

# 非平衡超伝導ダイナミクスシミュレーションの 並列化 (HPFとMPIの比較)

町田昌彦 叶野琢磨

日本原子力研究所  
計算科学技術推進センター

## 発表内容

### 1、プロジェクト概要

- i) 中性子デバイス開発を先導するシミュレーション
- ii) 超伝導ナノファブリケーションによる新奇物性

### 2、HPFを用いた並列化とチューニング

### 3、性能 (MPIとの比較)

### 4、まとめと結論

# 1、プロジェクト概要

**母体(コンソーシアム):** 戦略創造研究プロジェクト 福山秀敏統括

「超伝導ナノファブリケーションによる新奇物性とその応用」

研究代表者: 石田武和(大阪府立大教授)

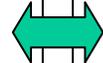
## 実験グループ:

大阪府立大学大学院工学研究科

大阪府立産業技術総合研究所

日本原子力研究所 東海研究所

通信総合研究所 関西先端研



## 理論・シミュレーショングループ:

大阪府立大学大学院工学研究科

東北大学 金属材料研究所

日本原子力研究所 計算科学



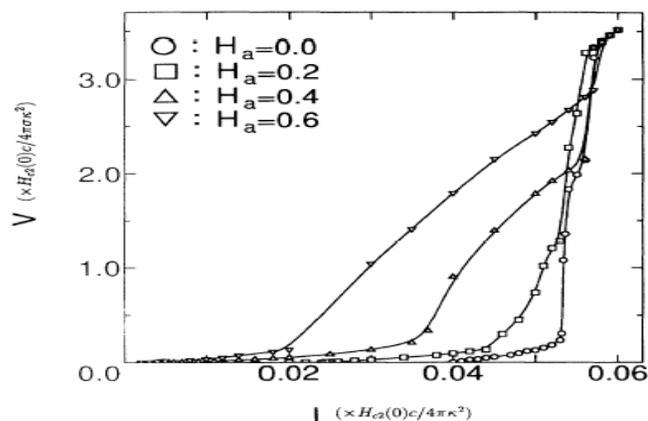
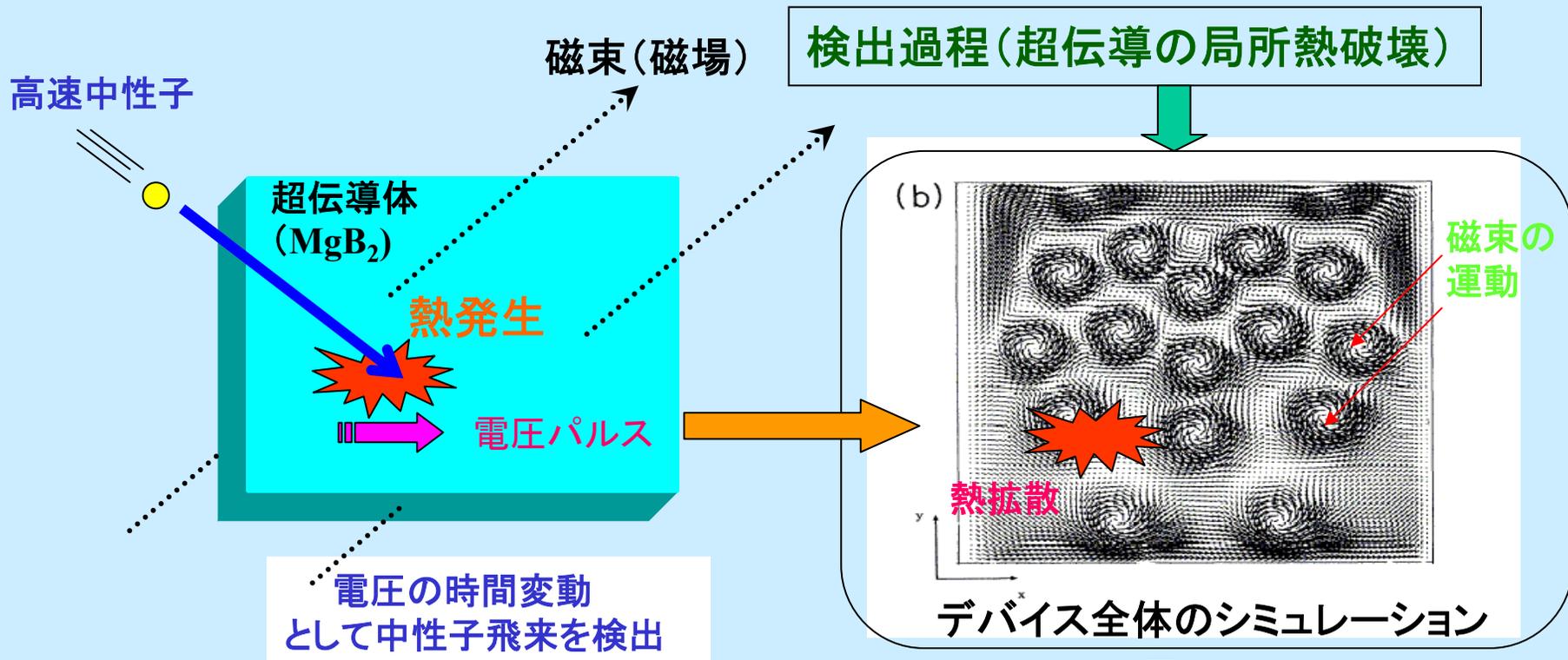
## 地球シミュレータ共同利用プロジェクト提案:

i) 中性子検出デバイス開発を先導するシミュレーション

ii) 超伝導ナノファブリケーションによる新奇物性

**アドバイザー:** 日本原子力学会・大規模シミュレーション専門部会

# i) 中性子検出デバイス開発を先導するシミュレーション



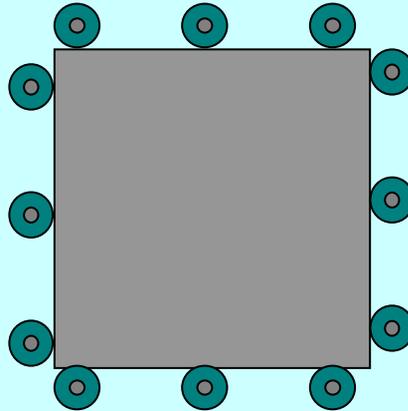
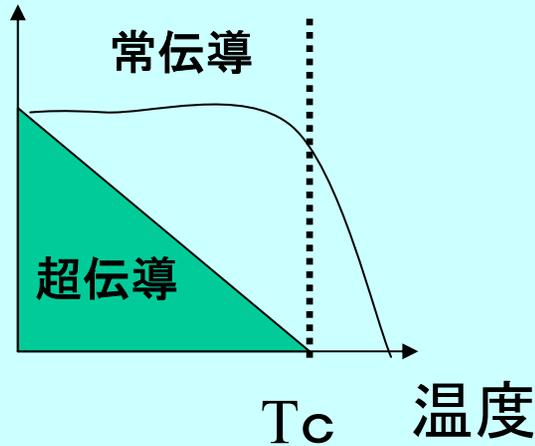
最適デバイス構成を決定  
動作メカニズムの解明  
非平衡超伝導

## ii) 超伝導ナノファブリケーションによる新奇物性

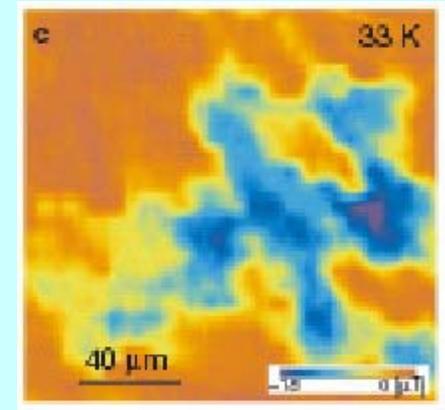
高温高磁場下の超伝導状態

超伝導機構

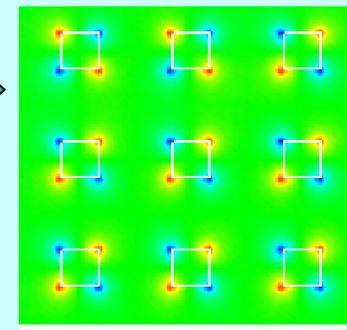
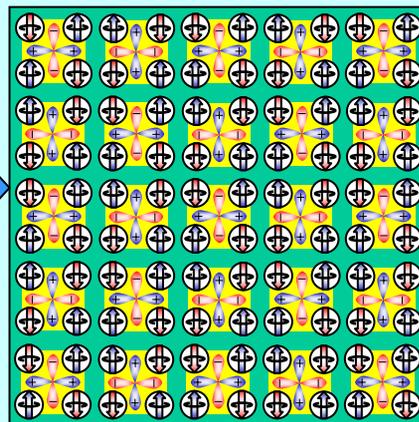
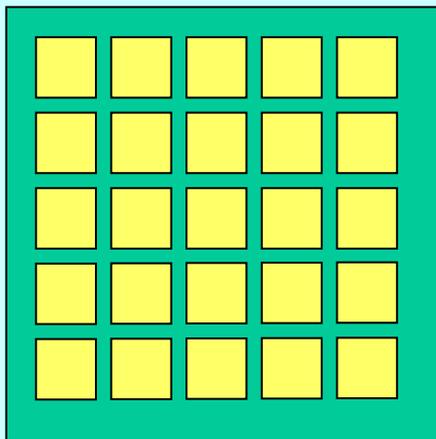
磁場



高温  
磁場



高温超伝導体と金属超伝導体の超伝導マトリックス



自発半磁束

# シミュレーション手法

時間依存のギンツブルク・ランダウ方程式  $\longrightarrow$  超伝導電子密度のダイナミクス

$$D^{-1} \left[ \frac{\partial}{\partial t} + i \frac{2e\varphi}{\hbar} \right] \Delta + \xi^{-2} (|\Delta|^2 - 1) \Delta + \left[ \frac{\nabla}{i} - \frac{2e}{\hbar c} \mathbf{A} \right]^2 \Delta = 0, \quad \text{テーマ i) ii)}$$

マックスウエル方程式  $\longrightarrow$  電磁場のダイナミクス

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{j} \quad \mathbf{j} = \sigma \left[ -\nabla\varphi - \frac{1}{c} \frac{\partial \mathbf{A}}{\partial t} \right] + \text{Re} \left[ \Delta^* \left[ \frac{\nabla}{i} - \frac{2e}{\hbar c} \mathbf{A} \right] \Delta \right] \frac{\hbar c^2}{8\pi e \lambda^2}$$

エネルギー保存則  $\longrightarrow$  温度と熱のダイナミクス テーマ i)

$$C_v \frac{dT}{dt} + \frac{dF}{dt} + \text{div}(j_n^Q + J_s^Q) + W = 0$$

期待される性能

高い演算密度による高いベクトル性能  
空間分解による高い並列性能

## シミュレーション計画

4月

6月

8月

10月

12月

2月

4月

プロトタイプコード開発  
(TDGL・Maxwell連立版)

チューニング  
(地球シミュレータ)

MPI(HPF)

シミュレーション  
テーマii)へ

実コード開発  
(エネルギー保存則含む)

チューニング

HPF

シミュレーション  
テーマi)へ

## プロトタイプコードのチューニング

MPIによる並列化 18.4Tflops

512ノード利用  
(ピーク:32.7Tflops)  
56%ピーク比

HPF利用

4.39Tflops  
4.56Tflops(MPI)

128ノード利用  
(ピーク:8.19Tflops)  
53%ピーク比  
96%対MPI比

# i) 中性子検出デバイス開発を先導するシミュレーション

## シミュレーション結果の現状

### 初期的結果

任意(小~大)の  
ホットスポット

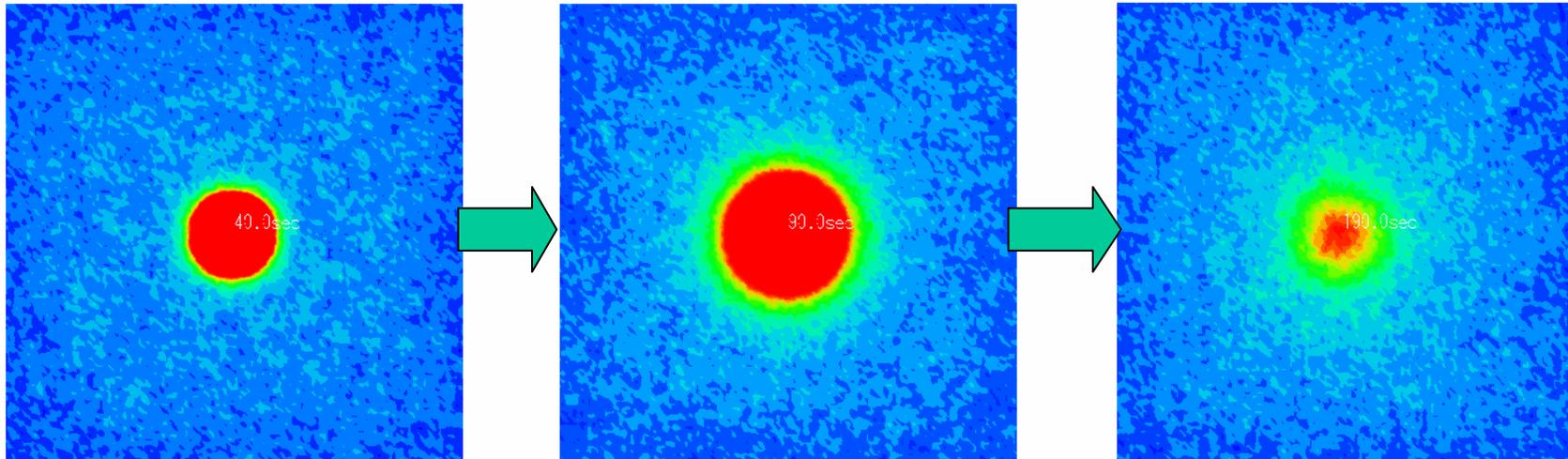
レーザー等  
で実現

小規模計算  
で再現

### 目標とする結果

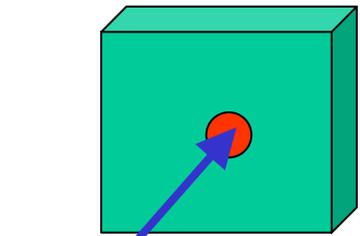
核反応に対応する  
ホットスポット

大規模計算が必要

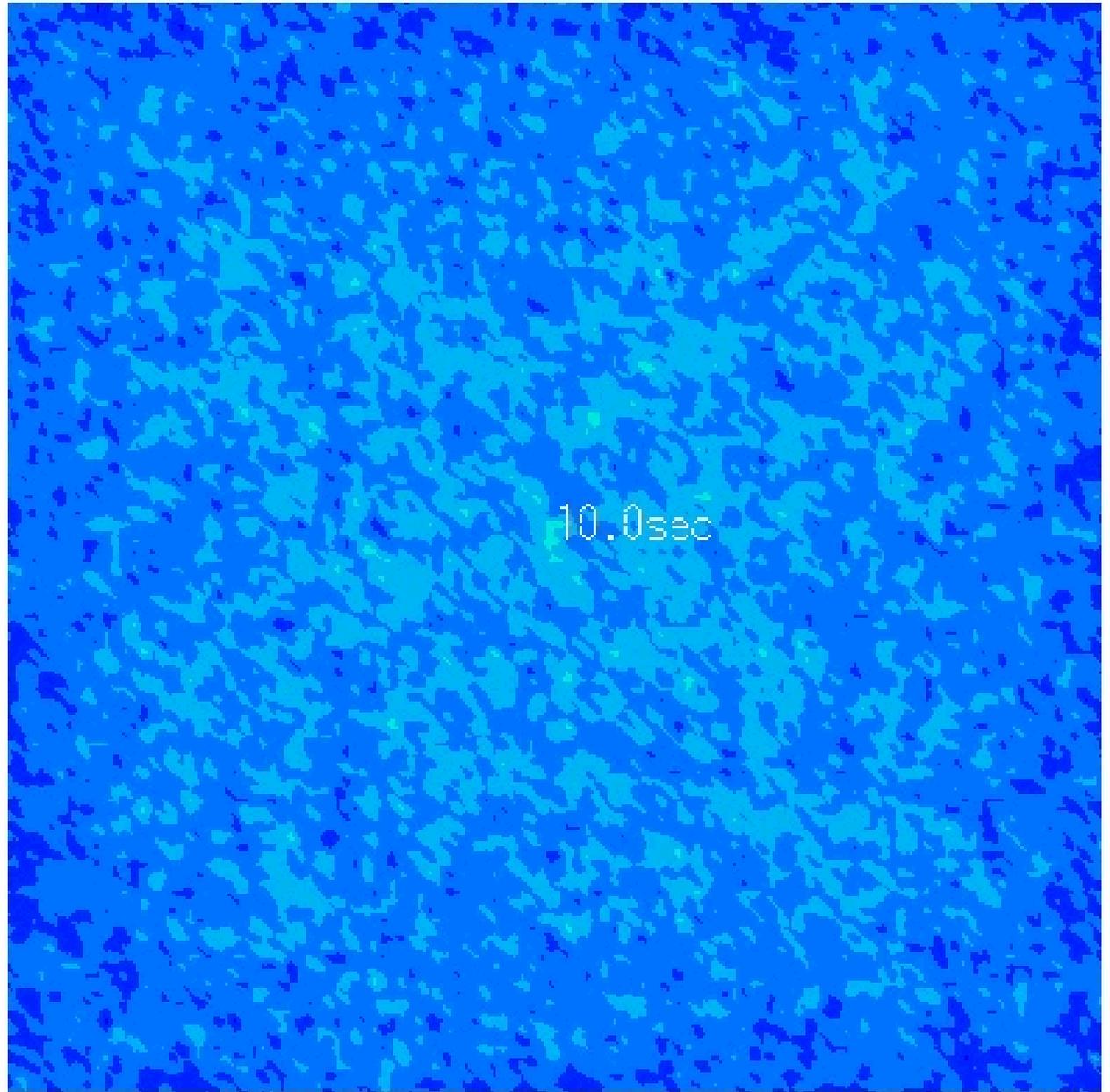


超伝導秩序パラメータのダイナミクス

# シミュレーション 結果(初期)

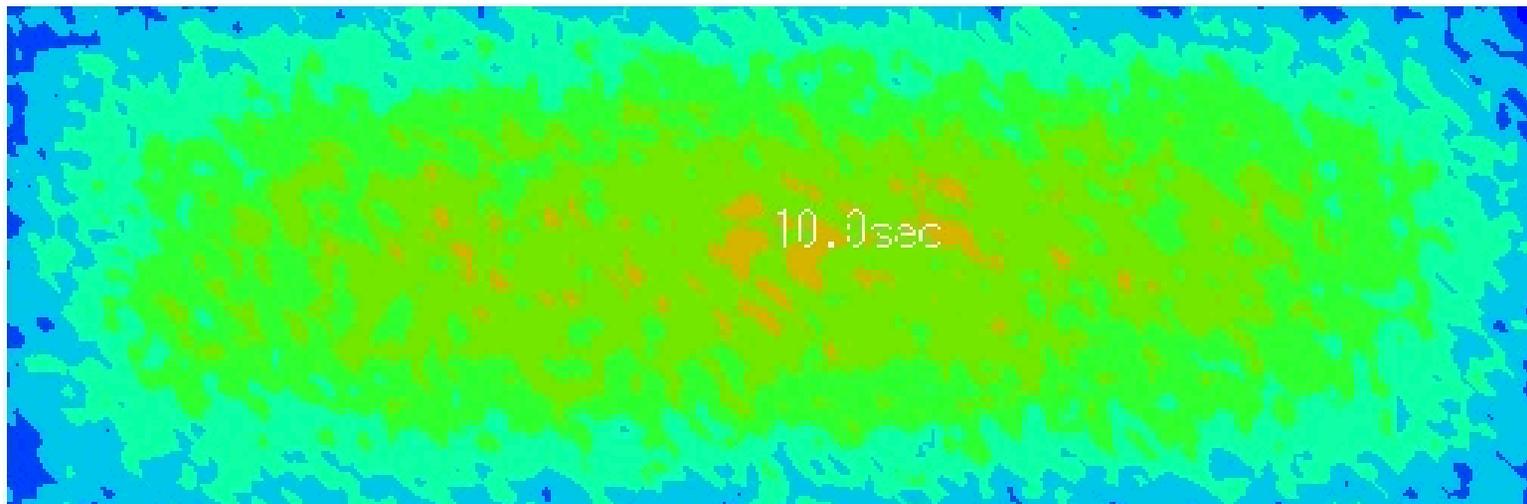
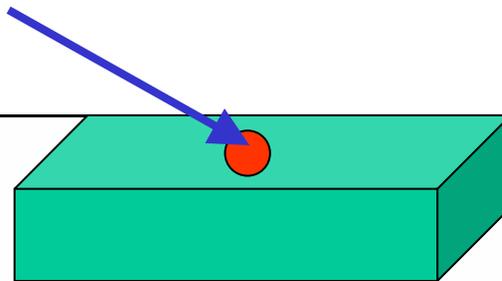


レーザー照射



10.0sec

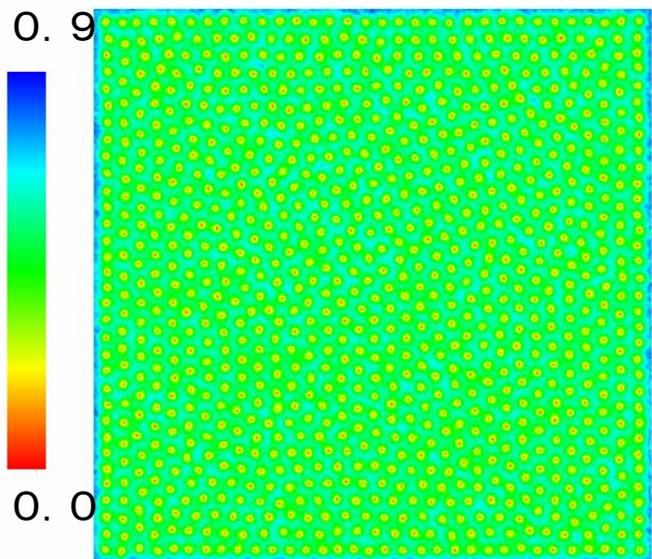
レーザー照射



## ii) 超伝導ナノファブリケーションによる新奇物性

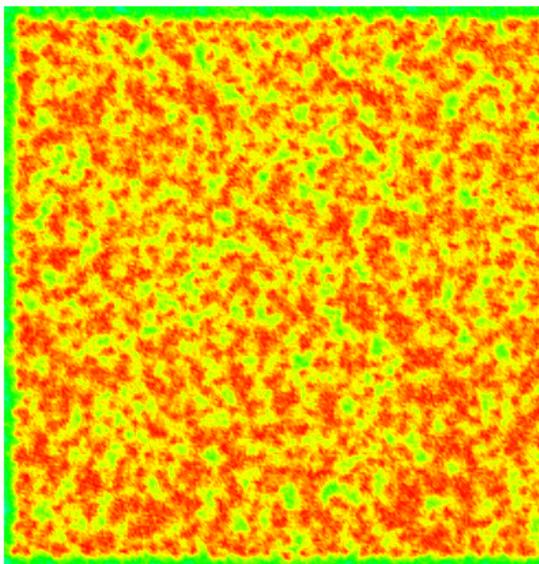
磁場下  $H=0.4H_{c2}(0)$

$T=10\text{K}$

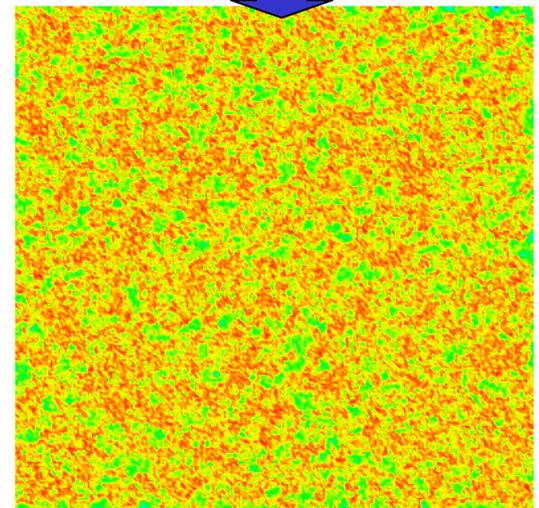
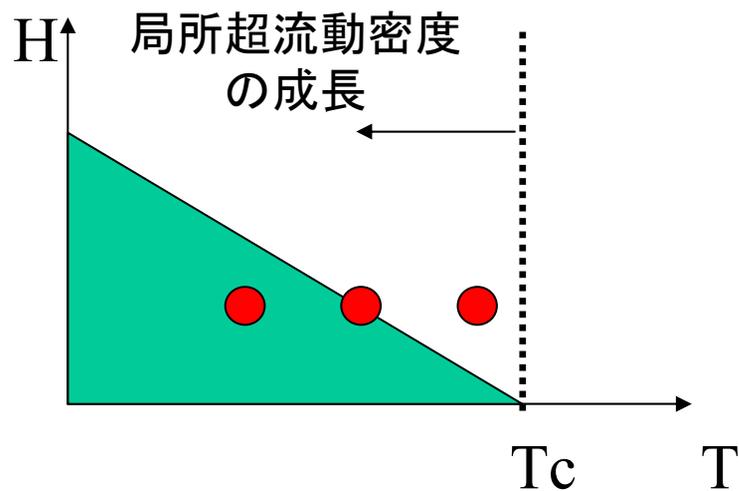
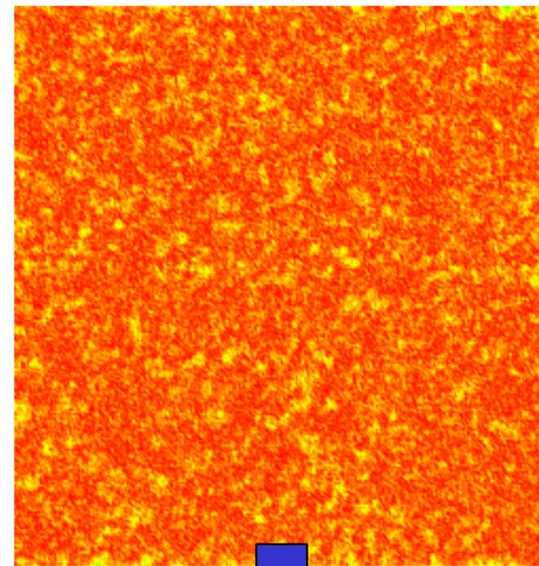


高温高磁場下の超伝導状態

$T=18\text{K}(H_{c2})$



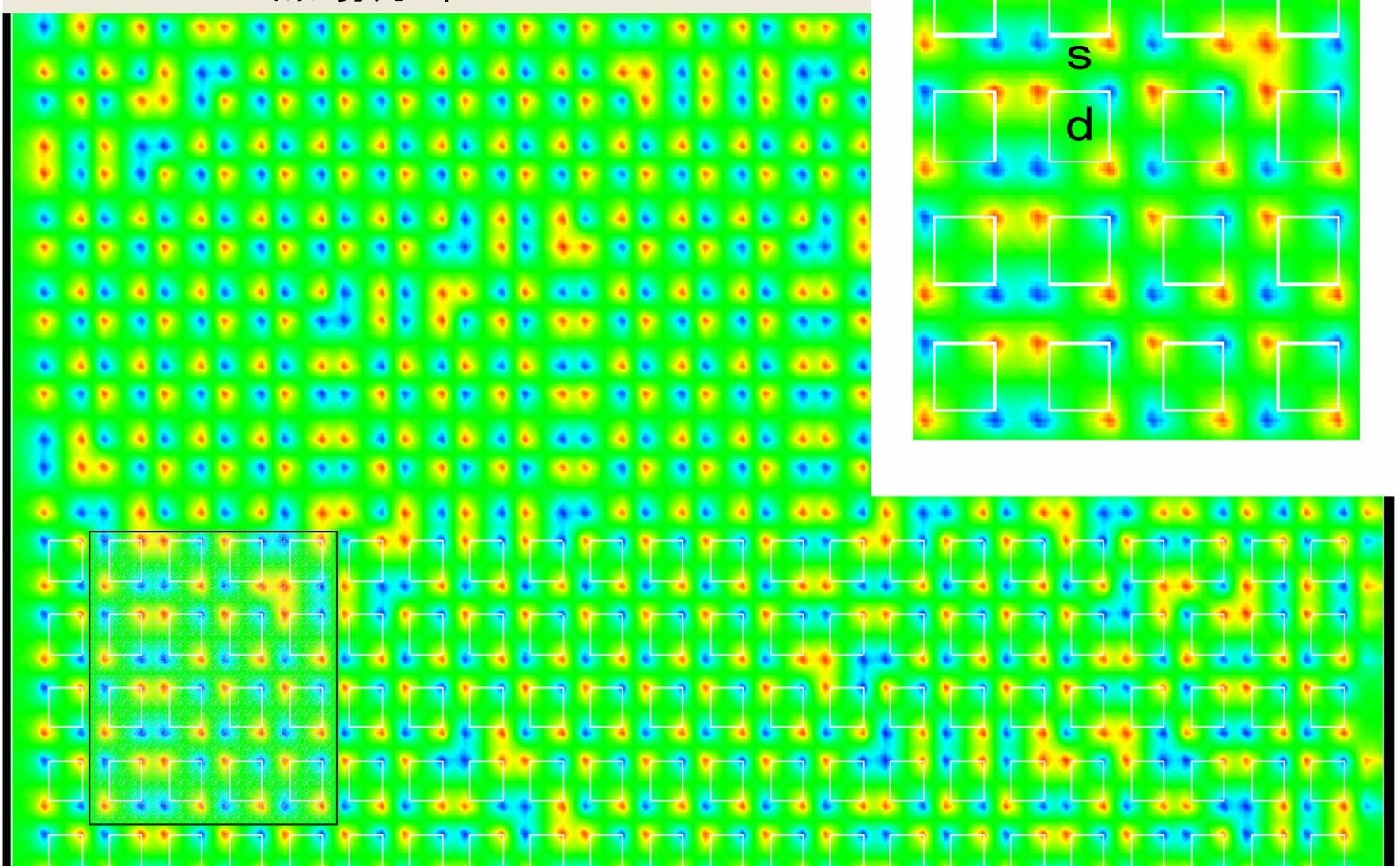
$T=25\text{K}$



# 高温超伝導体と金属超伝導体の超伝導マトリックス

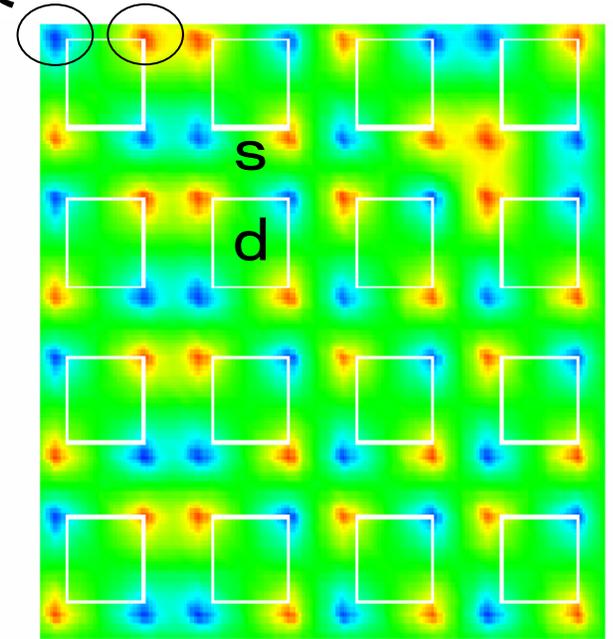
## 大規模シミュレーション例

磁場分布



半磁束

反半磁束



# 2、HPFを用いた並列化とチューニング

## プログラムの構成と特徴

初期条件設定

**Do** 時間=始まり、終わり

時間依存のギンツブルク・  
ランダウ方程式



マックスウエル方程式



エネルギー保存の方程式



測定

**Do End**

時間発展を含む  
偏微分方程式

高い演算密度  
による高いベクトル  
性能が期待できる。

# 例: 時間依存のギンツブルク・ランダウ方程式

```
C***** NEXT TIME STEP CALCULATE *****
C***** SIMPLE EULER METHOD *****
C
  DO 112 J=1, M      空間に対するループ
  DO 111 I=1, L
C
C
      FDELTR(I, J, K)=BDELTR(I, J, K)-DT12*(
&      -TTCC(I, J)
&      *(BDELTR(I, J, K))
&      +(BDELTR(I, J, K)**2+BDELTI(I, J, K)**2)*BDELTR(I, J, K)
&      -(FDERXR(I, J)+BDERXR(I, J))*DX2
&      -(FDERYR(I, J)+BDERYR(I, J))*DY2
&      -(FDERZR(I, J)+BDERZR(I, J))*DZ2
&      )
      FDELTI(I, J, K)=BDELTI(I, J, K)-DT12*(
&      -TTCC(I, J)
&      *(BDELTI(I, J, K))
&      +(BDELTR(I, J, K)**2+BDELTI(I, J, K)**2)*BDELTI(I, J, K)
&      -(FDERXI(I, J)+BDERXI(I, J))*DX2
&      -(FDERYI(I, J)+BDERYI(I, J))*DY2
&      -(FDERZI(I, J)+BDERZI(I, J))*DZ2
&      )
111  CONTINUE
112  CONTINUE
```

# 1PEでの実行性能： ピーク性能(8Gflops)

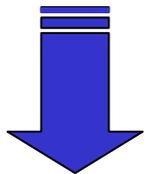
TDGL方程式： 5.911Gflops

マックスウエル方程式： 5.105Gflops

磁場の計算： 5.472Gflops

トータル： 5.260Gflops

6割を超える  
パフォーマンス



チューニングの目標

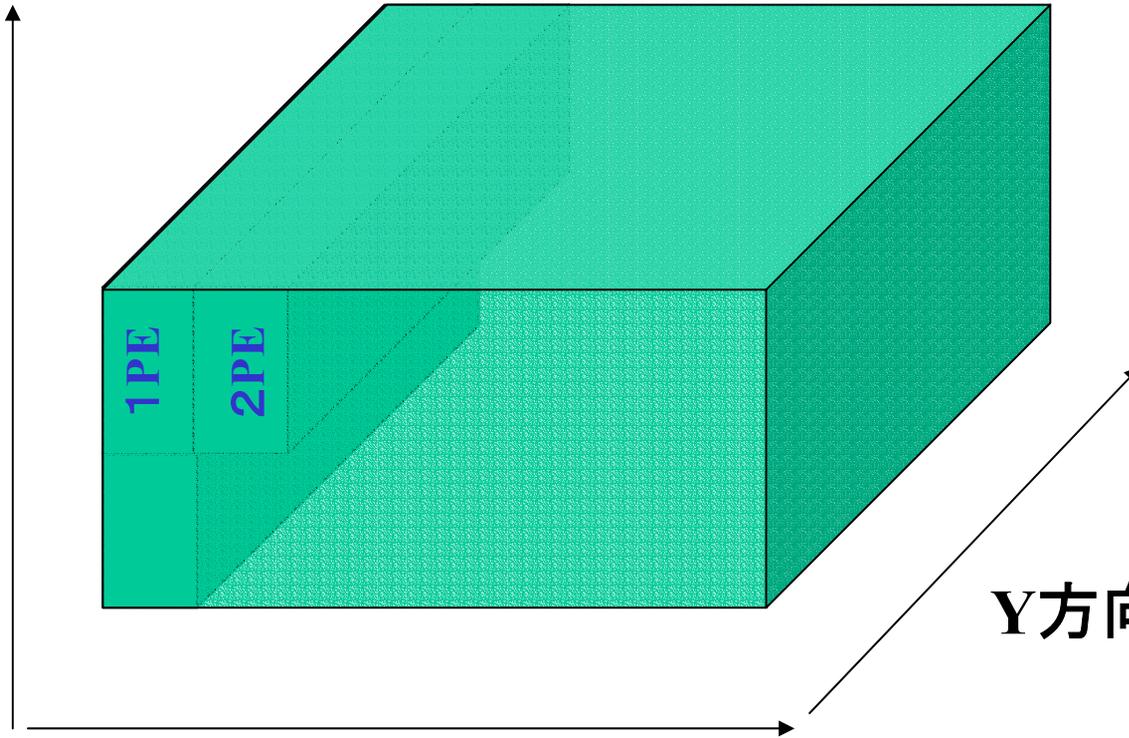


ベクトル性能を落とさずに並列化する。

i) シミュレーション対象の分割(並列化)

MPI  
HPF

Z方向 (1200)



Y方向 (2400)

X方向 (2400)

XZ方向分割

Y方向残す:ベクトル化(長いベクトル長)



# インクルードファイル(場の変数定義)

```
!- All Process number is 1024 = 32 * 32
INTEGER, DIMENSION(32) :: N_PE_J
INTEGER, DIMENSION(32) :: N_PE_K
DATA (N_PE_J(I), I=1, 32) /25*69, 7*68/ !-distoribute 2200+1 elems
DATA (N_PE_K(I), I=1, 32) /25*35, 7*34/ !-distoribute 1400 +1 elems
```

!-kano-

```
COMMON/DELTA1/BDELTR(L, M, N), BDELTI(L, M, N)
COMMON/UXVPX/FUXVPR(L, M, N), FUXVPI(L, M, N)
COMMON/FLUXX/BX(0:L, 0:M, 0:N)
```

!HPF\$ PROCESSORS PE(32, 32) !-1024PEs

!HPF\$ TEMPLATE TEMP(0:L, 0:M, 0:N)

!HPF\$ DISTRIBUTE TEMP(\*, GEN\_BLOCK(N\_PE\_J), GEN\_BLOCK(N\_PE\_K)) ONTO PE

2次元分割

\* 各変数配列に  
2次元分割指定

!HPF\$ ALIGN (I, J, K) WITH TEMP(I, J, K) :: BDELTR, BDELTI

!HPF\$ SHADOW(0, 1:1, 1:1) :: BDELTR, BDELTI

超伝導電子密度

!HPF\$ ALIGN (I, J, K) WITH TEMP(I, J, K) :: FUXVPR, FUXVPI

!HPF\$ SHADOW(0, 0:1, 0:1) :: FUXVPR, FUXVPI

ベクトルポテンシャル

!HPF\$ ALIGN (I, J, K) WITH TEMP(I, J, K) :: BX

!HPF\$ SHADOW(0, 1:0, 1:0) :: BX

磁場

\* 明示的にシャドウを指定  
(通信量削減を意識)

## 2) 物理量測定(総和等の計算)をMAINルーチンに 集中記述

### MAINルーチン(物理量測定)

### 磁場(BX、BZ)の総和計算

```
TOTBX=0.0
TOTBZ=0.0
!HPF$ INDEPENDENT, REDUCTION(TOTBX, TOTBZ)
      DO 334 K=1, N
        DO 335 J=1, M
          DO 336 I=1, L
            TOTBX=TOTBX+BX(I, J, K) / (DY*DZ)
            TOTBZ=TOTBZ+BZ(I, J, K) / (DX*DY)
          336 CONTINUE
        335 CONTINUE
      334 CONTINUE
      WRITE(6, '(I5, 4E15.7)') IT, TOTBX, TOTBZ, AEMHX, AEMHZ
```

\* REDUCTIONにより  
総和並列計算を指示

### 3) 方程式ソルバーでは、通信の要不要を明示的に指定

```
C===== X COMPONENT =====
```

```
!HPFJ REFLECT FUXVPR, FUXVPI, FUYVPR, FUYVPI, FUZVPR, FUZVPI
```

\* 各変数のシャドウ  
のアップデート

```
!HPF$ INDEPENDENT, NEW(TT1, TT2, TT3, TT4)
```

```
DO 13 K=1, N-1
```

```
DO 12 J=1, M-1
```

```
DO 11 I=1, L
```

```
!HPF$ ON HOME(BX(I, J, K)), LOCAL(FUYVPR, FUYVPI, FUZVPR, FUZVPI) BEGIN
```

```
TT1 = FUYVPR(I, J, K)*FUYVPR(I, J, K+1)
```

```
& + FUYVPI(I, J, K)*FUYVPI(I, J, K+1)
```

```
TT2 = FUYVPR(I, J, K)*FUYVPI(I, J, K+1)
```

```
& - FUYVPI(I, J, K)*FUYVPR(I, J, K+1)
```

```
TT3 = FUZVPR(I, J, K)*FUZVPR(I, J+1, K)
```

```
& + FUZVPI(I, J, K)*FUZVPI(I, J+1, K)
```

```
TT4 = FUZVPR(I, J, K)*FUZVPI(I, J+1, K)
```

```
& - FUZVPI(I, J, K)*FUZVPR(I, J+1, K)
```

```
BX(I, J, K) = TT1*TT4 - TT2*TT3
```

\* LOCALにより  
通信必要なしを明示

```
!HPF$ END ON
```

```
11 CONTINUE
```

```
12 CONTINUE
```

```
13 CONTINUE
```

高次元の計算では、明示的  
指定が本質的  
各次元に対し繰り返す

# 使用HPF指示文

変数配列指定用  
インクルード

**PROCESSOR  
TEMPLATE  
DISTRIBUTE  
SHADOW**

MAINルーチン  
物理量測定

**INDEPENDENT  
REDUCTION**

方程式ソルバー  
ルーチン

**REFLECT**  
**INDEPENDENT  
NEW**  
**ON HOME**  
**LOCAL**

\* ES利用により性能追求のため

HPF/JA1.0拡張仕様: LOCAL、REFLECTを  
用い通信位置を指示。

### 3、性能評価(MPIとの比較)

単位Gflops

	1024PE(128node)			512PE(64node)		
HPF	4391	53%	96%(MPI比)	2358	58%	97%(MPI比)
MPI	4560	55%		2413	59%	
ピーク	8192			4046		

#### HPFの印象

- \* 優れた性能を短期間で達成可
- \* 通信の制御を明示 (MPIに匹敵する高性能)
- \* 高次元の配列処理に注意 (拡張機能が対応?)

## 4、まとめと結論

- 1) 現在まで、MPI版とHPF版のプロトタイプコードを開発した。HPF版の開発は数日オーダである。
- 2) 通信の制御等によりMPI版に匹敵する高性能を大ノードで実現。
- 3) プロトタイプコードの経験を下にHPF化を先に進めることを検討。

短期間のプロジェクト(開発、チューニング、シミュレーション)では、HPFは貴重な役割を果たす！