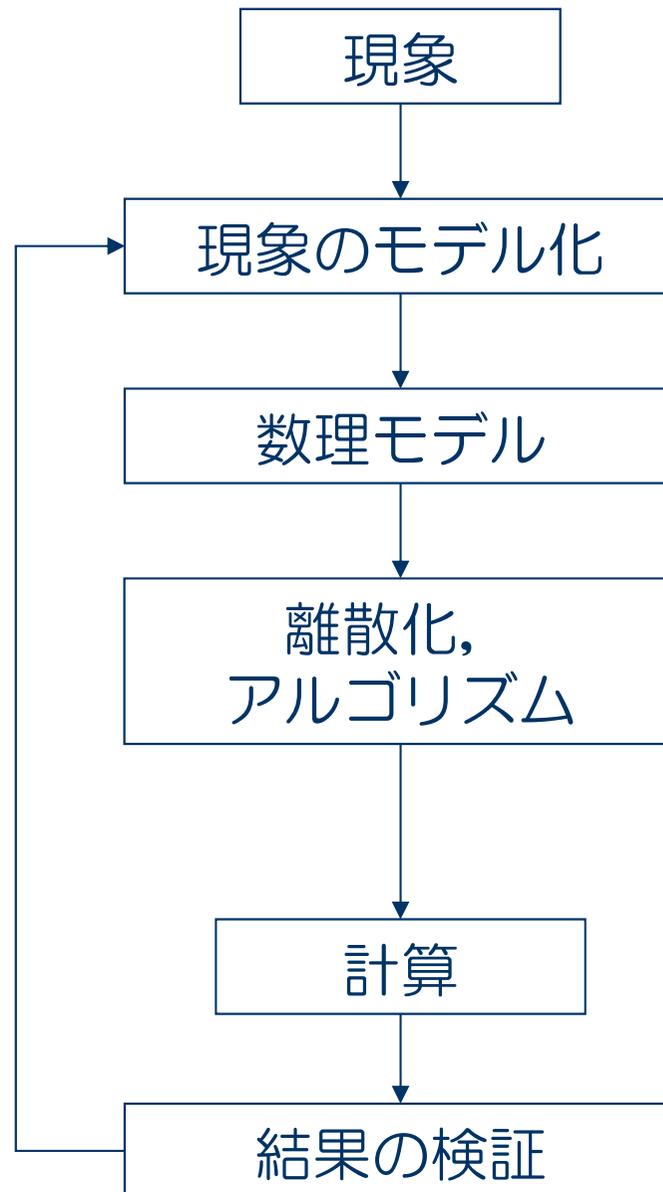
講義 1 : 離散化と並列化

2014年3月5日

神戸大学大学院システム情報学研究科計算科学専攻
横川三津夫

講義の内容

- 計算機シミュレーション
- 非定常問題の時間微分項の離散化
- 定常問題
- 線型方程式の反復解法
- 実行時間の評価



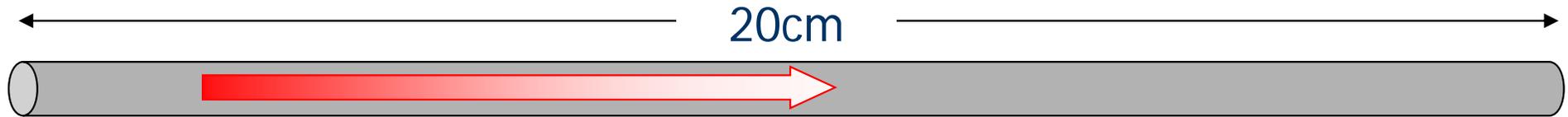
偏微分方程式など

離散化（計算格子も選択）

- ・有限差分法, 有限要素法, スペクトル法など

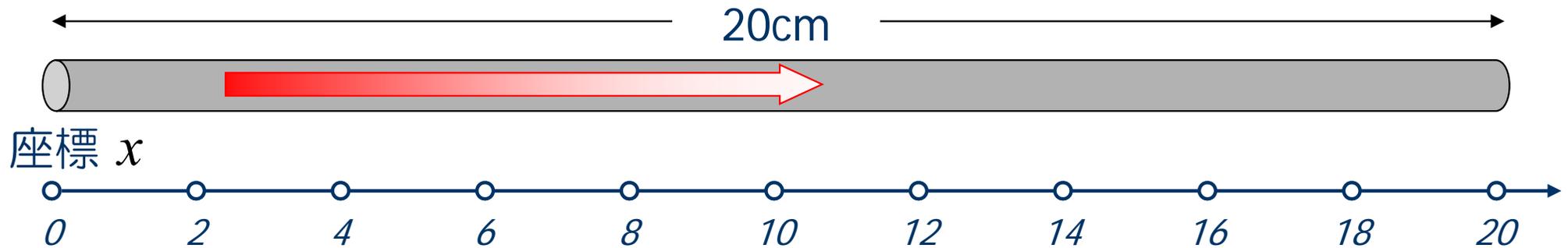
- ・足し算, 掛け算だけの式で記述

プログラム, コンピュータ



- 長さ20cmのアルミニウム棒の左端を温めたとき，1分後の右端の温度は何度になっているのか？ 棒の温度がどんな風に変化するかわかりたい。
 - 最初の棒の温度は20°C（初期状態）
 - 左端を加熱し，温度100°Cにする。
 - 熱伝導度 $9.7 \times 10^{-5} \text{m}^2/\text{s}$
- 長さ方向の温度変化だけを見れば良い。 ⇒ 1次元問題

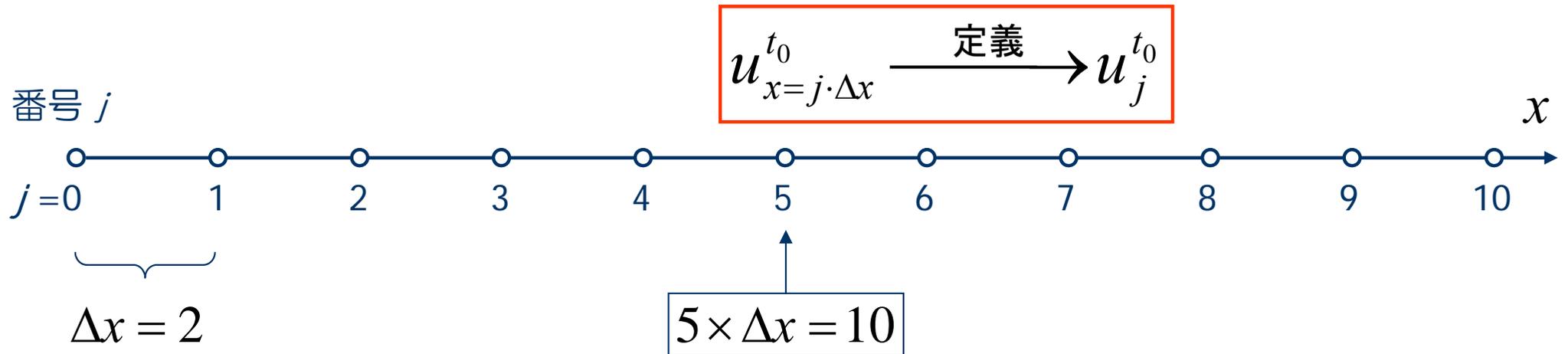
モデル化（1次元非定常熱伝導問題）



- 長さ20 cmのアルミニウム棒の左端を温めたとき，1分後の右端の温度は何度になっているのか？ 棒の温度がどんな風に変化するかを知りたい。
 - 最初の棒の温度は20°C（初期状態）
 - 左端を加熱し，温度100°Cにする。
 - 熱伝導度 $9.7 \times 10^{-5} \text{m}^2/\text{s}$
- 長さ方向の温度変化だけを見れば良い。 ⇒ 1次元問題
- 1次元の数理モデル（偏微分方程式）で表現
 - 1次元非定常熱伝導方程式

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2}$$

離散化：連続な量を有限個の点で表現すること



$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2}$$



時間微分：前進オイラー法
空間微分：有限差分法

$$\frac{u_j^{t_0 + \Delta t} - u_j^{t_0}}{\Delta t} = a \frac{u_{j-1}^{t_0} - 2u_j^{t_0} + u_{j+1}^{t_0}}{(\Delta x)^2}$$

$$u_j^{t_0 + \Delta t} = u_j^{t_0} - r(u_{j-1}^{t_0} - 2u_j^{t_0} + u_{j+1}^{t_0})$$

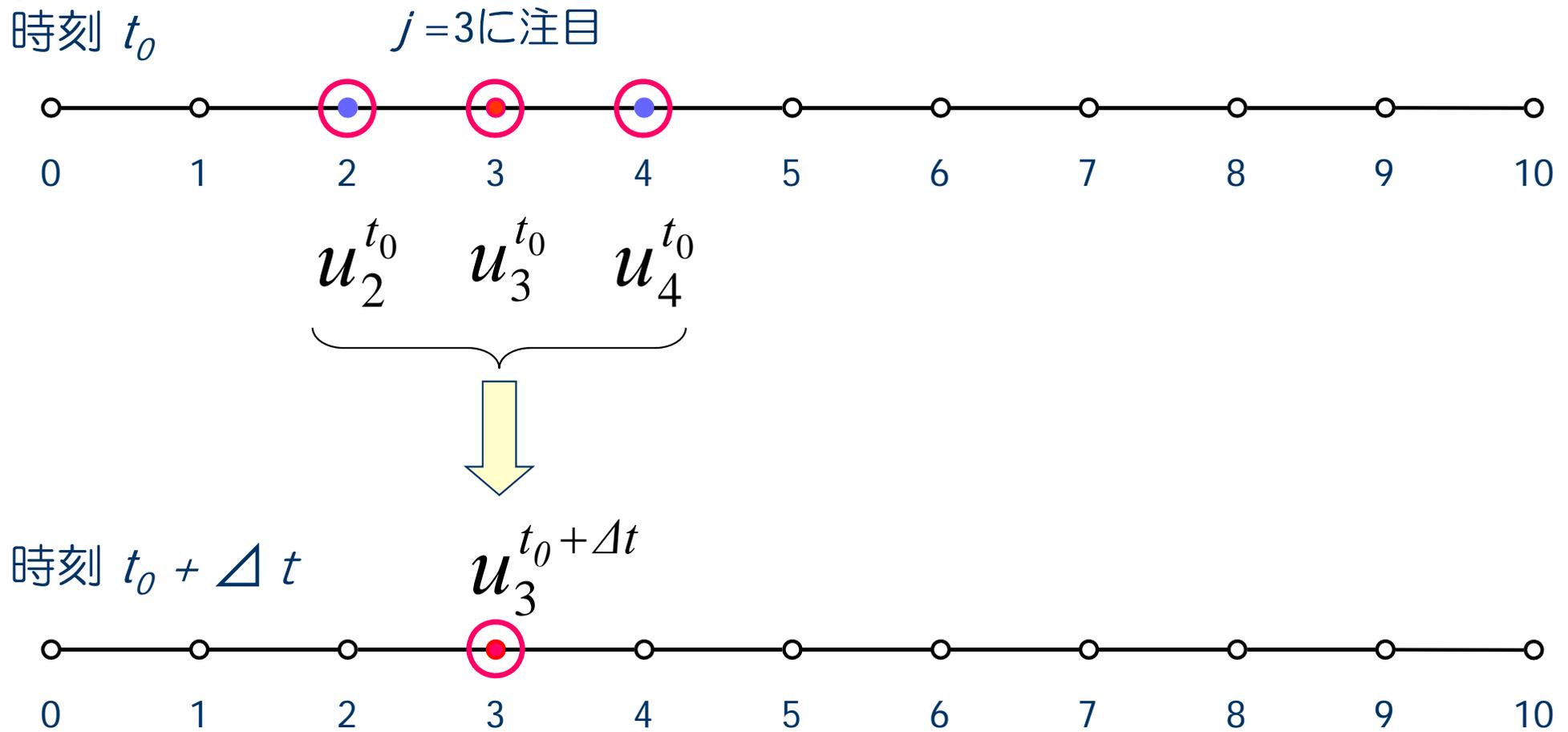
$$u_j^{t_0 + \Delta t} = u_j^{t_0} - \frac{a\Delta t}{(\Delta x)^2} (u_{j-1}^{t_0} - 2u_j^{t_0} + u_{j+1}^{t_0})$$

\equiv

時刻 t_0 の温度から Δt 経過した後の温度を求める式

$$u_j^{t_0+\Delta t} = u_j^{t_0} - r(u_{j-1}^{t_0} - 2u_j^{t_0} + u_{j+1}^{t_0})$$

乗算： 2回
加減算： 3回



プログラミング言語FORTRAN77によるプログラム

格子分割数 20

格子間隔 $\Delta x=1\text{cm}$

時間間隔 $\Delta t=0.5\text{秒}$

熱伝導度 $a=9.7 \times 10^{-5}$

初期温度を設定する。

左端を100度にする。

120×0.5秒=60秒まで繰り返し

各格子点の値を求める。

結果を書き出す。

```
real*8  u_new(0:20), u_old(0:20)
```

```
n  = 20
```

```
dx = 0.2/real(n)
```

```
dt = 0.5
```

```
a  = 9.7e-5
```

```
r  = a*dt/(dx*dx)
```

```
do j = 0, 20
```

```
  u_old(j) = 20.0
```

```
enddo
```

```
u_old(0) = 100.0
```

```
do it = 1, 120
```

```
  do j = 1, 19
```

```
    u_new(j) = (1.0-2.0*r)*u_old(j) + r*(u_old(j-1)+u_old(j+1))
```

```
  enddo
```

```
  do j = 1, 19
```

```
    u_old(j) = u_new(j)
```

```
  enddo
```

```
  u_old[20] = u_old[19]
```

```
  write(6,*) dt*it, (u_old(j), j=0,20)
```

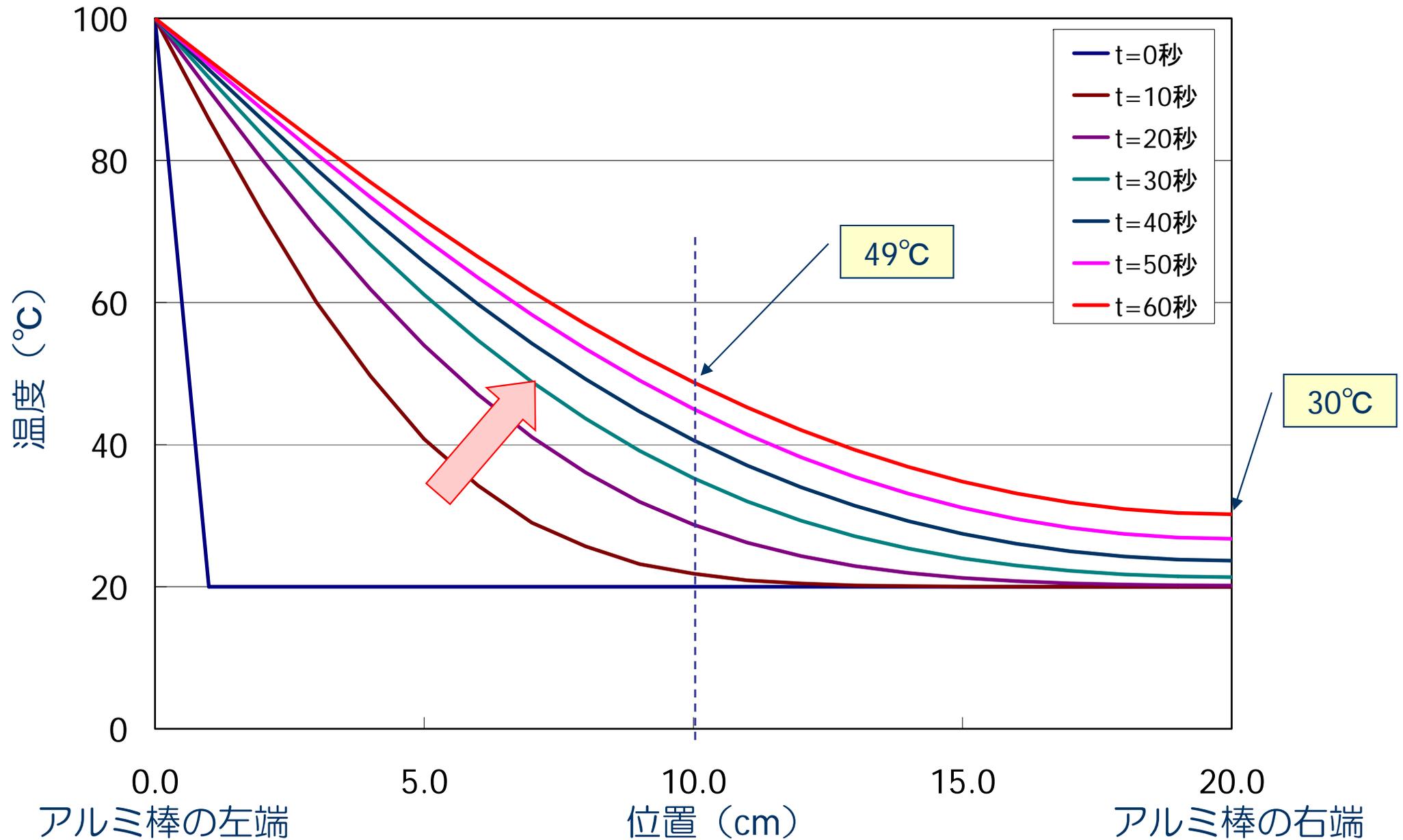
```
enddo
```

```
stop
```

```
end
```

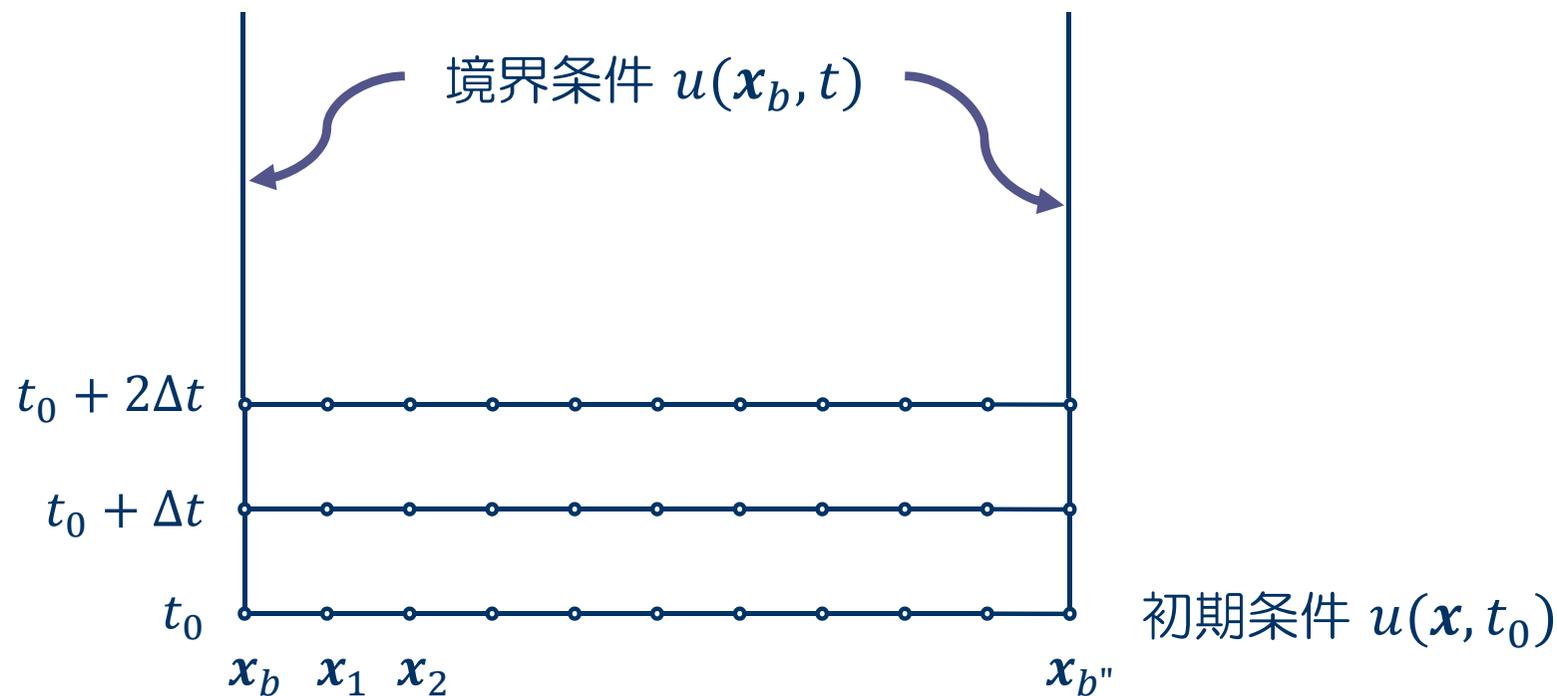
$$u_j^{t_0+\Delta t} = u_j^{t_0} - r(u_{j-1}^{t_0} - 2u_j^{t_0} + u_{j+1}^{t_0})$$


0秒から60秒までの温度変化（10秒間隔）



$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \mathbf{M}(u) \quad : \quad \mathbf{M}(u) \text{は空間の微分演算子を含む項}$$

初期条件 $u(\mathbf{x}, t_0)$ ，境界条件 $u(\mathbf{x}_b, t)$ が与えられる。



時間発展（非定常）方程式に対する時間微分項の離散化

$$u(\mathbf{x}, t_n) \triangleq u^n, \quad t_n \triangleq t_0 + n \cdot \Delta t, \quad \Delta t \text{は時間刻み幅（時間ステップ）}$$

	陽解法	陰解法
	過去の情報だけで、次の時刻 t_{n+1} の値を求める方法	次の時刻 t_{n+1} の値を求めるために、空間の微分式 $\mathbf{M}(u^{n+1})$ が成立するとして求める方法
単段法 時刻 t_n の値だけを用いて、時刻 t_{n+1} の値を求める方法	<ul style="list-style-type: none"> 前進オイラー法 $\frac{u^{n+1} - u^n}{\Delta t} = \mathbf{M}(u^n)$ ルンゲ・クッタ法など 	<ul style="list-style-type: none"> 後退オイラー法など $\frac{u^{n+1} - u^n}{\Delta t} = \mathbf{M}(u^{n+1})$
多段法 複数の過去の時刻 t_n, t_{n-1}, \dots の値を用いて、時刻 t_{n+1} の値を求める方法	<ul style="list-style-type: none"> アダムス-バッシュフォース法 リーブ-フロッグ法など $\frac{u^{n+1} - u^{n-1}}{2\Delta t} = \mathbf{M}(u^n)$ 	<ul style="list-style-type: none"> アダムス-モルトン法など

熱伝導方程式

$$\frac{\partial u(x,t)}{\partial t} = \alpha \nabla^2 u$$

の定常状態の解（ $t \rightarrow \infty$ の時の解）はどうなるか？



$$\nabla^2 u = 0 \quad \& \quad \text{境界条件}$$

【境界条件】

- Dirichlet（ディリクレ）問題：対象となる領域の境界上で解の値が指定される場合
- Neumann（ノイマン）問題：境界上で法線方向の微分が指定されている場合

- 2次元正方形領域 $[0,1] \times [0,1]$ での熱伝導問題

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

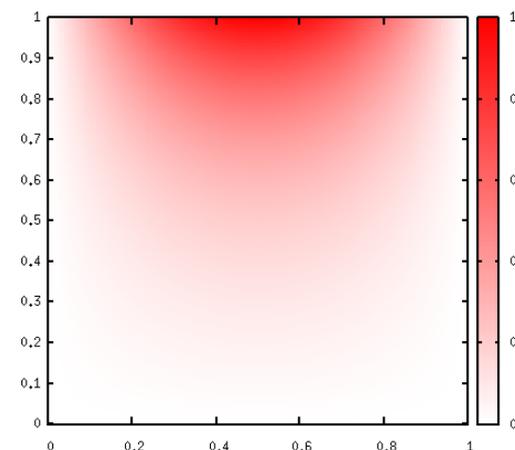
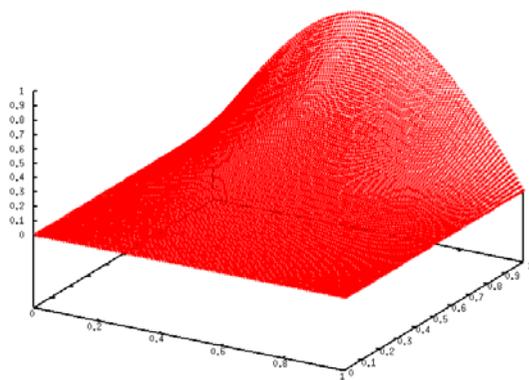
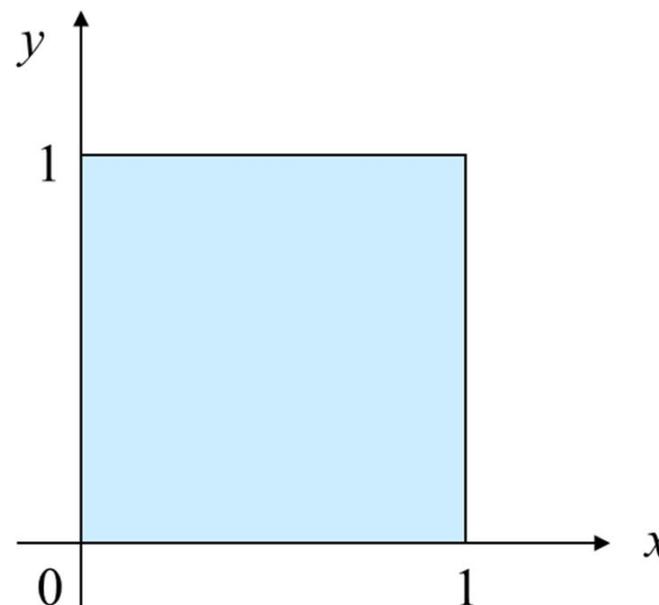
- 境界条件

$$u(0, y) = 0 \quad (0 \leq y \leq 1)$$

$$u(1, y) = 0 \quad (0 \leq y \leq 1)$$

$$u(x, 0) = 0 \quad (0 \leq x \leq 1)$$

$$u(x, 1) = \sin(\pi x) \quad (0 \leq x \leq 1)$$



差分式のテイラー級数展開からの導出 (2変数)

$$u(x+h, y) = u(x, y) + u_x(x, y)h + \frac{u_{xx}(x, y)}{2!}h^2 + \dots$$

$$u(x-h, y) = u(x, y) - u_x(x, y)h + \frac{u_{xx}(x, y)}{2!}h^2 - \dots$$

これらの式から. . .

$$u_x(x, y) = \frac{u(x+h, y) - u(x, y)}{h} + O(h)$$

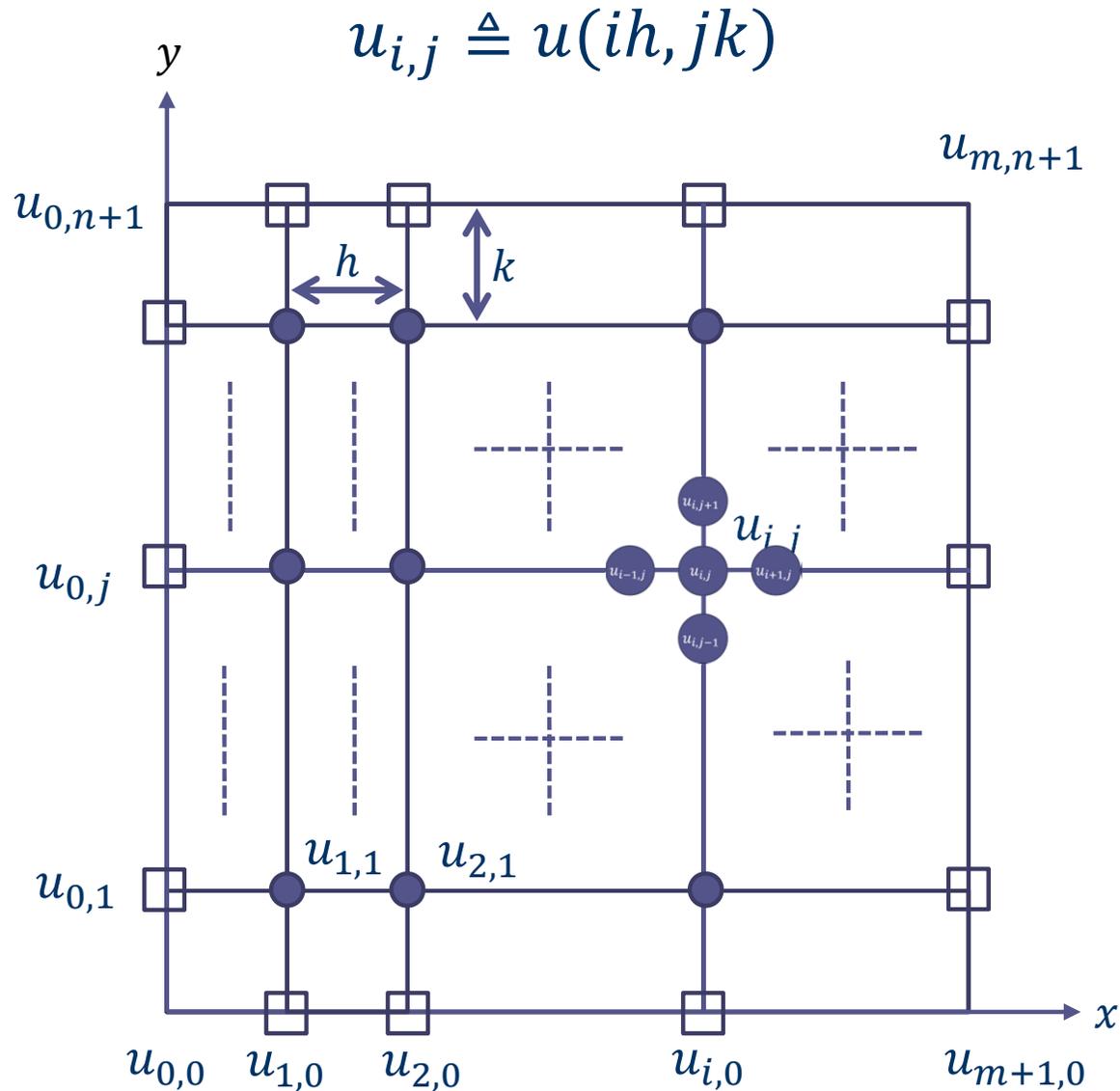
$$u_{xx}(x, y) = \frac{u(x-h, y) - 2u(x, y) + u(x+h, y)}{h^2} + O(h^2)$$

$$u_y(x, y) = \frac{u(x, y+k) - u(x, y)}{k} + O(k)$$

$$u_{yy}(x, y) = \frac{u(x, y-k) - 2u(x, y) + u(x, y+k)}{k^2} + O(k^2)$$

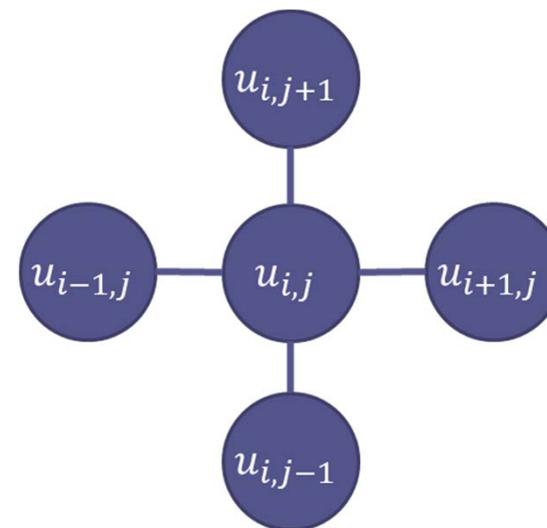
2次元定常熱伝導問題の離散化 (その1)

x 方向を $(m + 1)$ 等分 : $h = 1/(m + 1)$, y 方向を $(n + 1)$ 等分 : $k = 1/(n + 1)$



- 境界条件として既知の値
- 内部の格子点 (未知数)

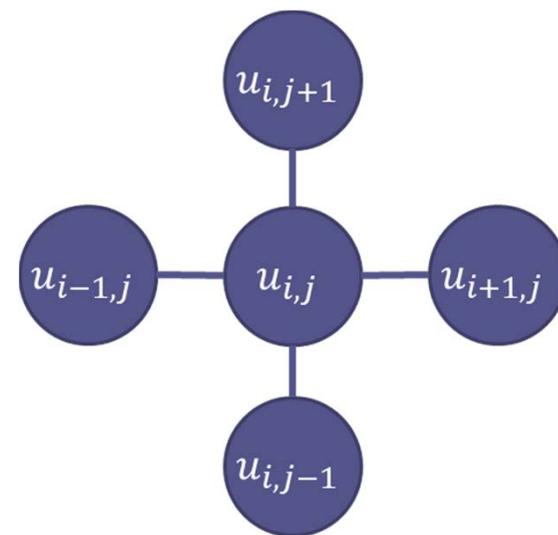
内部の格子点での関係式を作る。



偏微分方程式の中の偏導関数をその差分近似で置き換えることにより、偏微分方程式が連立一次方程式になる。

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

を、点 $(x, y) = (ih, jk)$ で離散化



$$\frac{1}{h^2} (u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1}{k^2} (u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = 0$$

$$h = k \text{ とすると } u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = 0$$

$u_{i,j}$ を辞書式順序で並べたベクトルを \mathbf{u} とすると、点 (i,j) に関する離散式は

$$\mathbf{u} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ \vdots \\ u_{m,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ \vdots \\ u_{m,2} \\ \vdots \\ \vdots \\ u_{1,j} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{m,j} \\ \vdots \\ \vdots \\ u_{1,n} \\ u_{2,n} \\ u_{3,n} \\ \vdots \\ u_{m,n} \end{bmatrix}$$

$$u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = 0$$

$$\begin{bmatrix} 0 & \cdots & \underbrace{1 \quad \cdots \quad 1}_{m \text{離れている}} & -4 & \underbrace{1 \quad \cdots \quad 1}_{m \text{離れている}} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ \vdots \\ u_{m,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ \vdots \\ u_{m,2} \\ \vdots \\ \vdots \\ u_{1,j} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{m,j} \\ \vdots \\ \vdots \\ u_{1,n} \\ u_{2,n} \\ u_{3,n} \\ \vdots \\ u_{m,n} \end{bmatrix} = 0$$

$A = (a_{ij}) : n \times n$ 正方行列, 正則かつ $a_{ij} \neq 0 (1 \leq i \leq n)$

$\mathbf{x} = (x_i), \mathbf{b} = (b_i) : n$ 次元ベクトル

n 次線形方程式

$$A\mathbf{x} = \mathbf{b}, \text{ または } \sum_{j=1}^n a_{ij}x_j = b_i (1 \leq i \leq n)$$

を解く.

■ 直接法： 行列操作により厳密解を求める方法

- ガウス消去法
- LU分解法（対称行列用 コレスキー法）など

■ 反復法

- ◆ $\mathbf{x}^{(0)}$ を真の解 \mathbf{x}^* の近似値とし, 反復ベクトルの系列 $\mathbf{x}^{(0)} \rightarrow \mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)} \rightarrow \dots \rightarrow \mathbf{x}^{(m)}$ により, 真の解に近づける方法
- ◆ 反復ベクトルの系列の作り方によりいろいろな解法がある.

▶ 定常反復法

- ヤコビ法 (Jacobi法)
- ガウス・ザイデル法 (Gauss-Seidel法)
- 逐次過大緩和法 (Successive Over-Relaxation : SOR) 法
- 交互方向法 (Alternating-Direction Implicit Iterative Method: ADI法) など

▶ クリロフ (Krylov) 空間法

- 共役勾配法 (Conjugate Gradient Method, CG法)
- 双共役勾配法 (Bi-CG法)
- 一般化最小残差法 (GMRES法) など

→ 講義4

$$A = D - E - F$$

行列分離： $A = D - E - F$

$D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$ ：対角行列

E：狭義下三角行列

F：狭義上三角行列

$$Ax = b \quad \Rightarrow \quad (D - E - F)x = b$$

(1) ヤコビ法 (Jacobi Method)

$$(D - E - F)x = b \quad \Rightarrow \quad Dx = (E + F)x + b$$

$$x^{(m+1)} = D^{-1}(E + F)x^{(m)} + D^{-1}b \quad (m \geq 0)$$

ヤコビ行列

で、反復ベクトル系列を生成する。

$$x_i^{(m+1)} = \left(- \sum_{j=1, j \neq i}^n a_{ij} x_j^{(m)} + b_i \right) / a_{ii}$$

$(1 \leq i \leq n, m \geq 0)$

$$\begin{array}{ccc} x_1^{(m)} & & x_1^{(m+1)} \\ x_2^{(m)} & & x_2^{(m+1)} \\ x_3^{(m)} & & x_3^{(m+1)} \\ \vdots & \Rightarrow & \vdots \\ \vdots & & \vdots \\ x_{n-1}^{(m)} & & x_{n-1}^{(m+1)} \\ x_n^{(m)} & & x_n^{(m+1)} \end{array}$$

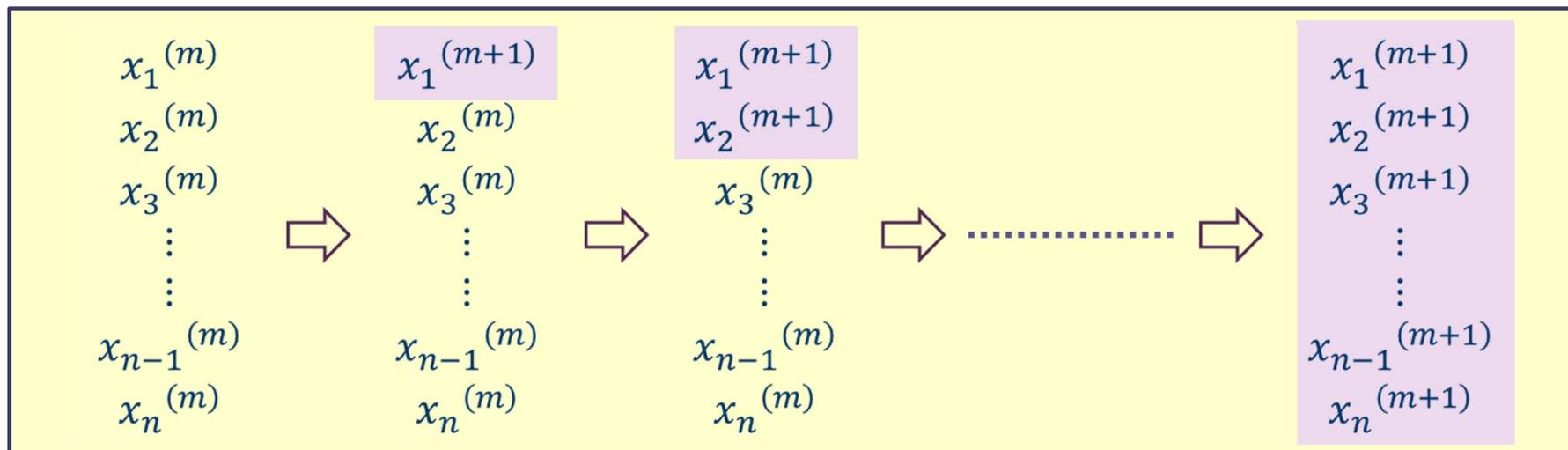
(2) ガウス・ザイデル法 (Gauss-Seidel Method)

$$(D - E - F)x = b \quad \Rightarrow \quad (D - E)x = Fx + b$$

$$x^{(m+1)} = (D - E)^{-1}F x^{(m)} + (D - E)^{-1}b \quad (m \geq 0)$$

ガウス・ザイデル行列

$$x_i^{(m+1)} = \left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(m)} + b_i \right) / a_{ii} \quad (1 \leq i \leq n, m \geq 0)$$



(3) SOR法 (Successive Over-Relaxation Method)

$$x_i^{(m+1)} = x_i^{(m)} + \omega(\tilde{x}_i - x_i^{(m)}) = (1 - \omega)x_i^{(m)} + \omega\tilde{x}_i$$

$$\tilde{x}_i = \left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(m)} + b_i \right) / a_{ii} \quad (1 \leq i \leq n, m \geq 0)$$

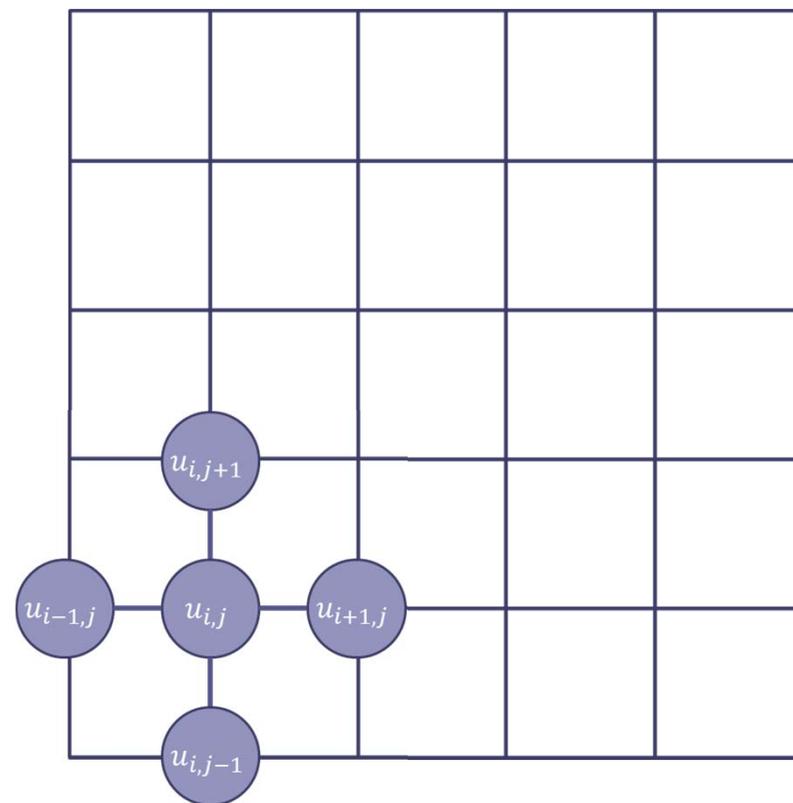
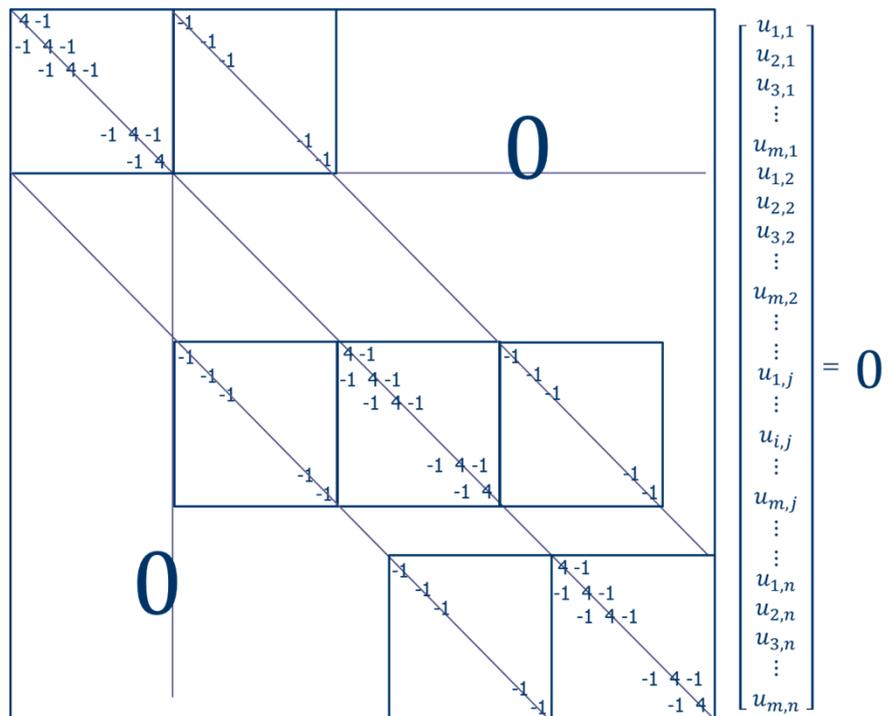
これらを整理すると,

$$a_{ii}x_i^{(m+1)} = (1 - \omega)a_{ii}x_i^{(m)} + \omega \left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(m)} + b_i \right)$$

$$\mathbf{x}^{(m+1)} = (I - \omega L)^{-1} \{ (1 - \omega)I + \omega U \} \mathbf{x}^{(m)} + \omega (I - \omega L)^{-1} D^{-1} \mathbf{b}$$

SOR行列

熱伝導問題への反復法の適用 (ヤコビ法)



■ ヤコビ法

$$u_{i,j}^{m+1} = (u_{i,j-1}^m + u_{i-1,j}^m + u_{i+1,j}^m + u_{i,j+1}^m) / 4$$

→ 演習 1

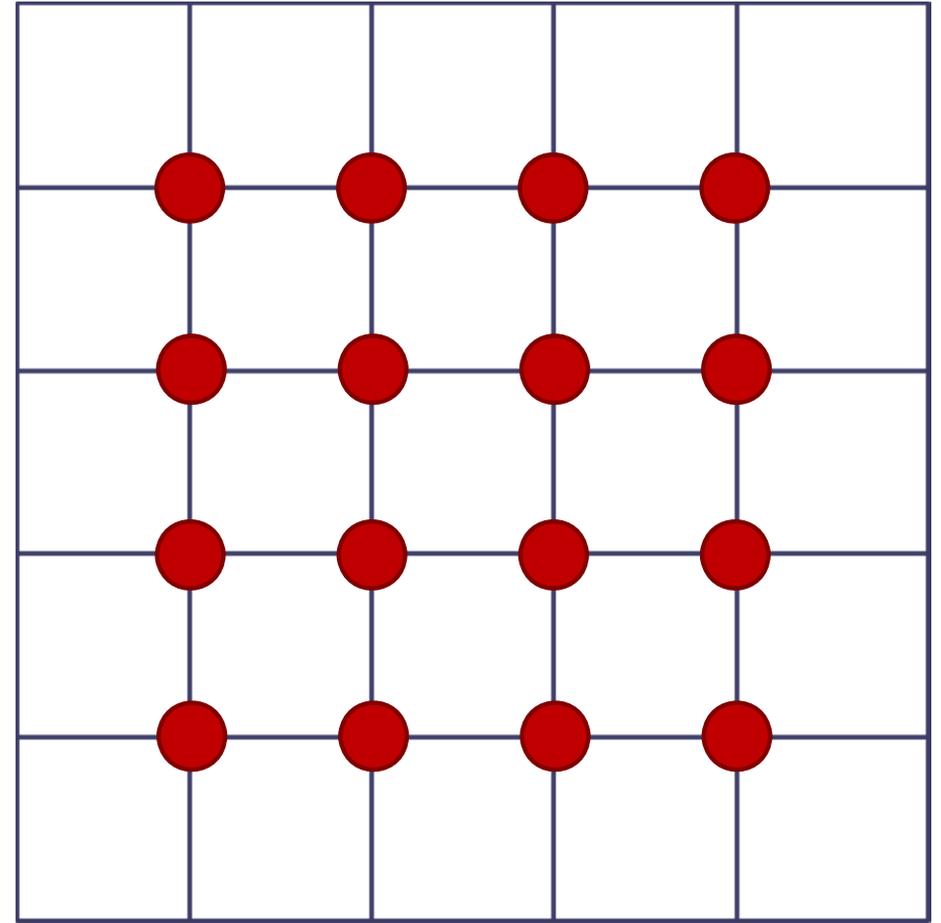
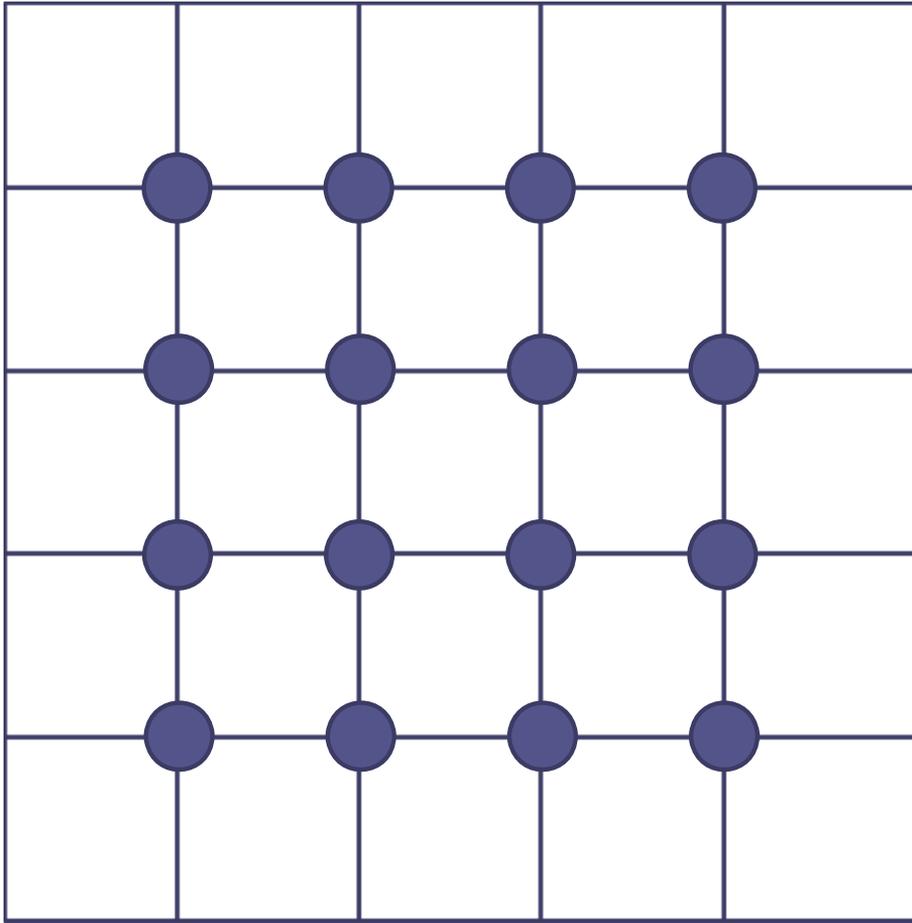
■ ガウス・ザイデル法

$$u_{i,j}^{m+1} = (u_{i,j-1}^{m+1} + u_{i-1,j}^{m+1} + u_{i+1,j}^m + u_{i,j+1}^m) / 4$$

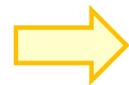
$u^{(m)}$



$u^{(m+1)}$



$u_{i,j}^m$ から $u_{i,j}^{m+1}$ を計算するので、どの点から計算しても同じ結果になる。

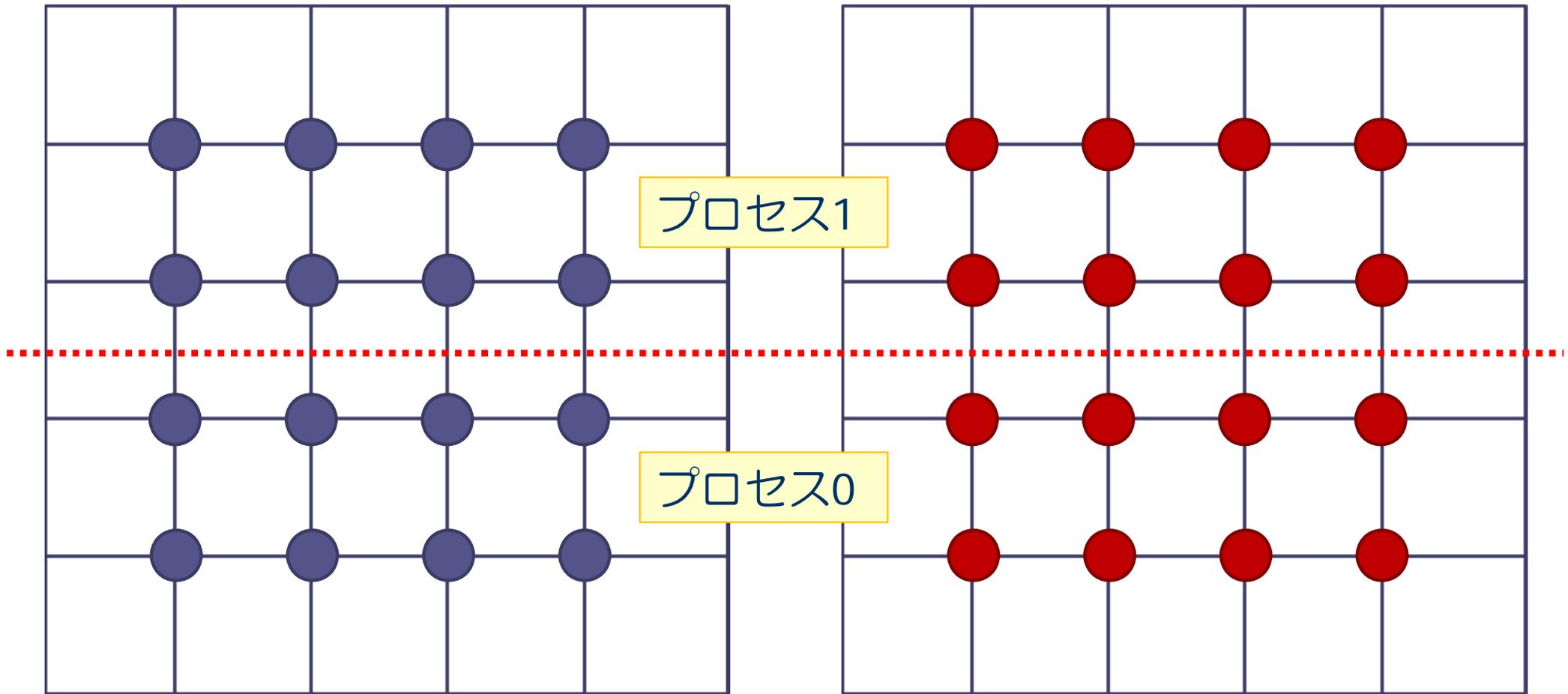


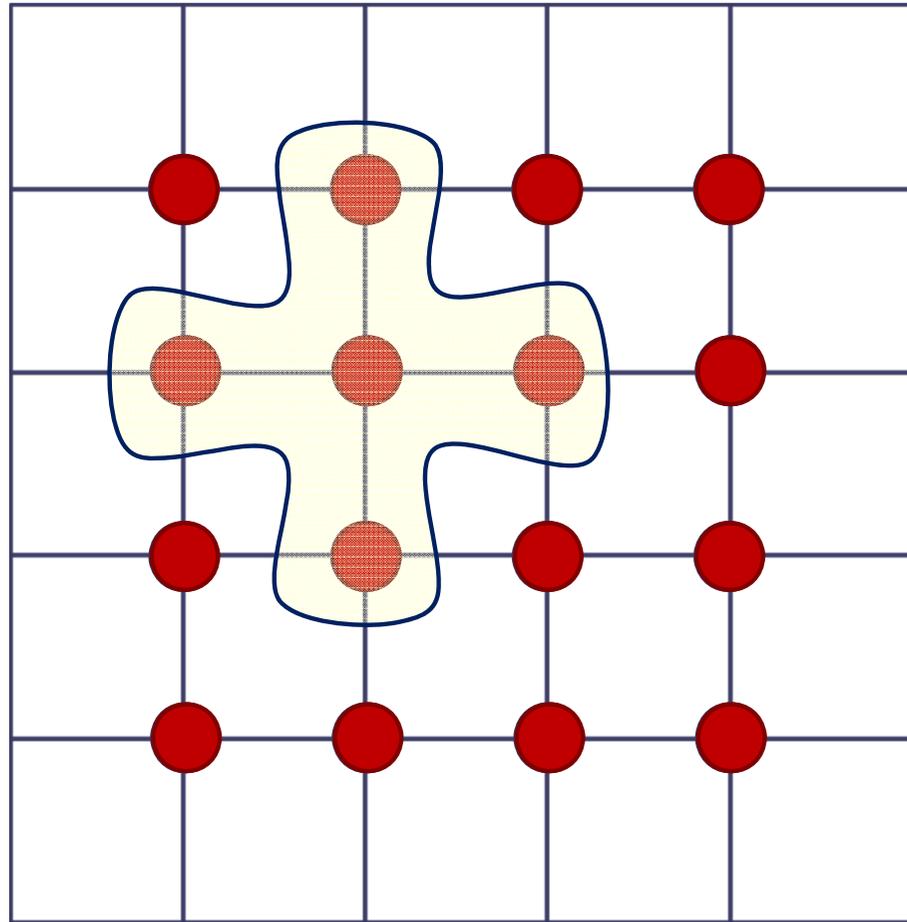
領域分割によって並列化が可能。

$x^{(m)}$

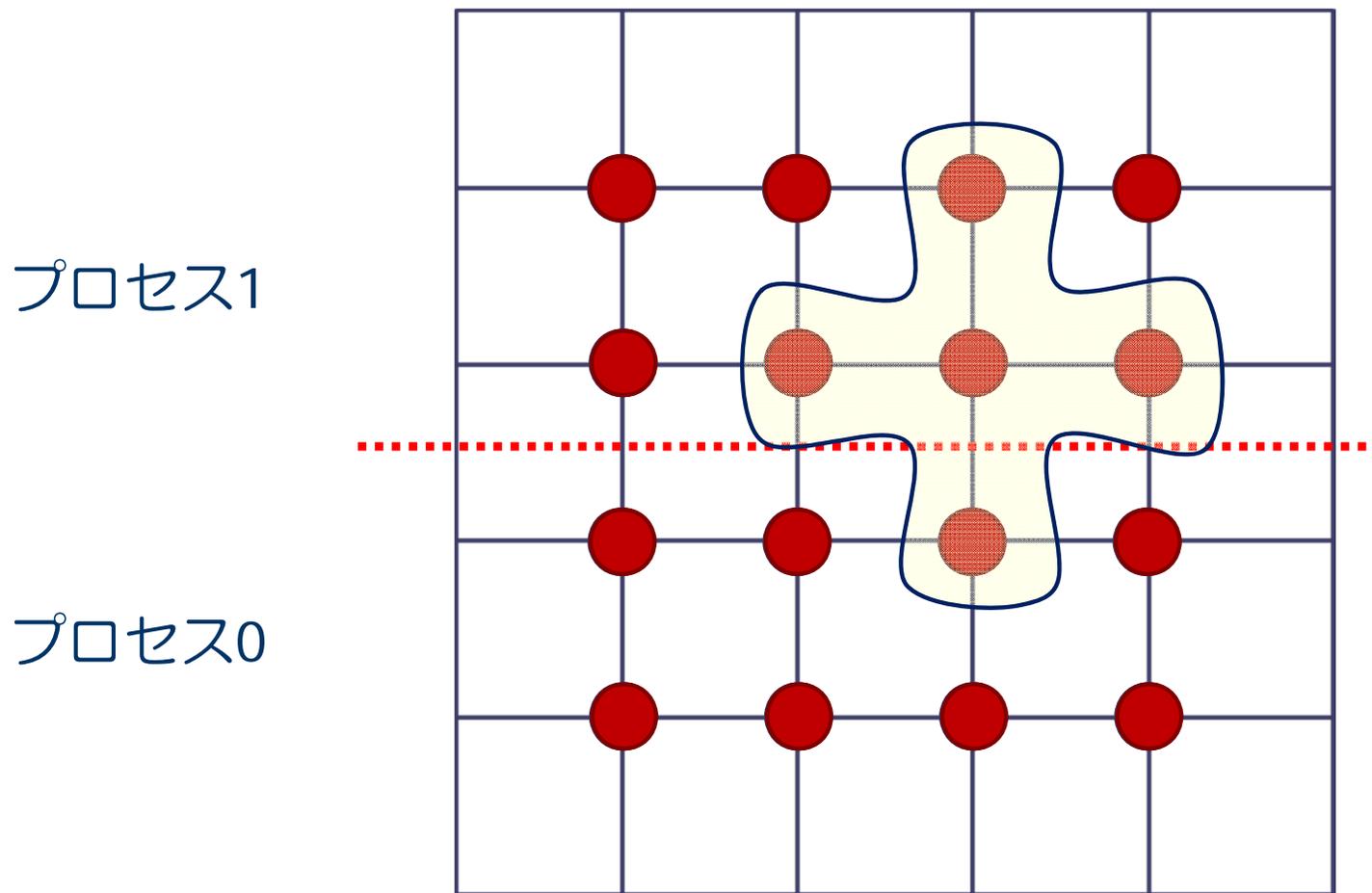


$x^{(m+1)}$





$u_{i,j}^{m+1}$ の計算には, 更新された $u_{i,j-1}^{m+1}$, $u_{i-1,j}^{m+1}$ が必要. このままでは
並列化出来ない.

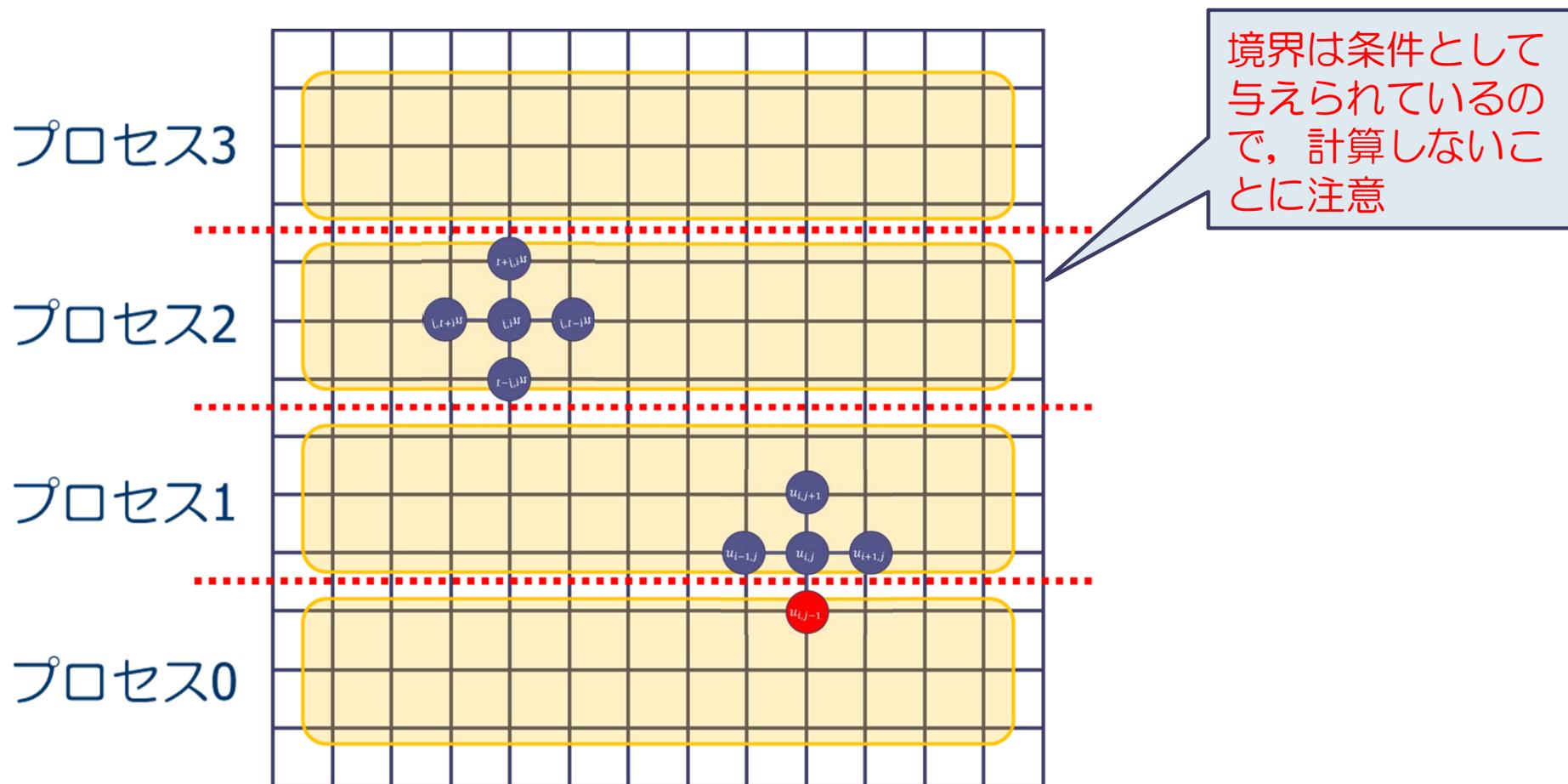


計算の順番に着目して、並列化可能なプログラムを見つける。

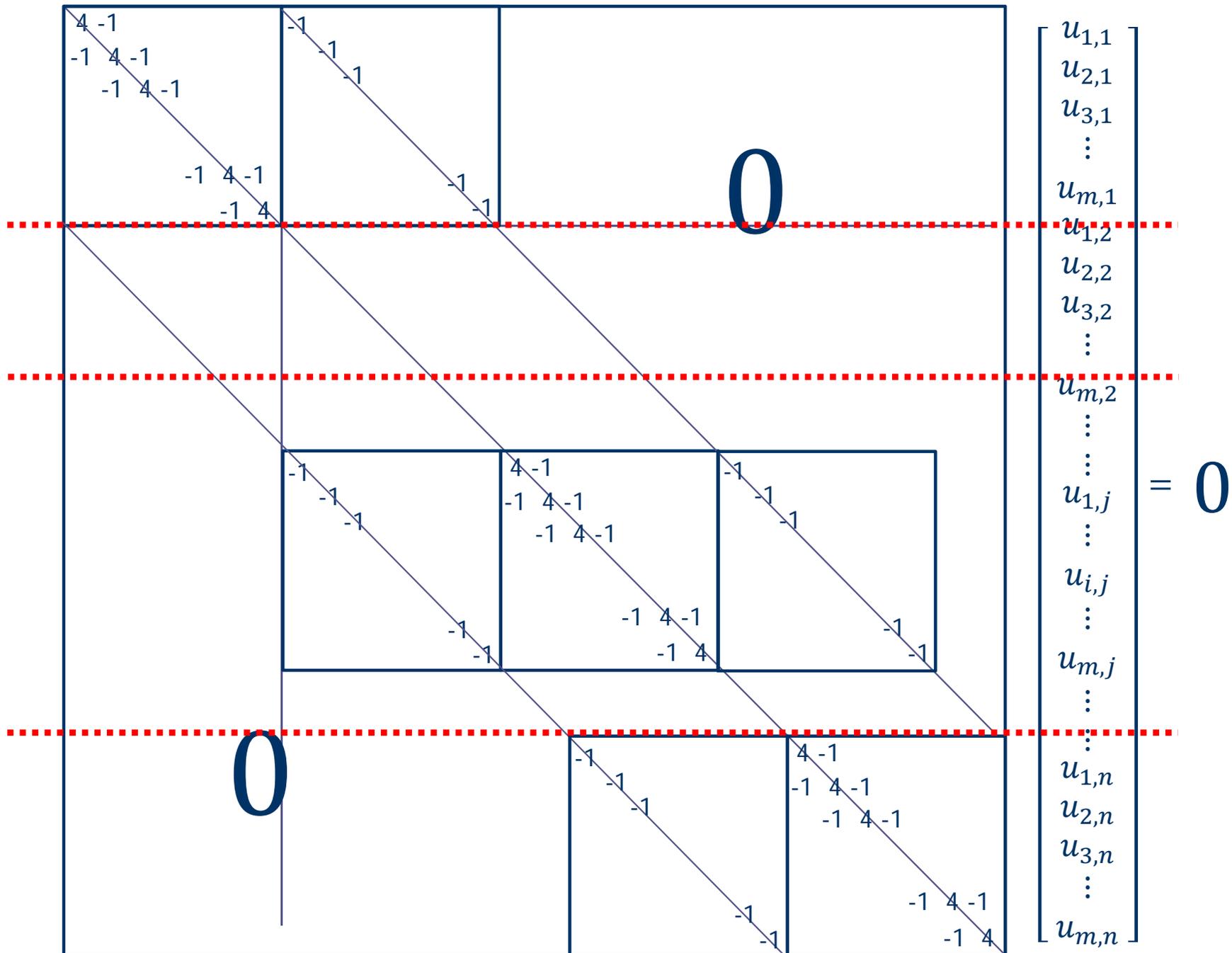
➡ マルチカラー法

ヤコビ法のMPI並列化の方針（その1）

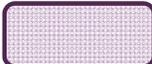
- 領域分割によりMPI並列化（プロセス並列化）を行う。
 - 各プロセスは、 部分の計算を担当する。



ヤコビ法のMPI並列化の方針 (その2)

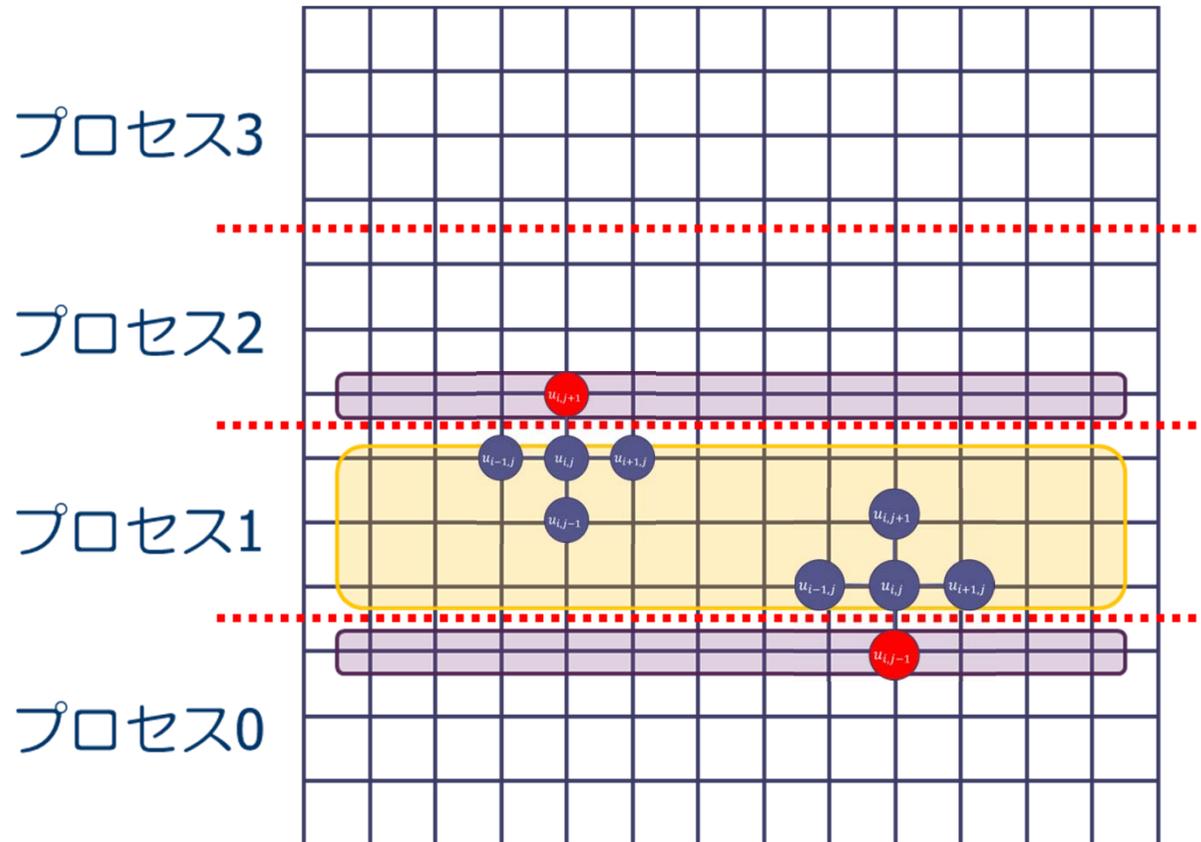


ヤコビ法のMPI並列化の方針（その3）

- 各プロセスでの計算には、上下の部分の値  が必要であることに注意。

$$u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = 0$$

- スレッド並列では、 の部分に同時にアクセスしないようにする。
- プロセス並列では、上下部分の値を転送する必要がある。



- 並列実行数とともに、実行時間が減少することが期待される。ただし、並列化により、計算結果が異ならないよう注意する。
 - ◆ 並列数 n に対し、計算時間は $1/n$ になることが期待
 - ◆ 実行時間を計測して確認。

- スケーラビリティ
 - ◆ 弱スケーリング (weak scaling)

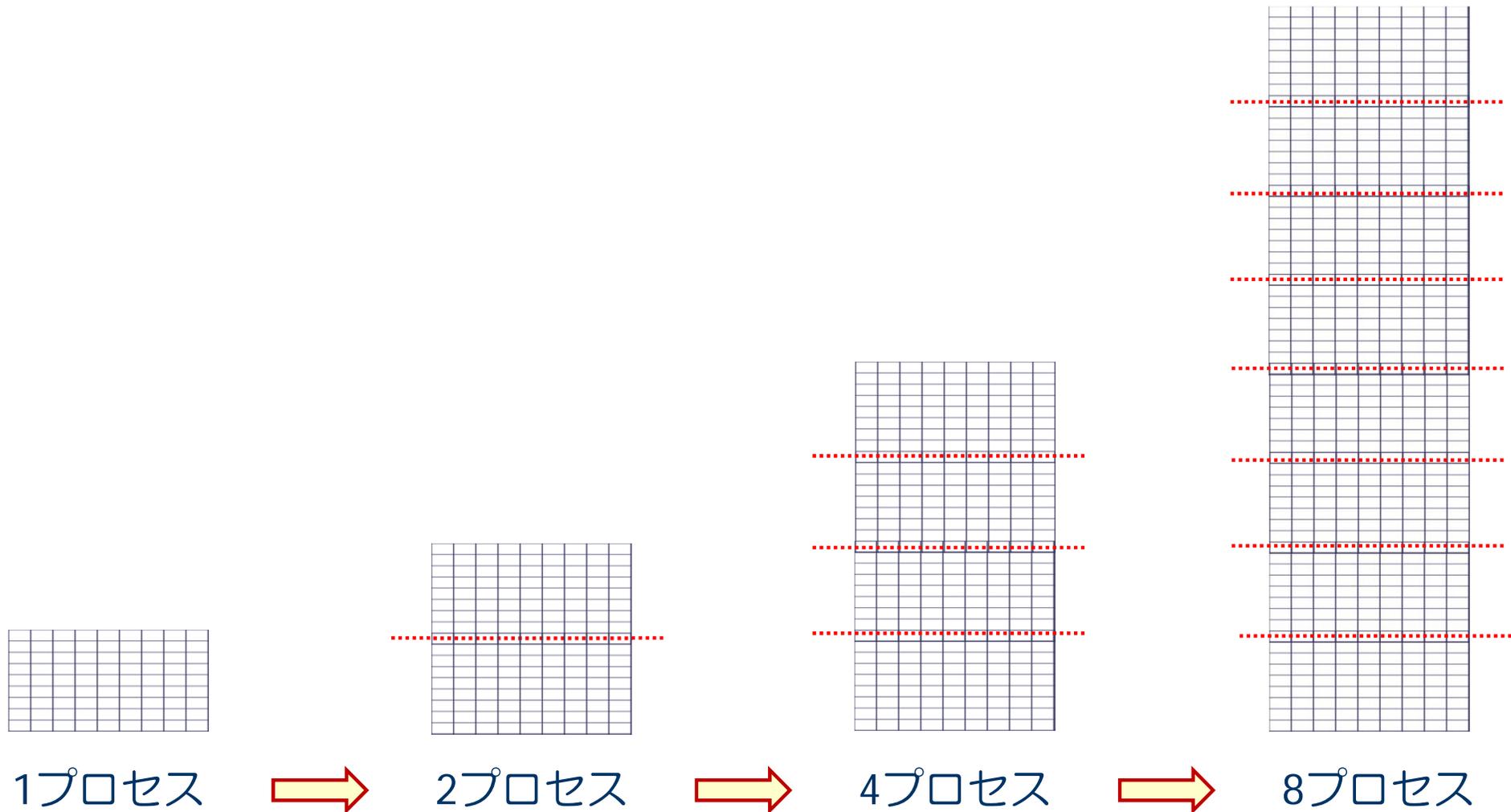
並列実行単位 (スレッド, またはプロセス) あたりの問題サイズを一定に保ったまま (したがって問題のサイズは大きくなる), 並列実行数 (スレッド数, プロセス数) を増加させた時に, 実行時間がどのように変化するかをみる性能評価指標

 - ◆ 強スケーリング (strong scaling)

全体の問題サイズを固定したまま, 並列実行数 (スレッド数, プロセス数) を増加させた時に, 実行時間がどのように変化するかをみる性能評価指標

弱スケーリング

- 並列実行単位（スレッド，またはプロセス）あたりの問題サイズを一定に保ったまま，並列実行数（スレッド数，プロセス数）を増加させた時に，実行時間がどのように変化するかをみる性能評価指標



強スケーリング

- 全体の問題サイズを固定したまま、並列実行数（スレッド数，プロセス数）を増加させた時に，実行時間がどのように変化するかをみる性能評価指標

