

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 計算科学演習 I

## 第 12 回: 実践編

陰山 聡

神戸大学大学院 システム情報学研究科 計算科学専攻

2014.07.17

# 内容

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

「床暖房問題」を例にとり、計算科学の実践的演習を行う。

- 問題の定式化
- 離散化
- コーディング
- 並列化 (MPI + OpenMP (ハイブリッド並列化))
- 大規模並列 (最大 84 ノード = 1344 コア) 計算
- 可視化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 準備

# gnuplot が使えるように

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

前回と同様に Xming と Tera term の設定をすること

# サンプルコードのコピー

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

```
cp -r /tmp/140717_practical 自分のディレクトリ
```

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 問題設定

# 問題設定： 床暖房システム

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

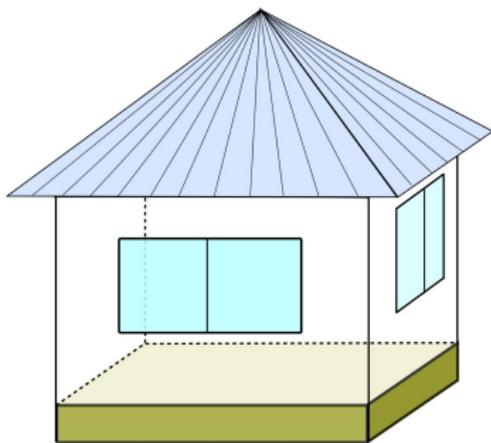
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

一間の家がある。床は長方形。外は冬。床暖房システムがあるので、家の中は暖かい。



# 問題設定：床暖房システム

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

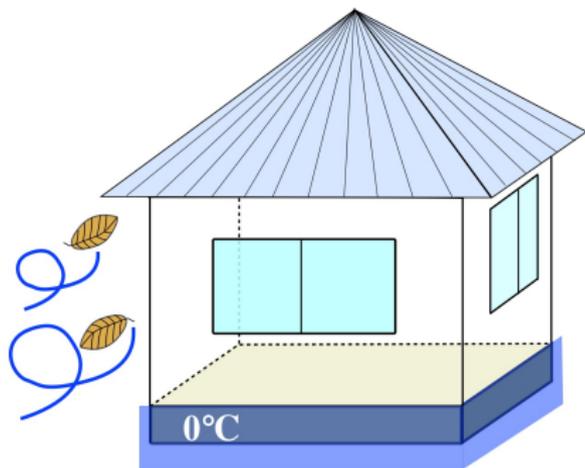
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

外気温は0度。壁も0度。



# 問題設定：床暖房システム

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

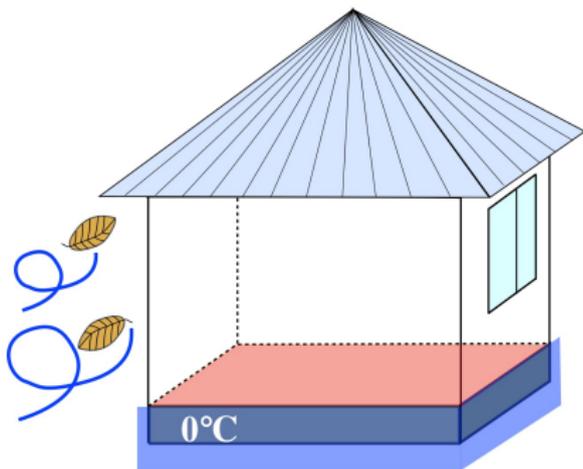
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

この床暖房システムは床全体を一様に加熱する。



# 問題設定： 暖房問題

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

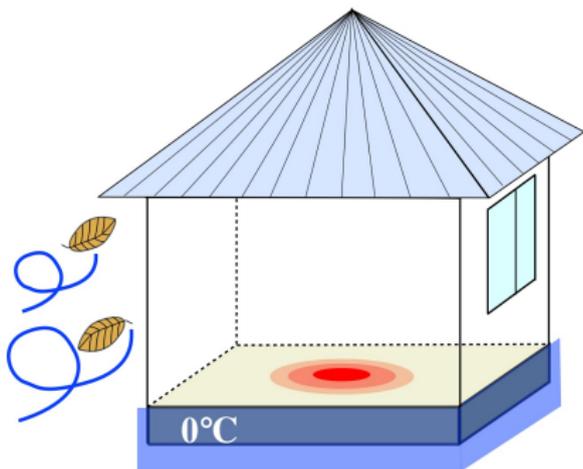
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

床暖房がオフだと、床面は全体が0度。床暖房をオンにすると床の温度は上がるが、壁に接している部分（床の周囲の長方形の辺上）は0度。問題： 最終的な温度分布は？



# 座標系

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

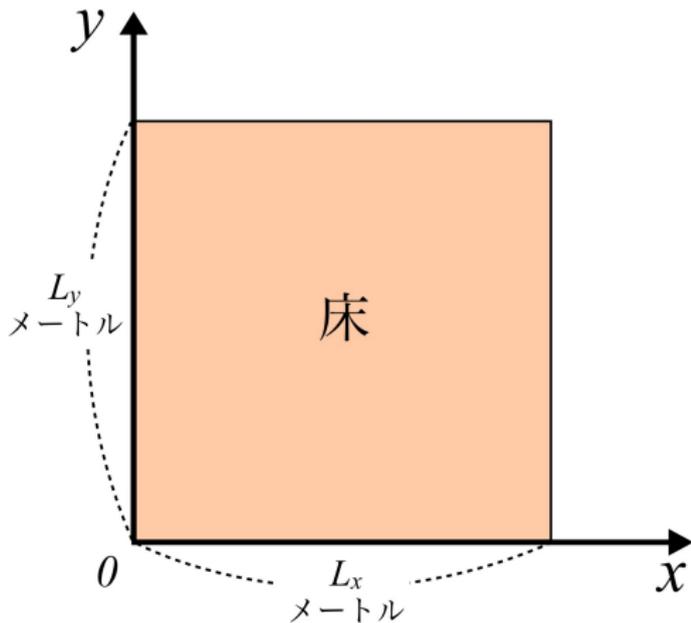
並列化

時間計測

課題

来週の演習と

課題



# 問題設定

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- $L_x \times L_y$  平方メートルの長方形領域
- 辺上の温度は常に  $0^\circ$  (固定)
- 面内に熱源が分布
- 面の熱拡散率  $k$
- 面内の温度分布は？

実践編

陰山

準備

問題設定

**熱伝導**

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 熱伝導

# 熱伝導問題

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

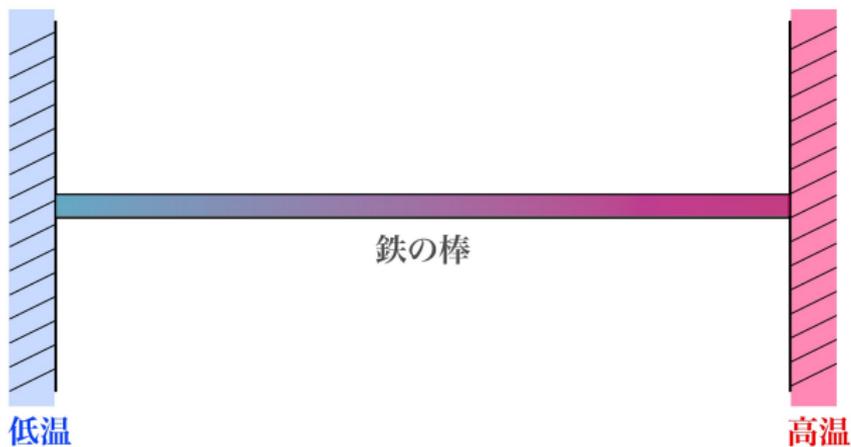
並列化

時間計測

課題

来週の演習と

課題



# 温度分布

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

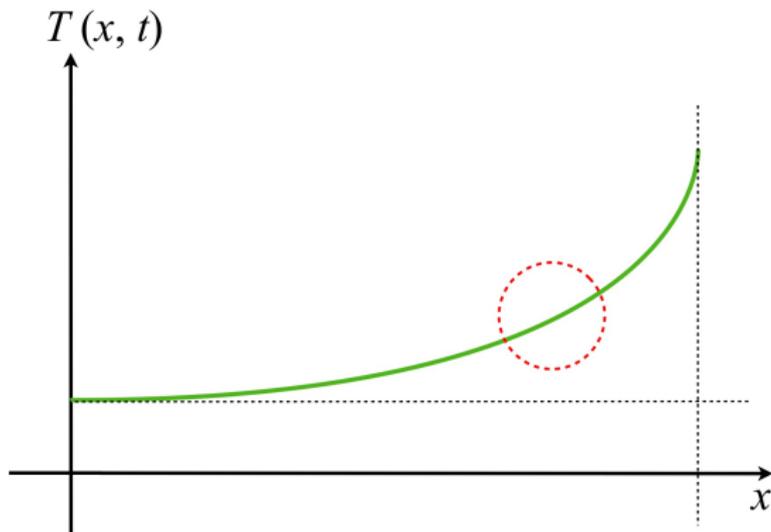
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



# 温度分布が線形（まっすぐ、傾き一定）の時

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

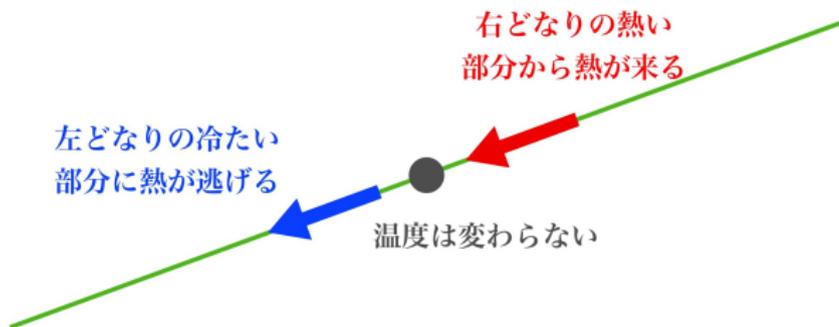
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



# 温度分布が下に凸の時

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

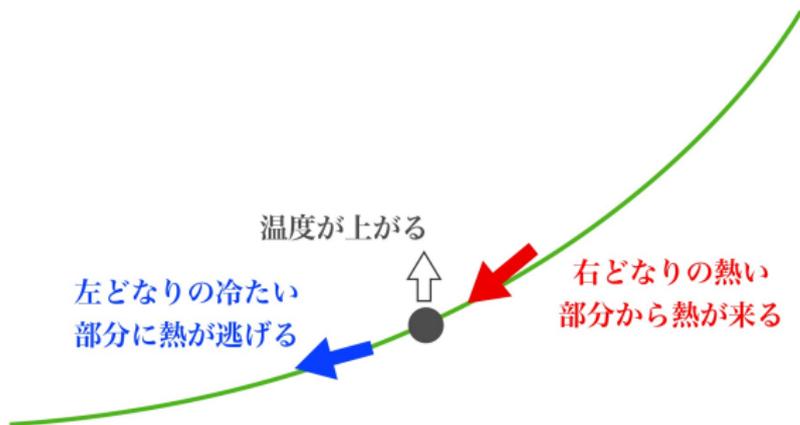
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



# 温度分布が上に凸の時

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

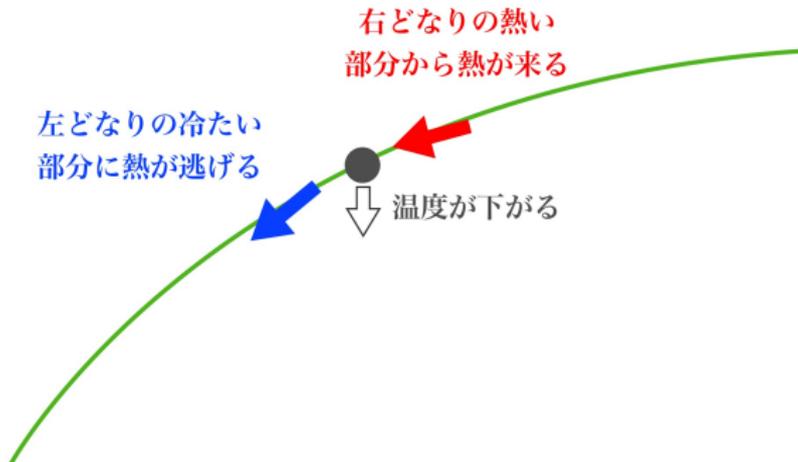
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



# 温度分布の空間微分と温度の時間変化の関係

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

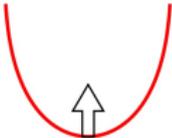
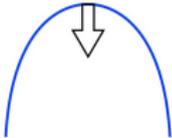
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

	$\frac{\partial T}{\partial x} = \text{一定}$ $\left( \frac{\partial^2 T}{\partial x^2} = 0 \right)$	$\frac{\partial T}{\partial t} = 0$
	$\frac{\partial^2 T}{\partial x^2} > 0$	$\frac{\partial T}{\partial t} > 0$
	$\frac{\partial^2 T}{\partial x^2} < 0$	$\frac{\partial T}{\partial t} < 0$

# 熱伝導（熱拡散）方程式

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

温度  $T(x, t)$  に対する基本方程式

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

$k$ : 熱拡散係数

# 熱源があるとき

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

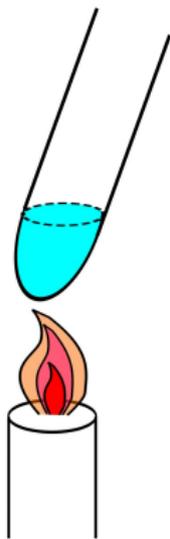
課題

来週の演習と  
課題

$$\frac{\partial T}{\partial t} = s$$

$s$  : 熱源 (heat source)

ろうそくで温度計を熱している図



# 熱拡散方程式

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

1D

$$\frac{\partial T(x, t)}{\partial t} = k \frac{\partial^2 T(x, t)}{\partial x^2} + s(x)$$

2D

$$\frac{\partial T(x, y, t)}{\partial t} = k \left\{ \frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\partial^2 T(x, y, t)}{\partial y^2} \right\} + s(x, y)$$

# 熱拡散方程式の表現

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\frac{\partial T(x, y, t)}{\partial t} = k \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) T(x, y, t) + s(x, y)$$

あるいは

$$\frac{\partial T(x, y, t)}{\partial t} = k \nabla^2 T(x, y, t) + s(x, y)$$

$\nabla^2$ : ラプラシアン

# 問題の数学的定式化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$(0, 0) \leq (x, y) \leq (L_x, L_y)$  の長方形領域で

$T(0, y) = T(L_x, y) = T(x, 0) = T(x, L_y) = 0$  という境界条件のもと

$$\frac{\partial T(x, y, t)}{\partial t} = k \nabla^2 T(x, y, t) + s(x, y)$$

という熱拡散方程式を解き、

定常状態 ( $\frac{\partial T}{\partial t} = 0$ ) の温度分布

$T(x, y)$  を求めよ。

実践編

陰山

準備

問題設定

熱伝導

**離散化**

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 離散化

# 時間・空間の離散化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

スーパーコンピュータが「スーパー」に速いのは四則演算である。

⇒ 全てを四則演算で表す。

連続点 

離散点 

# 熱拡散方程式の離散化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\frac{\partial T(x, y, t)}{\partial t} = k \nabla^2 T(x, y, t) + s(x, y)$$

をどう四則演算の問題として表現するか？

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 差分法

# 微分の離散化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\frac{\partial T}{\partial t} \quad (\text{一階の微分})$$

$$\frac{\partial^2 T}{\partial x^2} \quad (\text{二階の微分})$$

等を四則演算で表現する。

# 差分

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

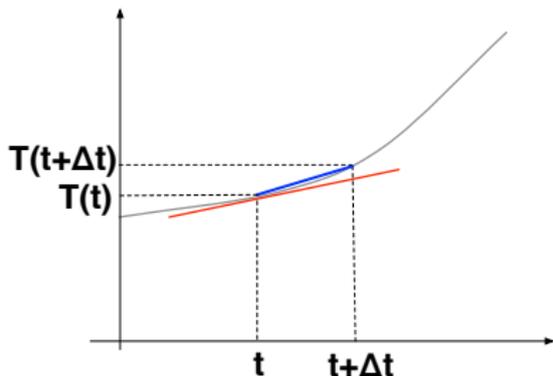
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



$$\frac{\partial T}{\partial t} \sim \frac{T(t + \Delta t) - T(t)}{\Delta t}$$

引き算 1 回 + 割り算 1 回で微分を近似 (  $1/\Delta t$  をあらかじめ計算しておけば、割り算の代わりに掛け算 )

# 2階差分

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\frac{d^2T(x)}{dx^2} \sim \frac{T(x + \Delta x) - 2T(x) + T(x - \Delta x)}{\Delta x^2}$$

# 偏微分の差分表現

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\frac{d^2T(x)}{dx^2} \sim \frac{T(x + \Delta x) - 2T(x) + T(x - \Delta x)}{\Delta x^2}$$

$$\frac{d^2T(x_i)}{dx^2} \sim \frac{T(x_{i+1}) - 2T(x_i) + T(x_{i-1}))}{\Delta x^2}$$

$$\frac{\partial^2T(x_i, y_j)}{\partial x^2} \sim \frac{T(x_{i+1}, y_j) - 2T(x_i, y_j) + T(x_{i-1}, y_j)}{\Delta x^2}$$

$$\frac{\partial^2T(x_i, y_j)}{\partial y^2} \sim \frac{T(x_i, y_{j+1}) - 2T(x_i, y_j) + T(x_i, y_{j-1}))}{\Delta y^2}$$

# 空間の離散化と格子点の番号付け

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

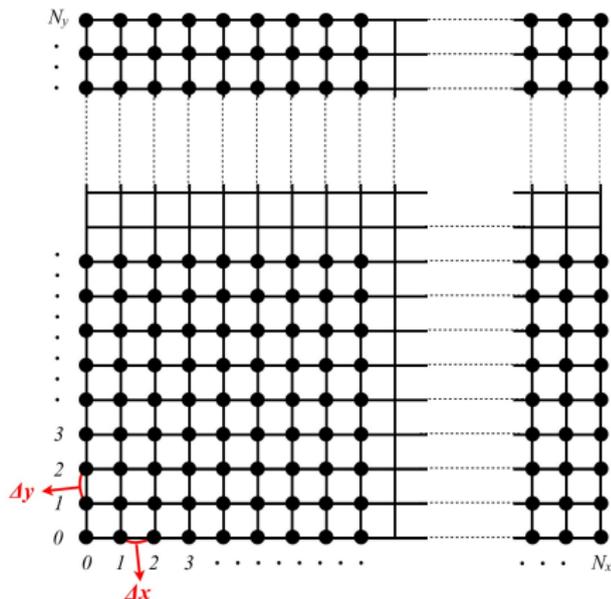
並列化

時間計測

課題

来週の演習と

課題



# 熱拡散方程式の差分表現

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

$$\frac{\partial T(x, y, t)}{\partial t} = k \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) T(x, y, t) + s(x, y)$$

$T(x_i, y_j, t)$   $T_{i,j}(t)$  と略記。

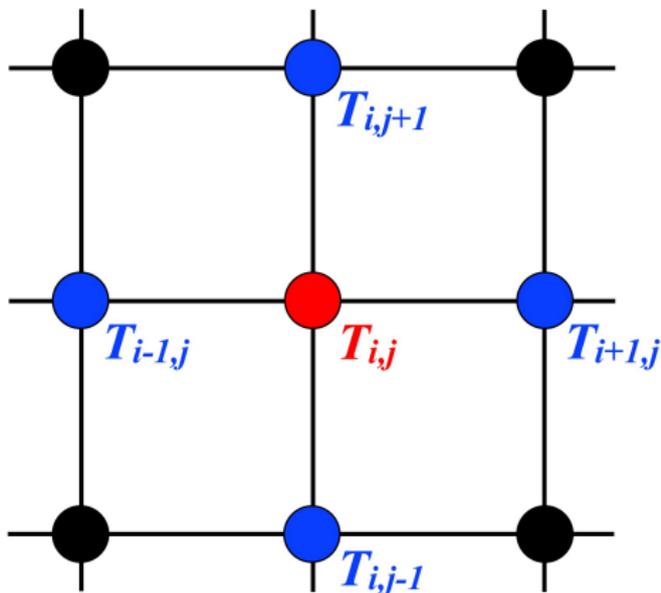
$$\begin{aligned} \frac{T_{i,j}(t + \Delta t) - T_{i,j}(t)}{\Delta t} &= k \left( \frac{T_{i+1,j}(t) - 2T_{i,j}(t) + T_{i-1,j}(t)}{\Delta x^2} \right. \\ &\quad \left. + \frac{T_{i,j+1}(t) - 2T_{i,j}(t) + T_{i,j-1}(t)}{\Delta y^2} \right) \\ &\quad + s_{i,j} \end{aligned}$$

# 差分テンシル

実践編

陰山

$$\frac{T(x_{i+1}, y_j) - 2T(x_i, y_j) + T(x_{i-1}, y_j))}{\Delta x^2}, \quad \frac{T(x_i, y_{j+1}) - 2T(x_i, y_j) + T(x_i, y_{j-1}))}{\Delta y^2}$$



準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

# 時間発展の式

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$\begin{aligned} T_{i,j}(t + \Delta t) = & T_{i,j}(t) \\ & + \frac{k \Delta t}{\Delta x^2} \{T_{i+1,j}(t) - 2T_{i,j}(t) + T_{i-1,j}(t)\} \\ & + \frac{k \Delta t}{\Delta y^2} \{T_{i,j+1}(t) - 2T_{i,j}(t) + T_{i,j-1}(t)\} \\ & + \Delta t s_{i,j} \end{aligned}$$

この式は

$\Delta t$  だけ未来の温度 = 現在の温度分布の四則演算

という形をしている。

# 式変形

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$T_{i,j}(t + \Delta t) = \Delta t \left\{ \frac{k}{\Delta x^2} (T_{i+1,j} + T_{i-1,j}) + \frac{k}{\Delta y^2} (T_{i,j+1} + T_{i,j-1}) + s_{i,j} \right\} + \left\{ 1 - \Delta t \cdot 2 \left( \frac{k}{\Delta x^2} + \frac{k}{\Delta y^2} \right) \right\} T_{i,j}$$

最後の項 ( $T_{i,j}$  に比例する項) の係数がゼロになるように  $\Delta t$  を決める。

つまり

$$\alpha_x = \frac{k}{\Delta x^2}$$

$$\alpha_y = \frac{k}{\Delta y^2}$$

を定義した上で、

$$\Delta t = \frac{1}{2(\alpha_x + \alpha_y)}$$

とする。

# 熱拡散方程式の離散化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$T_{i,j}(t + \Delta t) = \Delta t \{ \alpha_x (T_{i+1,j}(t) + T_{i-1,j}(t)) \\ + \alpha_y (T_{i,j+1}(t) + T_{i,j-1}(t)) \\ + s_{i,j} \}$$

現在の時刻の温度分布から、 $\Delta t$  だけ未来の温度分布を計算。  
これを繰り返して定常状態になるまで計算すればよい。

⇒ ヤコビ (Jacobi) 法

# 最も簡単な場合

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

熱拡散係数  $k = 1$  で、熱源  $s(x, y) = 4$ 、そして  $\Delta x = \Delta y \equiv h$  の時

$$\alpha_x = \alpha_y = 1/h^2, \quad \Delta t = h^2/4$$

$$T_{i,j}(t + \Delta t) = \frac{T_{i+1,j}(t) + T_{i-1,j}(t) + T_{i,j+1}(t) + T_{i,j-1}(t)}{4} + h^2$$

と簡単な式になる。

# ヤコビ法のイメージ

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

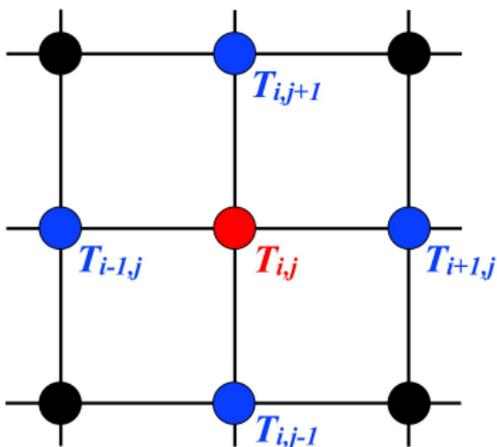
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$T_{i,j}(t + \Delta t) = \frac{T_{i+1,j}(t) + T_{i-1,j}(t) + T_{i,j+1}(t) + T_{i,j-1}(t)}{4} + h^2$$



# ヤコビ法のイメージ

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

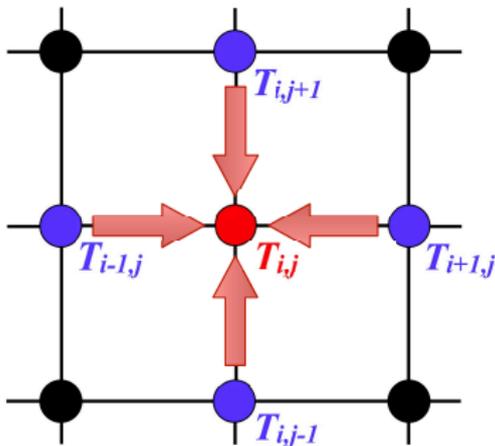
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$$T_{i,j}(t + \Delta t) = \frac{T_{i+1,j}(t) + T_{i-1,j}(t) + T_{i,j+1}(t) + T_{i,j-1}(t)}{4} + h^2$$



実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

**実装**

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 実装

# サンプルコード: heat2.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- ヤコビ法で正方形領域の定常状態の温度分布を求める
- 100 ステップに一度、中心の温度を書き出す

# コード解説: heat2.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
do n=1, LOOP_MAX ! 時間更新
  do j=1, NGRID
    do i=1, NGRID
      un(i,j)=(u(i-1,j)+u(i+1,j)+... ! 作業配列
    end do
  end do
  u ... = un ... ! un を u にコピー
end do
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- heat2.f90 を理解しよう
- pi-computer 上でシリアルで実行してみよう: `gfortran heat2.f90 && ./a.out`

ちなみに中心点の温度の収束値の解析解は 0.294...

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

**並列化**

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 並列化

# 1次元並列化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

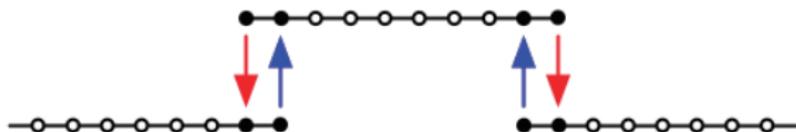
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

$x$  軸方向または  $y$  軸方向の 1 次元空間を複数の領域に分割する  
(領域分割)。



MPI の sendrecv で通信。  
以下では  $y$  軸方向に分割する。

# 1次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

**並列化**

可視化

2次元可視化

ジョブキュー

ハイブリッド

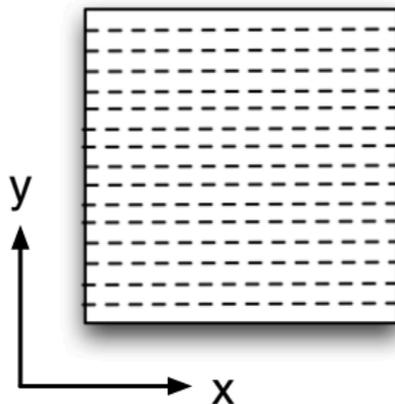
並列化

時間計測

課題

来週の演習と

課題



# 1次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

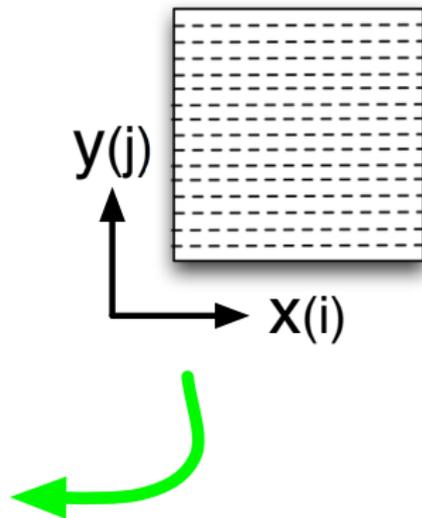
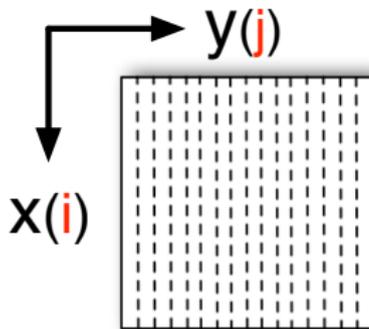
時間計測

課題

来週の演習と

課題

コードのコメント



# サンプルコード: heat3.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
!  
!       When NGRID=11, nprocs=3  
!  
!       j=0  1  2  3  4  5  6  7  8  9 10 11 12  
! i=0 +---+---+---+---+---+---+---+---+---+---+---+---+  
!   1 |   |   |   |   |   |   |   |   |   |   |   |   |  
!   2 |   |   |   |   |   |   |   |   |   |   |   |   |  
!   3 |   |   |   |   |   |   |   |   |   |   |   |   |  
!     .  
!     .  
!     .  
!     .  
! 11 |   |   |   |   |   |   |   |   |   |   |   |   |  
! 12 +---+---+---+---+---+---+---+---+---+---+---+---+  
!           | rank0 |   |   |   |   |   |   |   |   |   |  
!           jstt-----jend |   |   |   |   |   |   |   |   |  
!                           | rank1 |   |   |   |   |   |   |  
!                           jstt-----jend
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- heat3.f90 にはバグ（足りない部分）がある。それを修正しよう。
- デバッグしたら pi-computer 上で並列計算してみよう。実行方法は冒頭のコメント行。

```
!  
! heat3.f90  
!   + MPI, 1-D decomposition  
!   usage (on pi-computer)  
!     1) mpifirtpx heat3.f90  
!     2) pjsub heat3.sh
```

# ジョブスクリプト heat3.sh

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

```
#!/bin/bash
#PJM -N "heat3"
#PJM -L "rscgrp=small"
#PJM -L "node=4"
#PJM -L "elapse=02:00"
#PJM -j

drawLine()
{
    echo "#"{1..50} | sed 's/[ 0-9]//g'
}

drawLine
mpiexec ./a.out
drawLine
```

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 可視化

# 時間発展の可視化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

(標準) 出力ファイル `heat3.o....` の中身を確認せよ  
エディタで開くよりも `more / less / head / tail` コマンドで見  
る方が早い。

`gnuplot` を立ち上げ、コマンドプロンプトに  
`plot 'heat3.o....' w lp`  
と入れよ。

# 出力例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

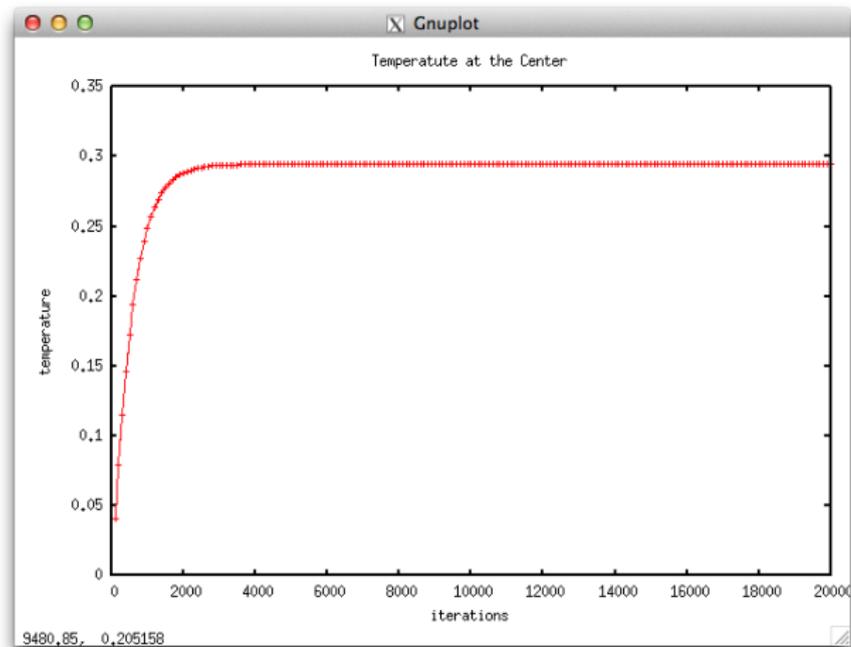
並列化

時間計測

課題

来週の演習と

課題



# 最終温度分布の可視化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

heat3.f90 で計算された最終的な平衡温度分布を gnuplot で見てみよう。

正方形を真ん中で横に切る  $y=0.5$  の線上での温度の  $x$  分布をグラフにする。

グリッド番号  $i$  ではなく、 $x$  座標の値を書き出す。 $i$  番目の格子点の  $x$  座標は

$$x_i = h \times (i - n_{\text{mid}})$$

という関係にある。

# data ディレクトリ

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2 次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

これからの演習でデータファイルを多数生成するので、データファイル出力専用のディレクトリを用意しよう。

- 例えばいまソースコードの置いていあるディレクトリと同じ高さに

```
mkdir ../data
```

などとして data ディレクトリを作成すること。別の場所、名前でも構わないが、以後のサンプルコードではそれに応じて適宜 data ディレクトリの名前を変更すること。

# heat3.f90 にデータ書き出し機能をつける

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
heat3_print_final_x_prof.f90
```

```
diff heat3.f90 heat3_print_final_x_prof.f90
```

```
! heat3_print_final_x_prof.f90
```

```
!   + open/close file 10
```

```
!   + print out cross section 1D data.
```

```
!   + integer jcut (cross section for output)
```

```
!   + function this_process_has()
```

```
!   usage (on pi-computer)
```

```
!     1) mkdir ../data (unless there is already.)
```

```
!     2) mpifrtpx heat3_print_final_x_prof.f90
```

```
!     3) pjsub heat3.sh (share the jobscript with heat
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

- (1) `heat3_print_final_x_prof.f90` をコンパイルし、実行せよ。
  - コンパイル `mpifort heat3_print_final_x_prof.f90`
  - ジョブ投入 `pjsub heat3.sh` (同じジョブスクリプトを使う)
- (2) うまくいけばディレクトリ `../data/` に `temp.final_profile_x` というファイルができています。
- (3) `more / less / head / tail` コマンドで確認。
- (4) `../data` ディレクトリに移動し、`gnuplot` を立ち上げる
- (5) `plot "temp.final_profile_x" w lp`

# gnuplot スクリプト src/heat3\_print\_final\_x\_prof.gp

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

```
# heat3_print_final_x_prof.gp
#
# final temperature profile at y=0.5 as a function of
#
set xrange [-0.5:0.5]
set yrange [0:0.5]
set xlabel "x"
set ylabel "temp at y=0.5"
plot "../data/temp.final_profile_x" w lp
pause -1
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

gnuplot がまだ立ち上がっていたら quit コマンドで終了し、改めて shell で (今回は src ディレクトリで)  
gnuplot heat3\_print\_final\_x\_prof.gp  
と打て。

# 結果例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

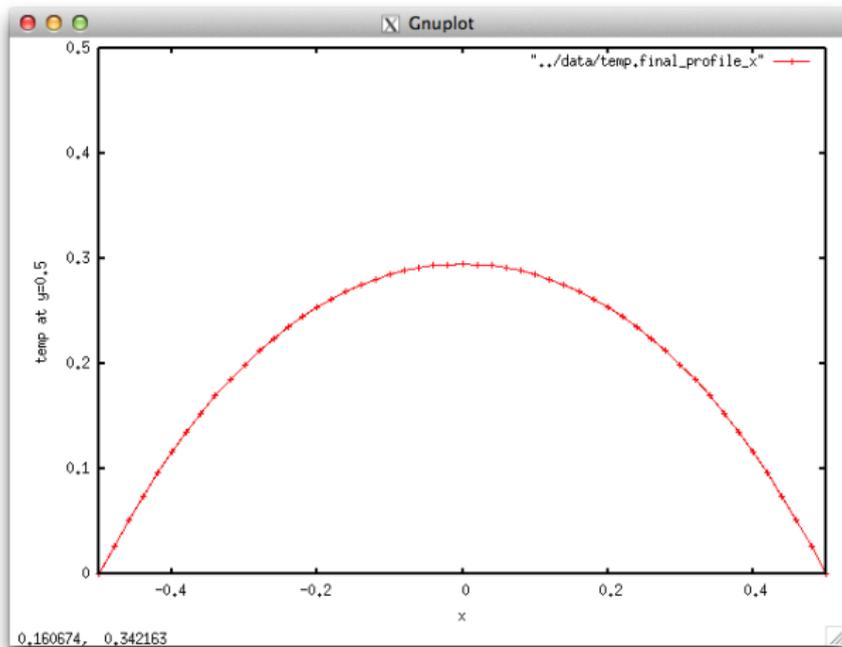
並列化

時間計測

課題

来週の演習と

課題



# アニメーション

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

アニメーションによって収束の様子を確認しよう。そのためのデータ（連番つきファイル群）を書き出すためのプログラム `heat3_print_x_prof_for_animation.f90`  
このプログラムをコンパイル+実行せよ。  
ジョブスクリプト（これまで同様）`heat3.sh`  
うまくいけば `data` ディレクトリに連番ファイルが出力されるはず。`ls -l` 等で確認せよ。

# アニメーション用スクリプトサンプル

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
#
# gnuplot script generated by heat3_animation_x_prof
#
set xlabel "x"           # x-axis
set ylabel "temperature" # y-axis
set xrange [-0.5:0.5]   # x-coordinate
set yrange [0.0:0.5]    # temperature min & max
plot "../data/temp.j=middle.0000" w lp
pause 5
plot "../data/temp.j=middle.0001" w lp
pause 1
plot "../data/temp.j=middle.0002" w lp
pause 1
.
.
```

# 【演習】1次元グラフ アニメーション

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- `heat3_print_x_prof_for_animation_plotscript_generator.f90`を確認せよ。
- 変数 `NGRID` と `counter_end` をチェックせよ。
- `gfortran`  
`heat3_print_x_prof_for_animation_plotscript_generator.f90`
- `./a.out > anyname.gp`
- ファイル `anyname.gp` の中身を確認する
- `gnuplot anyname.gp` で実行

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 2次元可視化

# diff heat3\_print\_x\_prof\_for\_animation.f90 heat4\_print\_final\_2d\_prof.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
! heat4_print_final_2d_prof.f90
!   + subroutine print__profile_2d
!   c module constants --> module common
!   + type ranks_t :: p
!   + type span_t :: jj
!   - myrank, nprocs, left, right (combined into "p")
!   - jstart, jend (combined into "jj")
!   + function adjust_jstart_and_jend
!   + function set_prof_2d
```

## 2D データ出力ルーチン (前半)

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

正方形上 ( $x,y$  平面上) に分布する温度をすべて書き出す。

```
subroutine print__profile_2d(p,jj,f)
  type(ranks_t), intent(in) :: p
  type(span_t),  intent(in) :: jj
  real(DP), dimension(0:NGRID+1, &
    jj%stt-1:jj%end+1), intent(in) :: f
  real(DP), dimension(0:NGRID+1,0:NGRID+1) &
    :: f_global ! 2d prof to be saved
  integer :: counter = 0 ! h
  type(span_t) :: jj2 ! u
  character(len=4) :: serial_num ! p
  character(len=*) , parameter :: base = "../data/temp.2d."
  integer :: i, j
```

## 2D データ出力ルーチン (後半)

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
    jj2 = adjust_jstart_and_jend(p,jj)
    write(serial_num,'(i4.4)') counter
    f_global(:, :) = set_prof_2d(jj,jj2,f)
    if ( p%myrank==0 ) then
        open(10,file=base//serial_num)
        do j = 0 , NGRID+1
            do i = 0 , NGRID+1
                write(10,*) i, j, f_global(i,j)
            end do
            write(10,*) ' ' ! gnuplot requires a blank line here
        end do
        close(10)
    end if
    counter = counter + 1
end subroutine print__profile_2d
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

heat4\_print\_final\_2d\_prof.f90 をコンパイル&実行してみよう。  
うまくいけば ../data/temp.2d.0000 ができているはず。

# 出力データの確認

実践編  
陰山

../data ディレクトリ中の連番つきファイル temp.2d.0000 の中身は以下のようにになっているはず。確認せよ。

準備	55	35	0.1165598588705999
問題設定	56	35	9.9624877672293416E-002
熱伝導	57	35	8.1734108631726782E-002
離散化	58	35	6.2857224006520482E-002
差分法	59	35	4.2963409431420671E-002
実装	60	35	2.2021479795155254E-002
並列化	61	35	0.0000000000000000
可視化			
2次元可視化	0	36	0.0000000000000000
ジョブキュー	1	36	2.1867122785152873E-002
ハイブリッド 並列化	2	36	4.2655284590767971E-002
時間計測	3	36	6.2396502500601705E-002
課題	4	36	8.1122529495226178E-002
来週の演習と 課題	.		

# gnuplot スクリプト生成

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
heat4_plot_contour_lines
```

```
#  
# A sample gnuplot script: heat4_plot_contour_lines.gp  
#  
# [ line contours ]  
#  
# set size square                # same side lengths for x and  
set size 0.65, 1                 # same side lengths for x and  
set xlabel "i"                   # x-axis  
set ylabel "j"                   # y-axis  
set xrange[0:50]                 # i-grid min & max  
set yrange[0:50]                 # j-grid min & max  
set nosurface                    # do not show surface plot  
unset ztics                      # do not show z-tics  
set contour base                 # enables contour lines  
set cntrparam levels 10         # draw 10 contours  
set view 0,0                     # view from the due north  
set title "Temperature"
```

# 【演習】2次元等高線の表示

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- data/temp.2d.0000 のファイルに記された温度の分布を gnuplot の等高線で可視化してみよう。
- ファイル名：heat4\_plot\_contour\_lines.gp
- 実行方法：gnuplot heat4\_plot\_contour\_lines.gp
- ファイル名やパラメータ等を自由に変更してその効果を試せ。

# 結果の例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

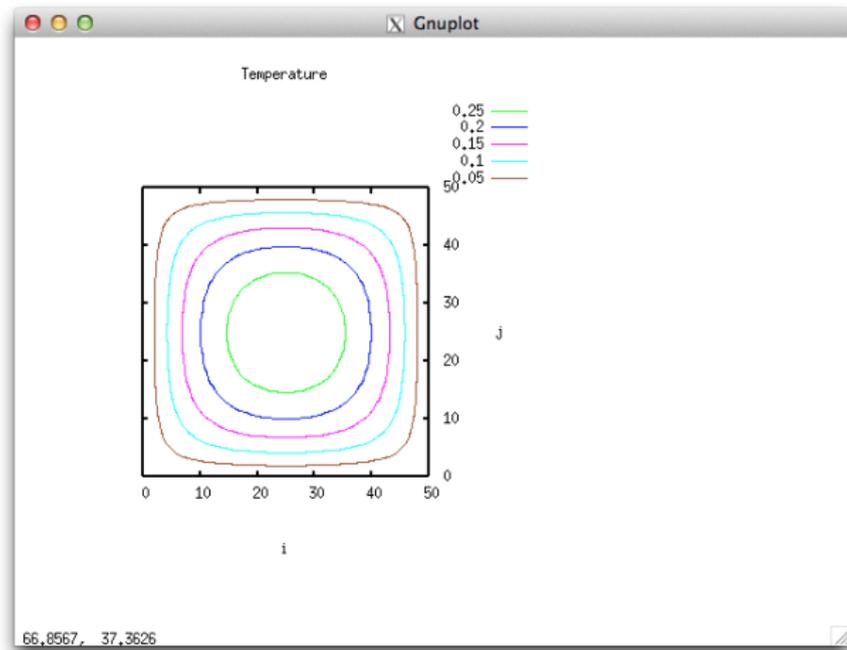
並列化

時間計測

課題

来週の演習と

課題



# 色分布による可視化（静止画）

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- 等高線を描く代わりに正方形領域内部各点の温度を色で表現することも可能である。
- 実際に描いてみよう。
- gnuplot のサンプルスクリプトは次のとおり。

# heat4\_plot\_contour\_colors.gp

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
#
# A sample gnuplot script: heat4_plot_contour_colors.gp
#
#   [ color contours ]
#
# set size square                # same side lengths for x and y
set size 0.65, 1                 # same side lengths for x and y
set xlabel "i"                   # x-axis
set ylabel "j"                   # y-axis
set xrange[0:50]                 # i-grid min & max
set yrange[0:50]                # j-grid min & max
set palette defined (0 "blue", 0.15 "red", 0.3 "yellow")
set nosurface                    # do not show surface plot
unset ztics                      # do not show z-ticks
set pm3d at b                    # draw with colored contours
set view 0,0                     # view from the due north
set title "Temperature "
plot "../data/temp.2d.0000" using 1:2:3
```

# 結果の例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

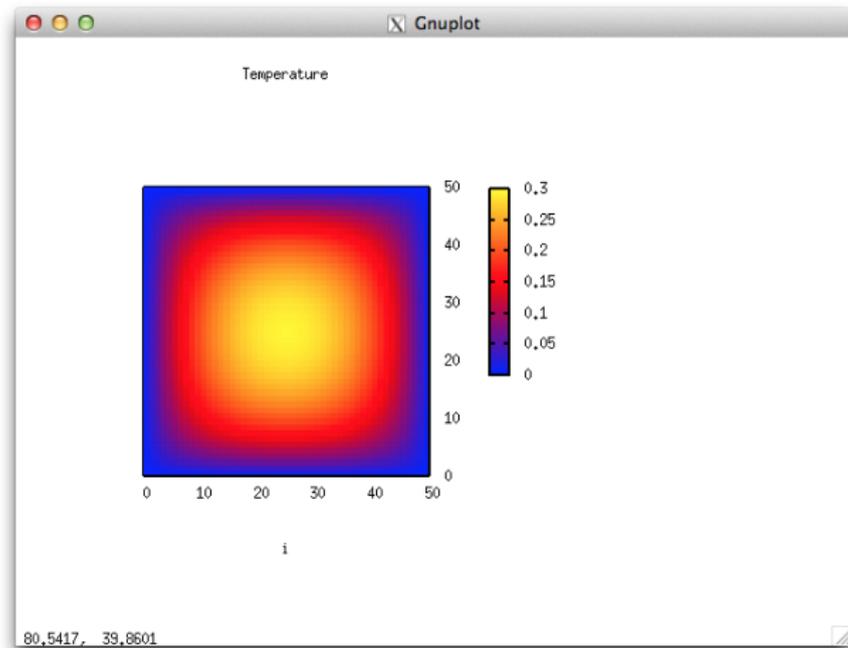
並列化

時間計測

課題

来週の演習と

課題



# gnuplot による鳥瞰図（静止画）

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- 2次元温度分布  $T(x,y)$  を高さ ( $z$ ) で表すことも可能である。
- この時の描画は3次的に行う必要がある。
- 空をとぶ鳥から見下ろしたような図は一般に鳥瞰図 (bird's eye view) と呼ばれる。

# gnuplot スクリプト

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

ファイル名 : plot4\_plot\_birdseyeview.gp

```
#
# a sample gnuplot script: plot4_plot_birdseyeview.gp
#
#   [ Bird"s Eye View ]
#
# set size square           # same side lengths for x and y
set size 0.65, 1
set xlabel "i"             # x-axis
set ylabel "j"             # y-axis
set xrange[0:50]           # i-grid min & max
set yrange[0:50]           # j-grid min & max
set contour base           # enables contour lines
set cntrparam levels 10    # draw 10 contours
# set palette defined (0 "blue", 0.15 "red", 0.3 "yellow")
# set pm3d                   # draw with colored contour
set title "Temperature  "
splot "../data/temp.2d.0000" using 1:2:3 w l
```

# 結果の例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

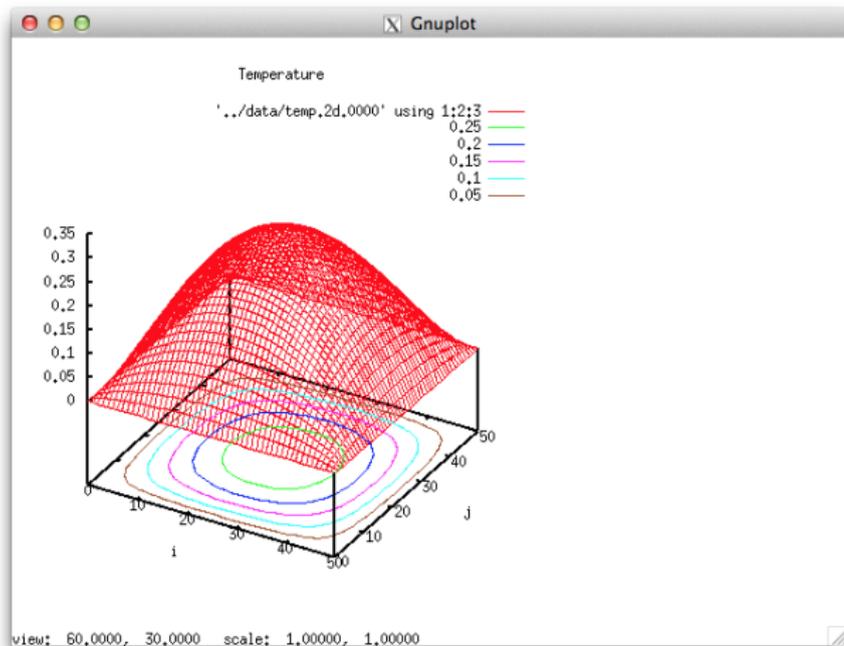
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

マウスで回転できる！



# gnuplot による鳥瞰図（回転のアニメーション）

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

gnuplot では `view` というパラメータで視線の方向（「鳥」がどの方向から見下ろしているか）を指定することが出来る。  
この `view` パラメータを変更したアニメーションを作ってみよう。

# gnuplot スクリプト生成プログラム

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

heat4\_plot\_rotating\_birdseyeview\_generator.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

**ジョブキュー**

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# ジョブキュー

# ジョブキュー

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

これまでジョブスクリプトでは、

```
#PJM -L "rscgrp=small"
```

としてきた。これはジョブキューの種別の指定。

$\pi$ -computer のジョブキュー：

- small: 12 ノードまで
- large: 84 ノードまで      ⇒ 最大  $84 \times 16 = 1344$  並列

今日と来週、この講義の時間は large キューが諸君だけに解放されている。

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

**ハイブリッド  
並列化**

時間計測

課題

来週の演習と  
課題

# ハイブリッド並列化

# MPI + OpenMP = ハイブリッド並列化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

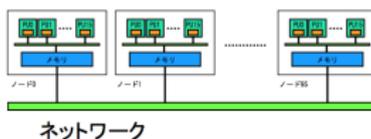
- 一つのノード（プロセッサ）に一つの MPI プロセスを走らせる。
- 各 MPI プロセスで OpenMP によるスレッド並列をする。
- 計算時間の最もかかる do-loop 部分（いまの場合はヤコビ法の部分）だけを本演習の「OpenMP を使った並列計算」で学んだ方法でコアの数（ $\pi$ -computer の場合は 1 プロセッサあたり 16 コア）だけ fork させて計算すればよい。

## 本演習の講義資料から引用

### 本演習で用いる並列計算機

#### ■ 富士通 FX10

- 16コア／プロセッサ
- 1プロセッサ／ノード
- メモリ32GB /ノード
- 全96ノード



富士通FX10(96ノード)

#### ■ 本セミナーでの利用法

- 各ノードを使い, OpenMP の練習
- 複数ノードを使い, MPI の練習
  - この場合, ノード内も含めて MPI で並列化

# ハイブリッド並列化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

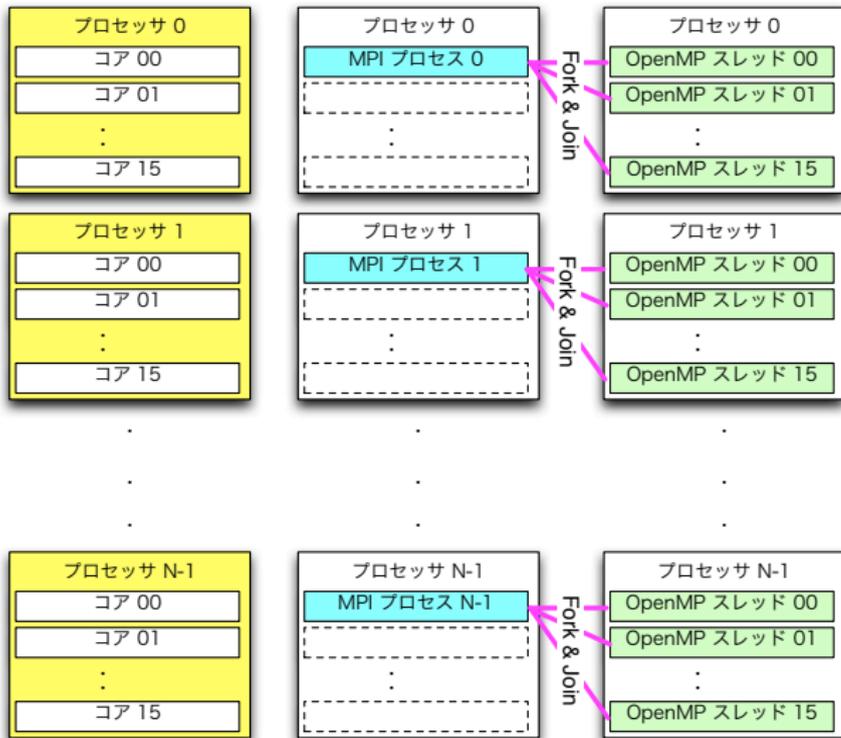
ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

**時間計測**

課題

来週の演習と  
課題

# 時間計測

## 計算時間を測る方法

- CPU 時間

- Fortran95 ⇒ `cpu_time()`

- 実時間 (wall clock time)

- MPI ⇒ `MPI_WTIME()`

- OpenMP ⇒ `omp_get_wtime()`

- Fortran90 ⇒ `system_clock()`

# heat5.f90

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

これまで使ってきた heat4...f90 に、system\_clock() 関数を使った時間計測モジュール stopwatch\_m を組み込んだ。

```
!  
! heat5.f90  
! + module stopwatch, to monitor time.  
! + many calls to stopwatch__stt and __stp.  
! - data output calls for profile 1d and 2d (commented out)  
!! usage (on pi-computer)  
!! 1) mkdir ../data (unless there is already.)  
!! 2) mpifrtpx -O3 heat5.f90 (copy un to u is slow in some cases)  
!! 3) pjsub heat5.sh
```

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

heat5.f90 をコンパイル・実行せよ。実行方法は コードの冒頭、usage を参照。

—

注意：このコンパイラでは-O3 オプションを付けないと以下の部分の処理（次のページの stopwatch 出力の copy un to u ラベルの部分）が遅い。

```
u(1:NGRID,jj%stt:jj%end)=un(1:NGRID,jj%stt:jj%end)
```

# 実行例

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
#####
job start at Tue Jul 16 21:07:29 JST 2013
#####
# myrank= 3  jj%stt & jj%end = 751 1001
# myrank= 0  jj%stt & jj%end = 1 250
# myrank= 2  jj%stt & jj%end = 501 750
# myrank= 1  jj%stt & jj%end = 251 500
//=====<stop watch>=====\\
        profile 1d:    0.000 sec
        main loop:    8.334 sec
        mpi sendrecv: 0.409 sec
                jacobi: 4.103 sec
        copy un to u: 3.799 sec
-----
                Total:    8.386 sec
\\=====<stop watch>=====//
#####
job end at Tue Jul 16 21:07:39 JST 2013
```

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# 課題

# 課題

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

heat5.f90 で中心部分（最も時間のかかる部分）の do-loop を OpenMP でスレッド並列化したコードを作れ。それを heat6.f90 とせよ。

—  
注意：heat5.f90 で行っていた便利な配列演算

```
u(1:NGRID, jj%stt:jj%end)=un(1:NGRID, jj%stt:jj%end)
```

の部分は、omp parallel do では並列化されないので（本演習の OpenMP の回参照）、2重 do loop に展開する必要がある。

# 【提出方法】

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

diff heat5.f90 heat6.f90 | mail kage

ただし、次に述べる方法でハイブリッド並列化の効果（スケールリング）を確認した上で提出すること。

# 演習

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- ( 1 ) コンパイル `mpifrtpx -Kopenmp heat6.f90`
- ( 2 ) ジョブスクリプト `heat6.sh` のジョブキュー指定を `large` とする
- ( 3 ) `pjsub heat6.sh` ジョブ投入
- ( 4 ) 結果をみる

# ジョブスクリプト：heat6.sh（中心部分）

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

```
#!/bin/bash
#PJM -N "heat6"
#PJM -L "rscgrp=small"
#PJM -L "node=4"
#PJM -L "elapse=02:00"
#PJM -j
export FLIB_CNTL_BARRIER_ERR=FALSE
.
.
for opn in 1 2 4 8 16
do
export OMP_NUM_THREADS=$opn
echo "# omp_num_threads = " $opn
mpiexec -n 4 ./a.out
done
.
```

# スケーリングの確認

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

heat6.f90 を使い、

- 1 ノード  $M$  ( $\leq 16$ ) スレッドのハイブリッド並列で、
- $N$  ( $\leq 84$ ) ノードを使い、
- スレッド総数  $P(= M \times N)$  v.s. 計算速度  $S$  のグラフ ( $S$  は stopwatch module の出力の “Total” で表示される秒数の逆数と定義する) を、

gnuplot で描く。並列化のスケールが悪い時には、格子点数 NGRID を増やすとよいかも。

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

## 来週の演習と課題

# 来週

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- 来週の演習はなし。
- ただし、諸君にはこの時間（24日木曜3限）だけ large キューが解放されている。 $\pi$ -computer で大規模な計算を贅沢にできる最後の（？）チャンスである。
- そこで… 挑戦課題。これは任意提出。提出しなくても構わない。

挑戦課題：前々ページに書いた heat6.f90 のスケーリング測定を実行し、レポートを提出せよ。

# レポートは（もし提出するなら）以下の内容で

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

- (a) 氏名・学籍番号
- (b) 使用した NGRID, ノード数  $N$ , ノードあたりのスレッド数  $M$  (=OMP\_NUM\_THREADS) の値
- (c) gnuplot で描いたグラフのキャプチャ図
- (d) そのグラフについての考察

提出先： [echo pj.ca.u-ebok.trop@egak](mailto:echo pj.ca.u-ebok.trop@egak) | rev

ファイルフォーマット： pdf

メールタイトル： 計算科学演習 2014 レポート

締めきり： 2014 年 7 月 31 日 23:59

くれぐれも注意： 興味のある学生だけ提出すればよい。

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

# アンケート

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

<https://questant.jp/q/NHOJH2NR>

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

## 補遺 A : Flat MPI 並列化

# Flat MPI 並列化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- これまでは1 ノード (1 プロセッサ、16 コア) に一つだけ MPI プロセスを動かしていた。
  - (OpenMP を使ったハイブリッド並列をしない場合) 他の 15 個のコアは遊んでいた。
  - 1 ノード (1 プロセッサ) に 16 個の MPI プロセスを走らせることも可能。
  - 例えば 4 ノード使う場合には合計  $4 \times 16 = 64$  MPI プロセスで並列化。
  - このような並列化を Flat MPI という<sup>1</sup>。
-

# FLAT MPI 並列化

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題



実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

## 補遺 B : 1次元領域分割と2次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

## 補遺 C: 2次元並列化

# 1次元並列

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

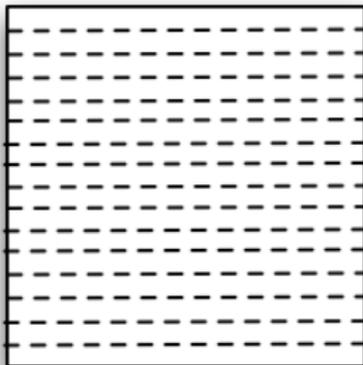
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

引き続き、正方形領域の熱伝導問題（平衡温度分布）を考える。これまでの並列化：1次元領域分割による並列化: 16並列



## 2次元並列

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

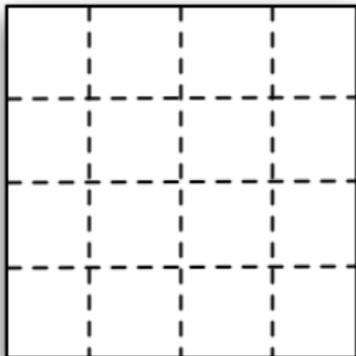
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

これも 16 並列。どちらが速いか？1次元領域分割と2次元領域分割どちらを採用すべきか？



# そもそも1次元分割ができない場合

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

- 格子数 NGRID 61
- 並列プロセス数 100
  - 1次元分割不可能
  - 2次元分割なら可能（総格子点数 3721）

# 1次元分割と2次元分割のちがい

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

格子点数 NGRID が十分大きければ 1次元分割と2次元分割は同じか？

- プロセスあたりの計算量は同じ
- 通信量が違う

# 計算と通信

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

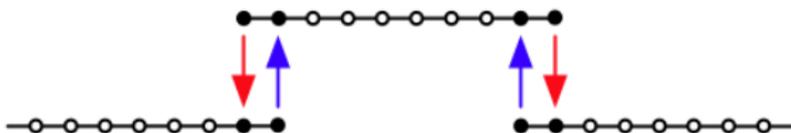
課題

来週の演習と  
課題

1次元空間を格子点で離散化した上で、MPIでプロセス間通信を行う場合を考える。

計算用格子点（白丸）：6個

通信用格子点（黒丸）：4個



# 計算と通信

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

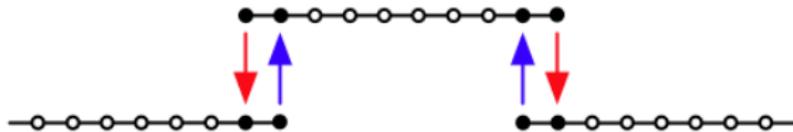
計算格子には2種類ある。

- 1 その上で計算だけを行う格子。
- 2 MPI通信のデータを送ったり、受けたりする格子である。

(一番外側から2番目の格子は計算も通信も行う。)

計算用格子点 (白丸) : 6個

通信用格子点 (黒丸) : 4個



通信は時間がかかるので、通信を行う格子点は少ないほうが望ましい。

# 2次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

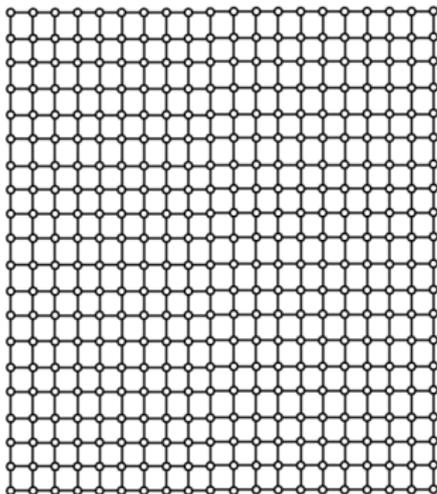
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

正方形領域を 400 個の格子点で離散化した場合



## 2次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

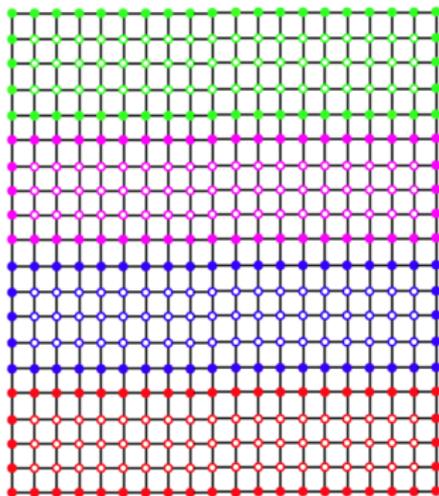
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

4つのMPIプロセスで並列化。1次元領域分割。



赤のプロセスの通信担当格子点の数：46

## 2次元領域分割

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

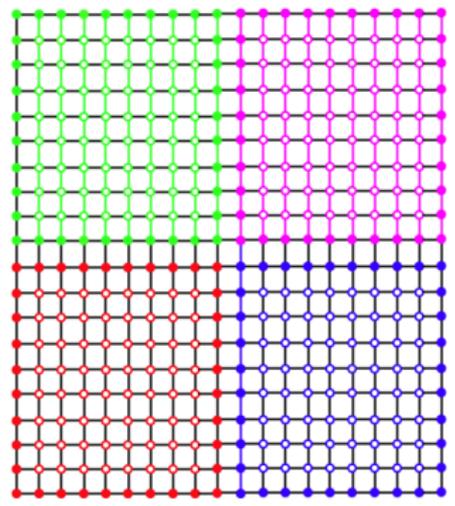
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

2次元領域分割の場合、同じく4つのMPIプロセスで並列化。



赤のプロセスの通信担当格子点の数：38

# 2次元領域分割の方法

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

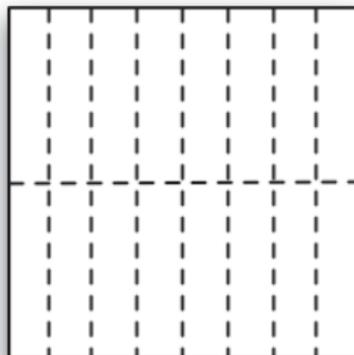
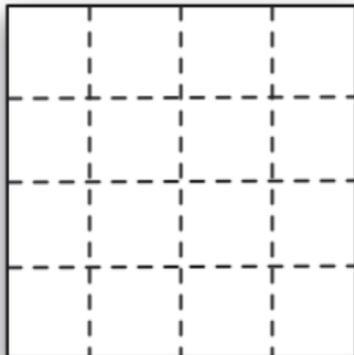
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

どちらがよいか？



# 2次元領域分割の方法

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

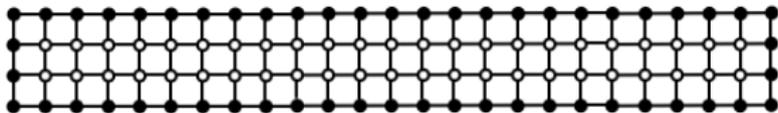
課題

来週の演習と  
課題

計算格子（白丸）：46 個

通信格子（黒丸）：54 個

合計：100 個



## 2次元領域分割の方法

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

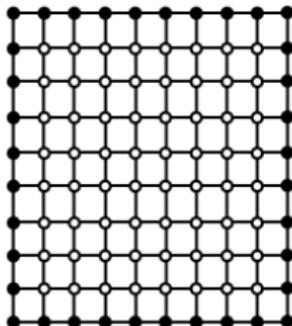
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

計算格子 (白丸) : 64 個  
通信格子 (黒丸) : 36 個  
合計 : 100 個



面積の等しい長方形の中で、4辺の長さの合計が最も小さいものは正方形。

## 2次元領域分割による並列コード

実践編  
陰山

準備  
問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

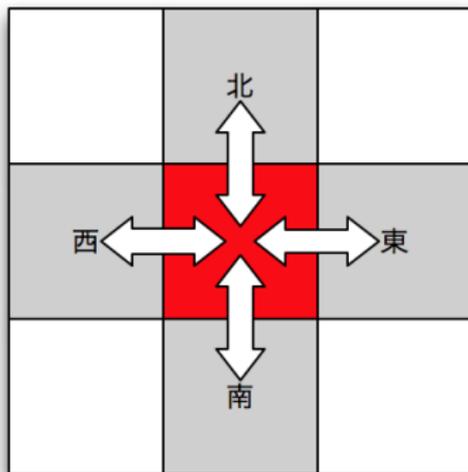
ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

領域分割による並列化を行うときに注意すべき点の一つは、MPI プロセスの配置方法。  
隣同士の通信がもっとも通信速度的に「近い」位置にプロセスを配置することが望ましい。



## 2次元領域分割による並列コード

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

4番のプロセスはランク番号1,3,5,7のプロセスと頻繁に通信する

2	5	8
1	4	7
0	3	6

## 2次元領域分割による並列コード

実践編  
陰山

準備  
問題設定  
熱伝導  
離散化  
差分法  
実装  
並列化  
可視化  
2次元可視化  
ジョブキュー  
ハイブリッド  
並列化  
時間計測  
課題

もしも使用している並列計算機のネットワークの構成上、4番のプロセスはむしろランク番号0,2,6,8のプロセスと通信した方が速い場合には、以下のようにプロセスを配置する方が望ましい。

5	6	7
0	4	8
1	2	3

来週の演習と  
課題

# MPI\_CART\_CREATE

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

MPI関数の一つMPI\_CART\_CREATEを使うと（使用する計算機がネットワークの通信性能に関する情報を提供している場合には）通信効率の点で最適な配置でプロセスを自動的に分配してくれる<sup>2</sup>。

---

# 速度のスケーリング

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド  
並列化

時間計測

課題

来週の演習と  
課題

【参考】 $\pi$ -computer とは別のシステムにおける測定結果

# 【参考】MPI プロセス数と計算速度の関係（例）

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

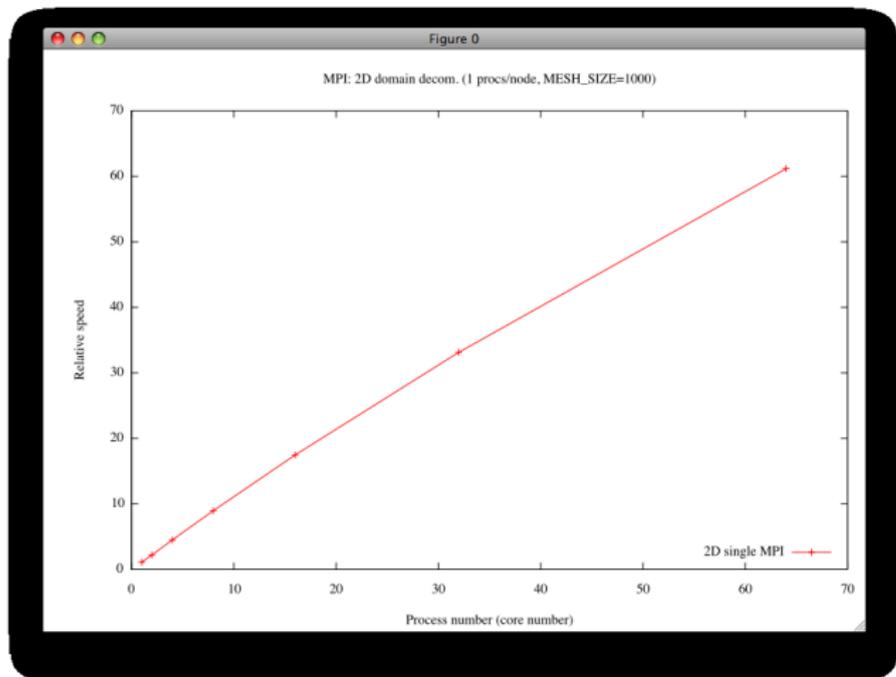
並列化

時間計測

課題

来週の演習と

課題



# 【参考】1次元領域分割と2次元領域分割の比較 (例)

実践編

陰山

準備

問題設定

熱伝導

離散化

差分法

実装

並列化

可視化

2次元可視化

ジョブキュー

ハイブリッド

並列化

時間計測

課題

来週の演習と

課題

