

差分法による 弾性波動並列シミュレーション

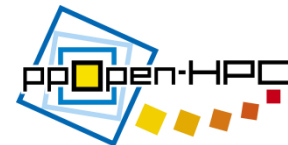
東京大学 情報学環 / 地震研究所

森 太志
(f-mori@eri.u-tokyo.ac.jp)

東京大学 情報学環 / 地震研究所

古村 孝志

本日の目次



- はじめに
 - 有限差分法 (FDM)による弾性波動計算の概要
 - ppOpen-APPL/FDM の概要
 - FX10によるパフォーマンステスト
 - 応用例: 大規模連成計算
- ppOpen-APPL/FDMの演習
 - 利用方法
 - 演習

- 構成方程式

- 弾性体の釣り合いの式 (運動方程式)

$$\rho \ddot{u} = \frac{\partial \sigma_{xp}}{\partial x} + \frac{\partial \sigma_{yp}}{\partial y} + \frac{\partial \sigma_{zp}}{\partial z} + f_p, \quad (p = x, y, z) \quad (1)$$

\ddot{u} : 加速度、 σ : 応力、 ρ : 密度、 f : 外力

- 等方完全弾性体の応力

$$\sigma_{pq} = \lambda(e_{xx} + e_{yy} + e_{zz})\delta_{pq} + 2\mu e_{pq}, \quad (p, q = x, y, z) \quad (2)$$

λ, μ : Lamé定数、 δ : クロネッカーのデルタ

- 歪みは変位の空間微分で求められる

$$e_{pq} = \frac{1}{2} \left(\frac{\partial u_p}{\partial q} + \frac{\partial u_q}{\partial p} \right), \quad (p, q = x, y, z) \quad (3)$$

e : 歪み

- 時間発展により波動伝播を進めるために、式(1)の速度変数を中間変数とする

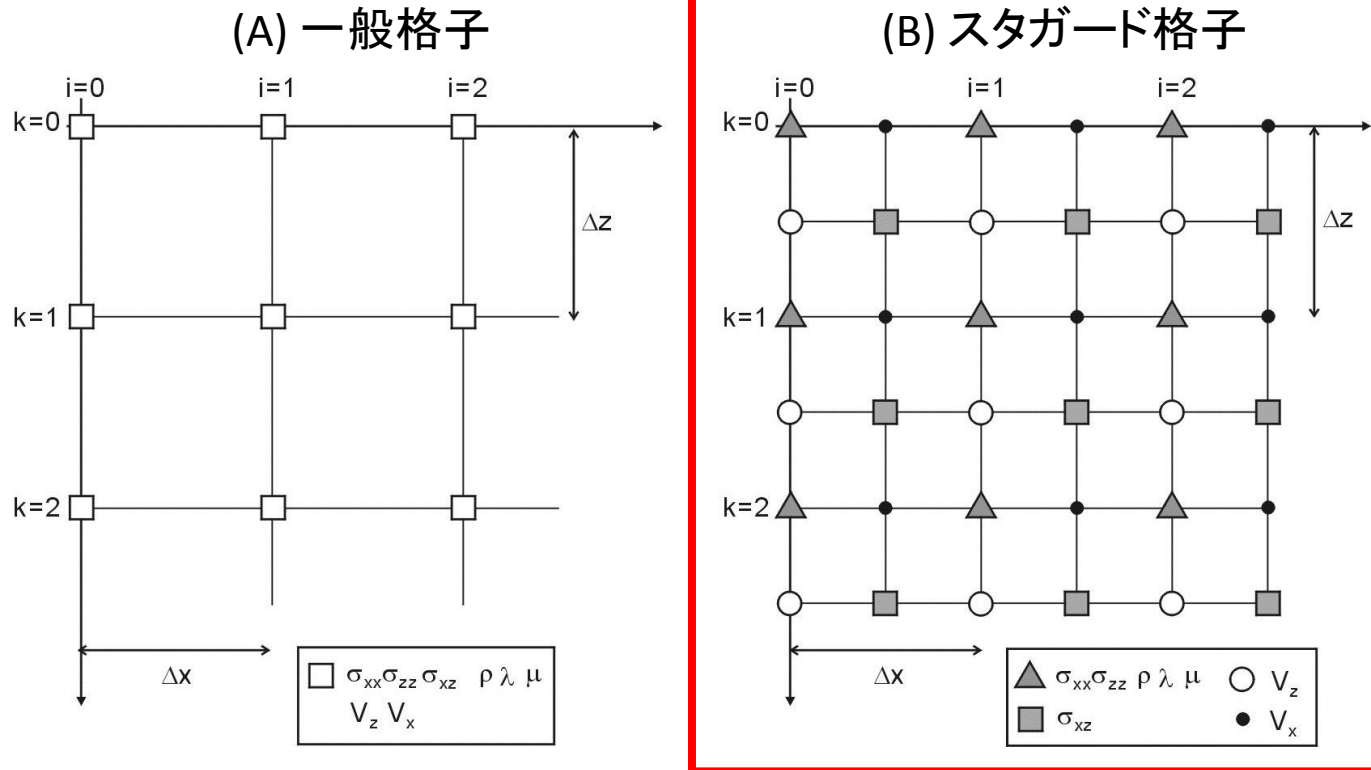
$$\dot{u}_p^{n+\frac{1}{2}} = \dot{u}_p^{n-\frac{1}{2}} + \frac{1}{\rho} \left(\frac{\partial \sigma_{xp}^n}{\partial x} + \frac{\partial \sigma_{yp}^n}{\partial y} + \frac{\partial \sigma_{zp}^n}{\partial z} + f_p^n \right) \Delta t, \quad (p = x, y, z) \quad (4)$$

$\dot{u}^{n+1/2}_p$ 粒子速度

- 式(2)と式(3)を結合した式(5)を用いて中央差分に基づく時間積分

$$\sigma_{pq}^{n+1} = \sigma_{pq}^n + \left[\lambda \left(\frac{\partial \dot{u}_x^{n+\frac{1}{2}}}{\partial x} + \frac{\partial \dot{u}_y^{n+\frac{1}{2}}}{\partial y} + \frac{\partial \dot{u}_z^{n+\frac{1}{2}}}{\partial z} \right) \delta_{pq} + \mu \left(\frac{\partial \dot{u}_p^{n+\frac{1}{2}}}{\partial q} + \frac{\partial \dot{u}_q^{n+\frac{1}{2}}}{\partial p} \right) \right] \Delta t, \quad (p, q) = (x, y, z) \quad (5)$$

格子モデル



- A) 変位や応力、物性値など全ての変数を同一格子点上に配置する一般格子
- B) 変位を半格子ずれた位置に定義するスタガード格子
- 変数の微分が定義される位置に関連の変数が位置するため計算精度が良い

- FDMによる式(4)と(5)の空間微分

- 中心差分による計算(2次精度、4次精度、8次精度)

(2次精度)
$$\frac{d}{dx} \sigma_{pq}(x, y, z) \simeq \frac{1}{\Delta x} \left[\sigma_{pq} \left(x + \frac{\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{\Delta x}{2}, y, z \right) \right]$$

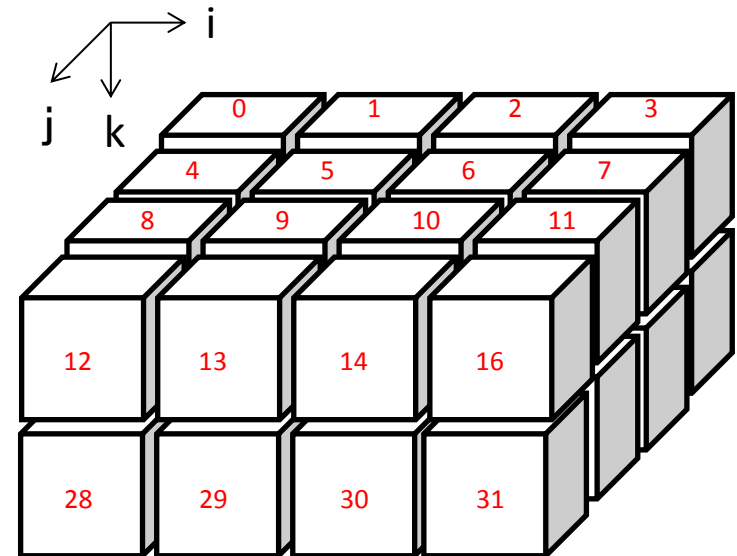
(4次精度)
$$\begin{aligned} \frac{d}{dx} \sigma_{pq}(x, y, z) \simeq \frac{1}{\Delta x} & \left[\frac{9}{8} \left\{ \sigma_{pq} \left(x + \frac{\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{\Delta x}{2}, y, z \right) \right\} \right. \\ & \left. - \frac{1}{24} \left\{ \sigma_{pq} \left(x + \frac{3\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{3\Delta x}{2}, y, z \right) \right\} \right] \end{aligned}$$

(8次精度)
$$\begin{aligned} \frac{d}{dx} \sigma_{pq}(x, y, z) \simeq \frac{1}{\Delta x} & \left[\frac{1225}{1024} \left\{ \sigma_{pq} \left(x + \frac{\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{\Delta x}{2}, y, z \right) \right\} \right. \\ & - \frac{245}{3072} \left\{ \sigma_{pq} \left(x + \frac{3\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{3\Delta x}{2}, y, z \right) \right\} \\ & + \frac{49}{5120} \left\{ \sigma_{pq} \left(x + \frac{5\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{5\Delta x}{2}, y, z \right) \right\} \\ & \left. - \frac{5}{7168} \left\{ \sigma_{pq} \left(x + \frac{7\Delta x}{2}, y, z \right) - \sigma_{pq} \left(x - \frac{7\Delta x}{2}, y, z \right) \right\} \right] \end{aligned}$$

ppOpen-APPL/FDM の概要

1. 弾性波動並列シミュレーション(地震)

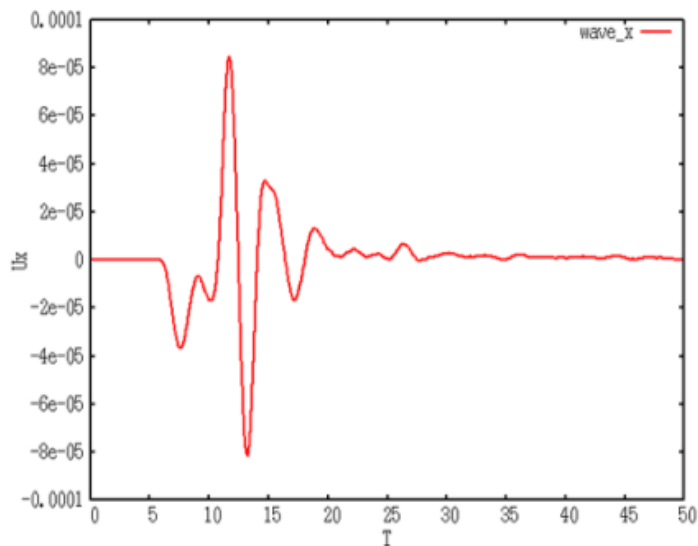
- Staggered グリッド、陽解法
- 3次元/2次元モデル
- 等間隔格子
- 微分: 2次、4次、8次精度
- MPI並列は3次元領域分割
- MPI/OpenMPハイブリッド並列



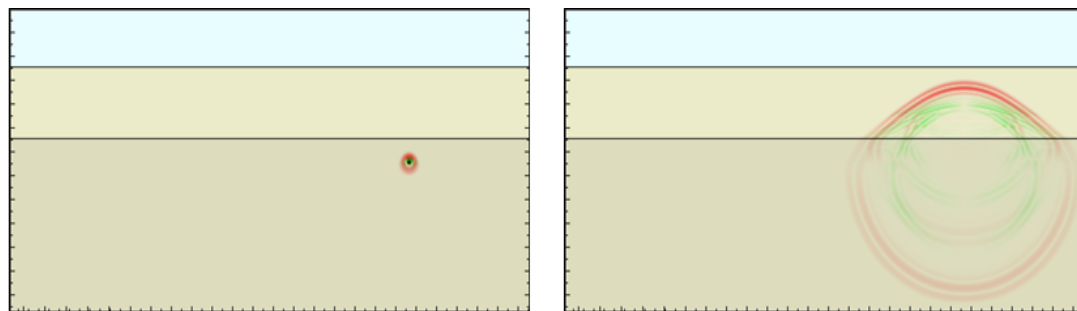
3次元領域分割
(赤文字: MPIランク)

2. サンプルプログラムとインターフェイス

- 観測点の波形
- 波動場のスナップショット

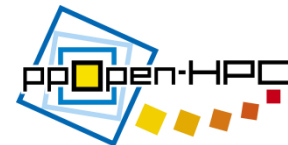


観測点の波形



波動場のスナップショット

ppOpen-APPL/FDM概要 (cont.)



ドキュメント

- ppOpen-APPL/FDMの使い方: ユーザマニュアル
- コード内のモジュール説明

Contents:	
1. Outline of seismic_2D/seismic_3D	5
1.1 Parallel simulation of seismic wave propagation in heterogeneous elastic media using ppOpenFDM	5
1.2 Grid and coordinate system	5
1.3 Equations of Motion for 3D Seismic Wavefields	5
1.4 Boundary conditions	6
1.4.1 Absorbing boundary	6
1.4.2 Free surface boundary	7
1.5 Anelastic attenuations	7
1.6 Spatial Differentiation	8
1.7 Input parameters	8
1.8 Requirements of time integration (CFL Condition)	9
1.9 Output and visualization of seismic waves	9
2. Parallel FDM simulation and performance	10
2.1 Domain partitioning and MPI	10
2.2 Parallel programming structure	11
2.3 Performance of parallel FDM simulation	11
2.4 Flat MPI model vs. thread MPI Hybrid model	12
References	13
3. Module/subroutine reference	14
3.1 seismic_2d_psv	15
3.1.1 module ppohFDM_m_absorb	17
3.1.2 module ppohFDM_m_convair	18
3.1.3 module ppohFDM_m_kernel	18
3.1.4 module ppohFDM_m_modnum	22
3.1.5 module ppohFDM_m_output	23
3.1.6 module ppohFDM_m_params	24
3.1.7 module ppohFDM_m_report	25
3.1.8 module ppohFDM_m_source	27
3.1.9 module ppohFDM_m_stlib	30
3.1.10 module ppohFDM_m_surfc	38
3.1.11 module ppohFDM_m_match	37
3.2 seismic_3d	39
3.2.1 module ppohFDM_boundary	50

3.2.1 module ppohFDM_boundary.

Description

This module applying a zero-stress boundary condition on free surface. Zero stress value is applied to stress components (S_{pz} , $p=x,y,z$) and the results of spatial derivatives just above and below the free-surface boundary are recalculated by using an one-side differentiation scheme. This scheme can treat irregular boundary as well as a flat boundary.

Dependency.

use ppohFDM_stdio.
use ppohFDM_param.

subroutine ppohFDM_bc_zero_stress

(KFSZ, NIFS, NJFS, IFSX, IFSY, IFSZ, JFSX, JFSY, JFSZ)

Description

Applying zero stress value to stress components (S_{pz} , $p=x,y,z$) on free surface.

Arguments.

integer, intent(in) :: KFSZ(NXP0:NXP1,NYP0:NYP1) ! depth of the free surface -
integer, intent(in) :: NIFS, NJFS ! number of points in x and y directions to examine free surface conditions-
integer, intent(in) :: IFSX(NFSMAX), IFSY(NFSMAX), IFSZ(NFSMAX) -
! evaluation point of free surface condition in x,y,z
integer, intent(in) :: JFSX(NFSMAX), JFSY(NFSMAX), JFSZ(NFSMAX)

MITライセンス

- 公開されているコードはユーザが自由に手を加えることができる



入力

- 計算パラメータの設定
- ソース、観測点、層構造の設定



計算



出力&可視化

- 観測点における波形
- 波動場

1. 入力パラメータの例 (計算パラメータ)



1. 計算パラメータ (m_param.f90)

モデルサイズ: 128*128*128

格子サイズ: 0.5

時間間隔: 0.025

MPI領域分割: 2*2*2

```
!-- << Model Size and Grid Width >>
```

```
integer, parameter :: NX = 128
```

```
integer, parameter :: NY = 128
```

```
integer, parameter :: NZ = 128
```

```
integer, parameter :: KFS = 25
```

```
integer, parameter :: NX1 = NX+1
```

```
integer, parameter :: NY1 = NY+1
```

```
integer, parameter :: NZ1 = NZ+1
```

```
integer, parameter :: NTMAX = 2000
```

```
integer, parameter :: NWRITE = 10
```

```
real(PN), parameter :: DX = 0.5_PN
```

```
real(PN), parameter :: DY = 0.5_PN
```

```
real(PN), parameter :: DZ = 0.5_PN
```

```
real(PN), parameter :: DT = 0.025_PN
```

```
integer, parameter :: NDUMP = 5
```

モデルサイズ

格子間隔

時間間隔

```
!--<< Parallel >>
```

```
integer, parameter :: IP = 2
```

```
integer, parameter :: JP = 2
```

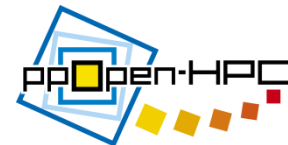
```
integer, parameter :: KP = 2
```

```
integer, parameter :: NP = IP*JP*KP ! Number of process
```

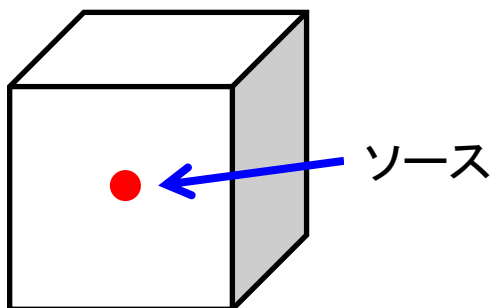
```
integer, parameter :: NL = 4 ! Order of the fd scheme
```

MPI領域分割

1. 入力パラメータの例 (ソースの設定、地下構造の設定)



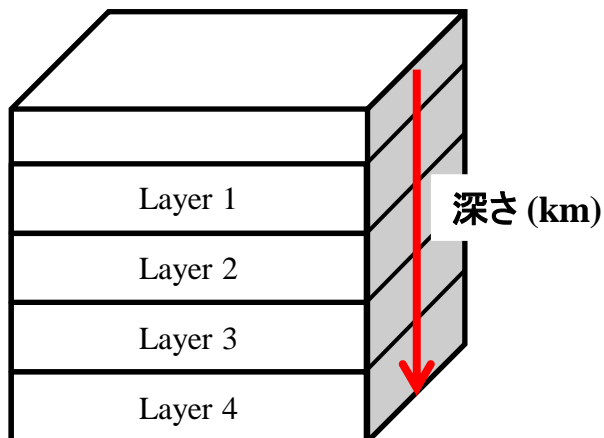
ソースの設定 (source.dat)



source.dat

```
16.0 16.0 16.0 # ソースの位置: X(km), Y(km), Z(km)
0.0 45.0 90.0 # 断層パラメータ
0.5 1.0 # ソース時間: at, t0 (s)
```

地下構造の設定 medium.dat



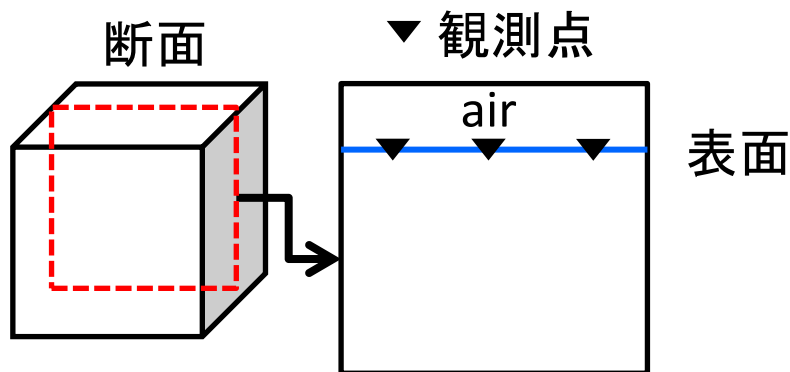
medium.dat

```
4 # 地下構造の層の数
20 2.3 3.0 1.7 # 深さ(km), 密度 (t/m3),
P波速度(km/s), S波速度 (km/s)
30 2.3 3.3 2.3
40 2.7 5.0 3.3
50 2.7 6.0 4.0
```

1. 入力パラメータの例 (観測点の設定)



観測点の設定
(station.dat)



station.dat

```

4          # ステーション数
10.0 10.0 0.0  # ステーション1: X, Y, Z(km)
                の設定
40.0 10.0 0.0
10.0 40.0 0.0
40.0 40.0 0.0
  
```

(注意) ./examples/seismic_3D-exampleにパラメータファイルが用意されています。

./src/seismic_3D/1.pureMPI-ppohVIS or 2.pure-MPIにCOPY

最大周波数と安定条件

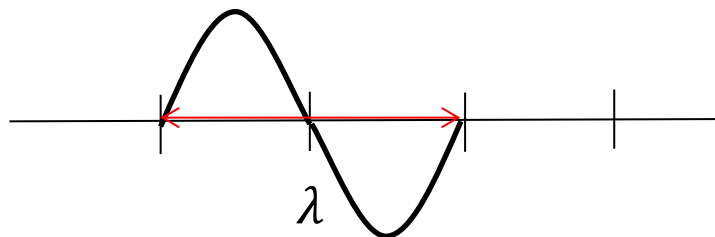


(1) 最大周波数の決め方

$$\frac{\lambda^{min}}{\Delta x} = 6$$

4次精度の場合6-8個
(経験的に)

$$\lambda^{min} = 6\Delta x = 6 \times 0.5 = \mathbf{3.0}$$



$$V_s^{min} = f^{max} \times \lambda^{min}$$

(伝わる速度) (周波数) (波長)

$$1.7 = f^{max} \times \mathbf{3.0}$$

$$f^{max} = \mathbf{0.6(Hz)}$$

(2) Δtの決め方

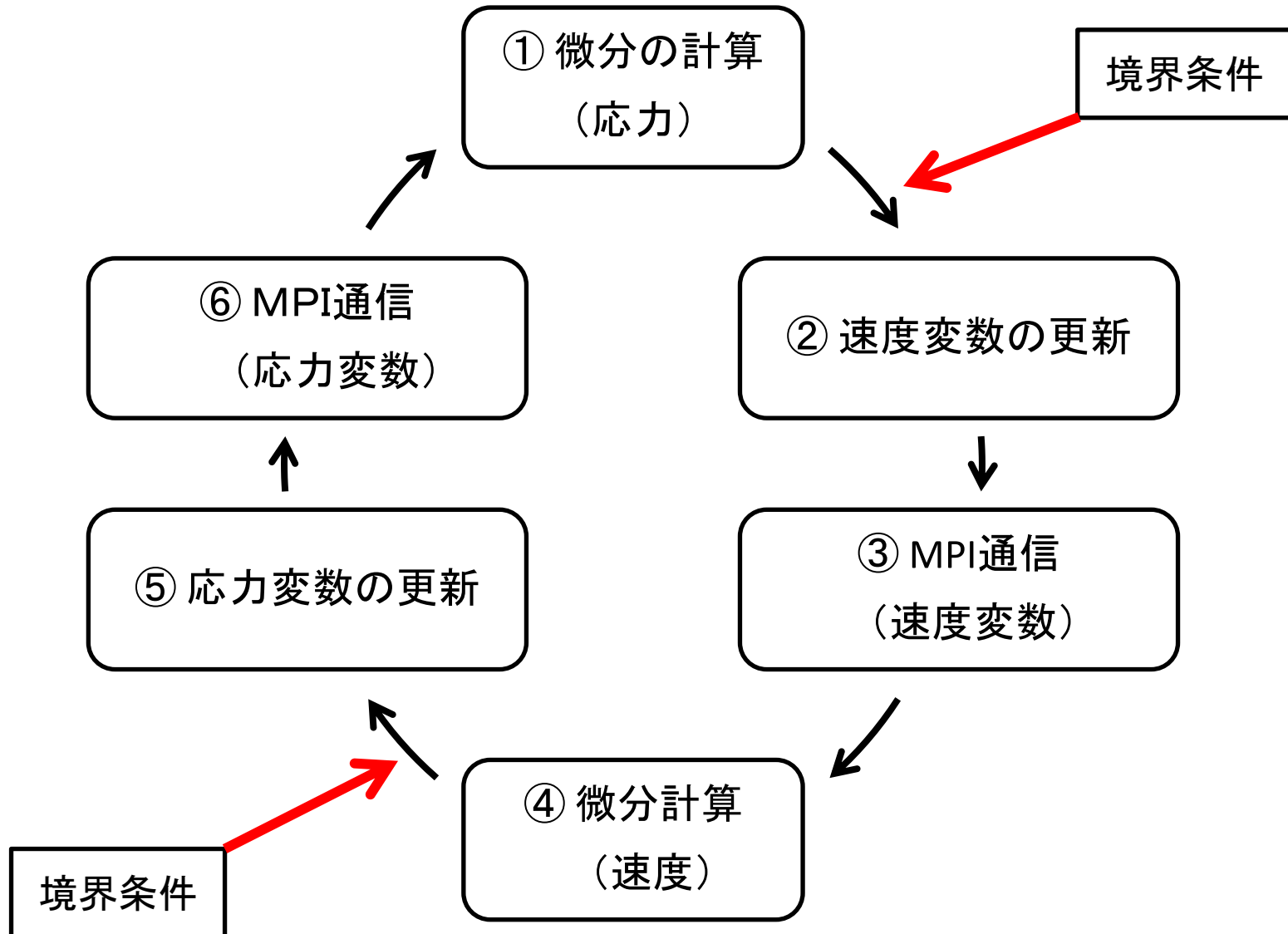
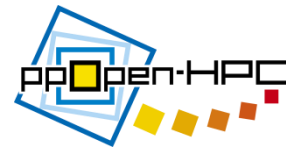
$$\Delta t < 0.2 \frac{\Delta x}{V_{max}}$$

$$\Delta t < 0.2 \frac{0.5}{4.0} = \mathbf{0.025}$$

青: 未知数

黒: 既知数

2. 計算手順



3. 出力および可視化

- 出力データ

- 各MPIランクで出力されている

```

SEISM3D3.prm          SEISM3D3.SUR.000.001.000  SEISM3D3.XY.001.001.000
SEISM3D3.SPS.000.000.000  SEISM3D3.SUR.000.001.001  SEISM3D3.XZ.000.000.000
SEISM3D3.SPS.000.000.001  SEISM3D3.SUR.001.000.000  SEISM3D3.XZ.000.000.001
SEISM3D3.SPS.000.001.000  SEISM3D3.SUR.001.000.001  SEISM3D3.XZ.001.000.000
SEISM3D3.SPS.000.001.001  SEISM3D3.SUR.001.001.000  SEISM3D3.XZ.001.000.001
SEISM3D3.SPS.001.000.000  SEISM3D3.SUR.001.001.001  SEISM3D3.YZ.000.000.000
SEISM3D3.SPS.001.000.001  SEISM3D3.WAV.000.000.000  SEISM3D3.YZ.000.000.001
SEISM3D3.SPS.001.001.000  SEISM3D3.WAV.000.000.001  SEISM3D3.YZ.000.001.000
SEISM3D3.SPS.001.001.001  SEISM3D3.XY.000.000.000  SEISM3D3.YZ.000.001.001
SEISM3D3.SUR.000.000.000  SEISM3D3.XY.000.001.000
SEISM3D3.SUR.000.000.001  SEISM3D3.XY.001.000.000
  
```

SEISM3D3.prm

SEISM3D3.WAV.***

SEISM3D3.SPS.***

SEISM3D3.SUR.***

SEISM3D3.XY(XZ, YZ).***

計算パラメータ

観測点における波形

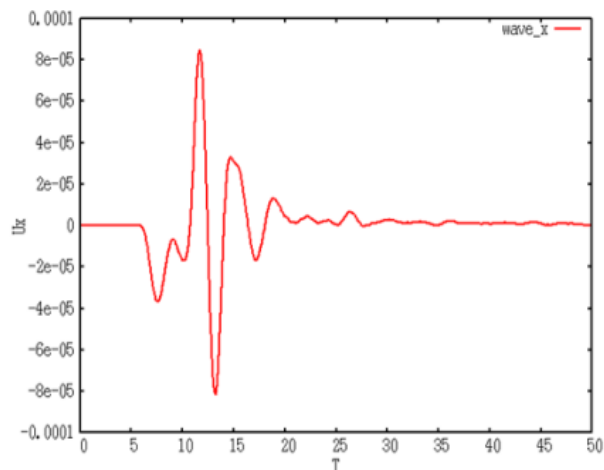
P波とS波の波動場

表面上での波動場

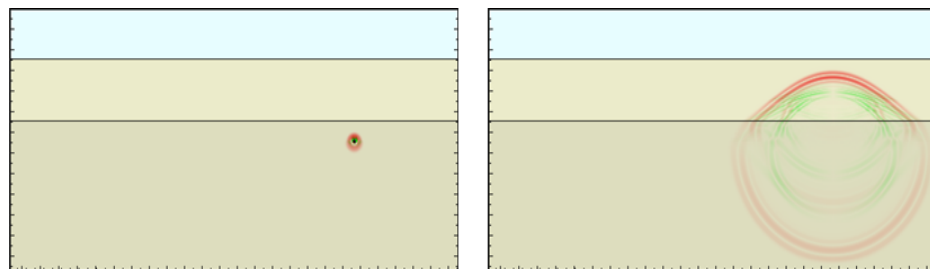
各断面での波動場

3. 出力および可視化 (cont.)

- `./tools/seismic_3D-tools`においてmakeすると4つ実行ファイル(`catsnap`, `catwav`, `ppmxy3d3`, `rwav3d`)が生成される
- `% catsnap SEISM3D3.prm` → 分割されたファイルが結合される(波動場)
- `% catwav SEISM3D3.prm` → 分割されたファイル が結合される(波形)
- `% ppmxy3d3` → 波動場のスナップショット (`xv`や`imagemagick`で可視化)
- `% rwav3d` → 観測点の波形 (gnuplotで可視化)



観測点の波形

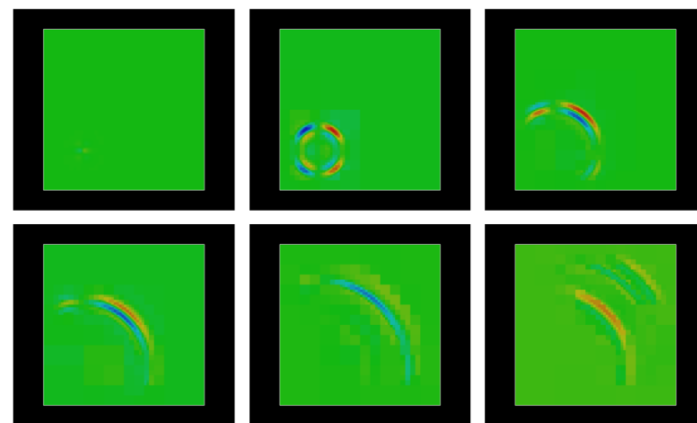
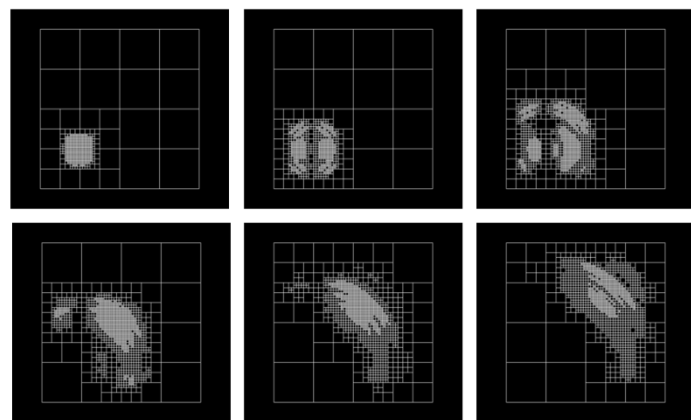


波動場のスナップショット
(P波: 赤、S波: 緑)

可視化 ppOpen-MATH/VIS



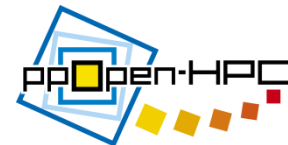
- プロジェクト内で公開している大規模データのための可視化ライブラリppOpen-MATH/VIS
- ppOpen-APPL/FDM ver0.2.0に実装済み
 - ./src/seismic_3D/1.pureMPI-ppohVIS
 - 出力ファイルは ./src/seismic_3D/1.pureMPI-ppohVIS
 - control.datのMaxVoxelCountやMaxRefineLevelの値を大きくすると細かくなる (./examples/seismic_3D-example)
 - AVSやParaviewで可視化することができる



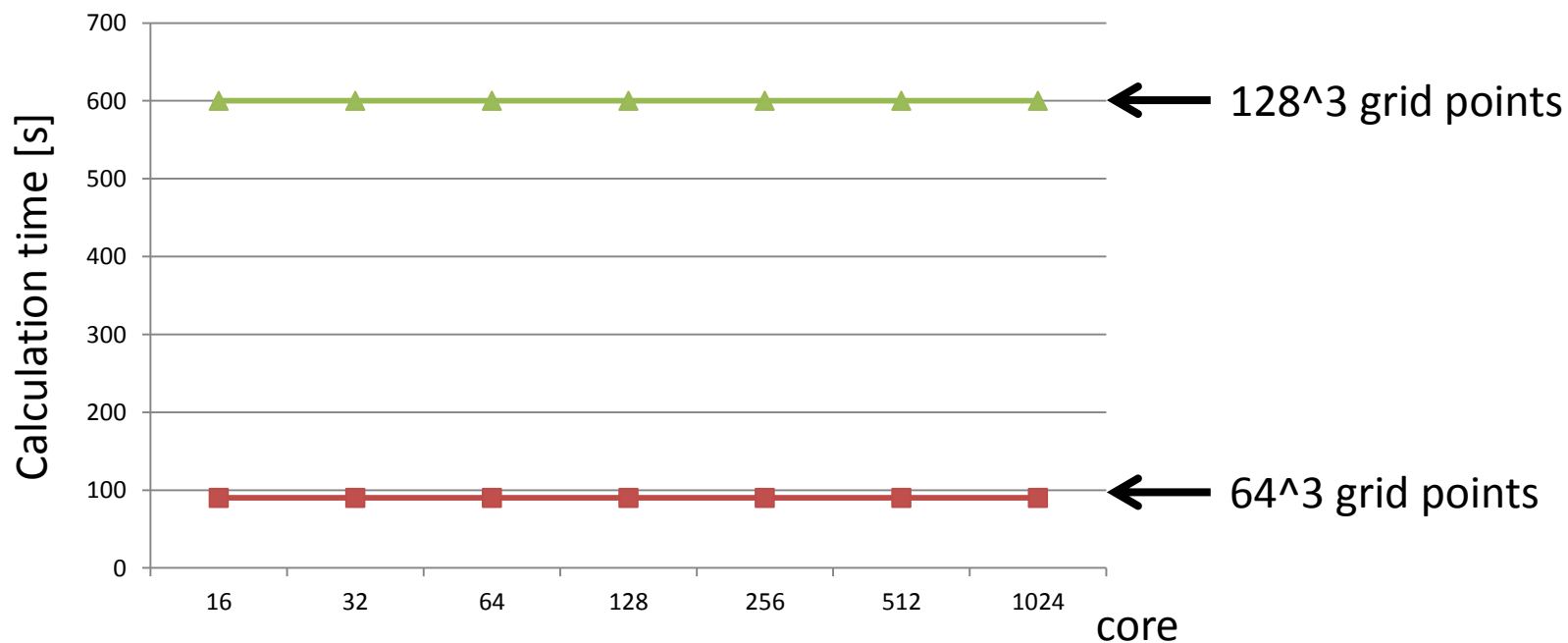
VXの速度場

ppOpen-APPL/FDM Performance in FX10

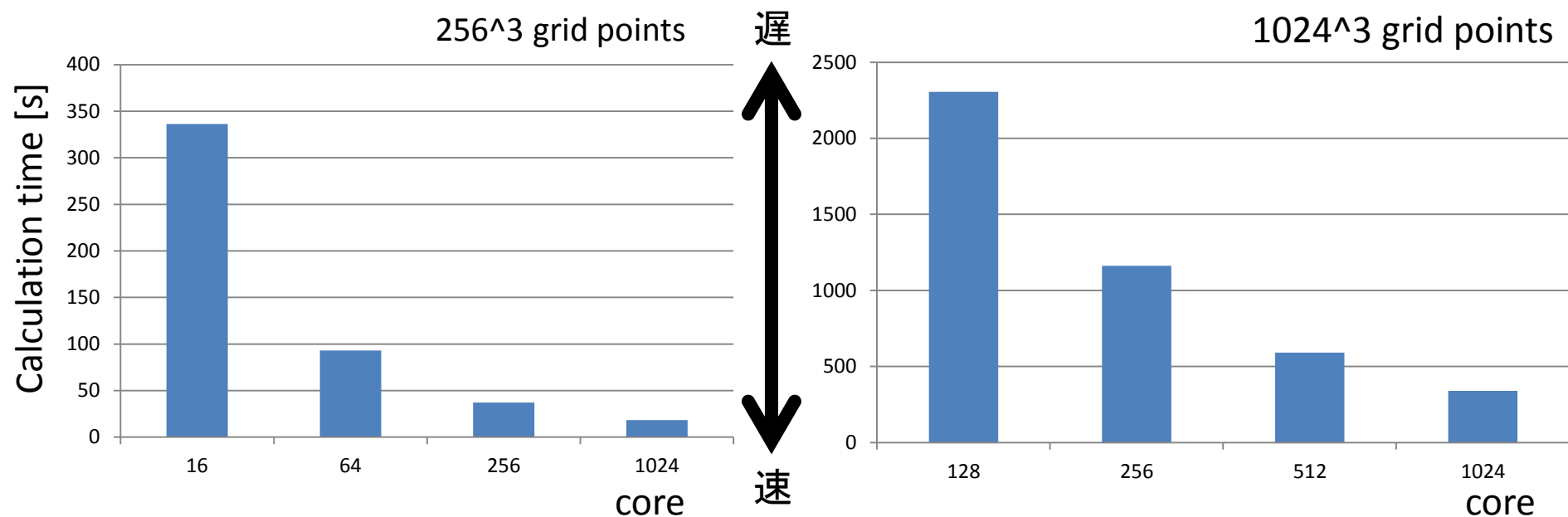
Weak scaling test in flat MPI



- モデルサイズ: 64^3 , 128^3 grid points
- 並列数: 16~1024 コア
- 3D domain partitioning of MPI



Strong scaling test in the flat MPI

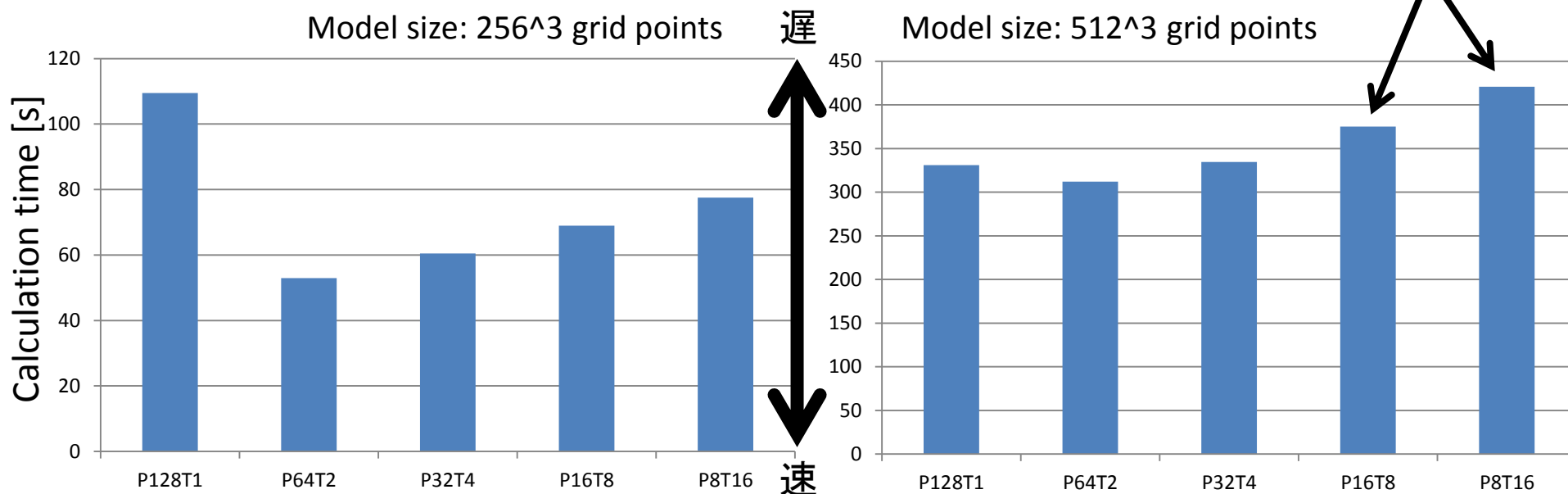


- 計算時間はコア数が増加することで低減している
 - 256³ grid points
 - 1024コアでの計算時間は、16コアの計算時間よりも18倍速くなっていた
 - 1024³ grid points
 - 1024コアでの計算時間は、128コアの計算時間よりも7倍速くなっていた

Storing scaling test in the MPI/OpenMP hybrid parallel computing on FX10



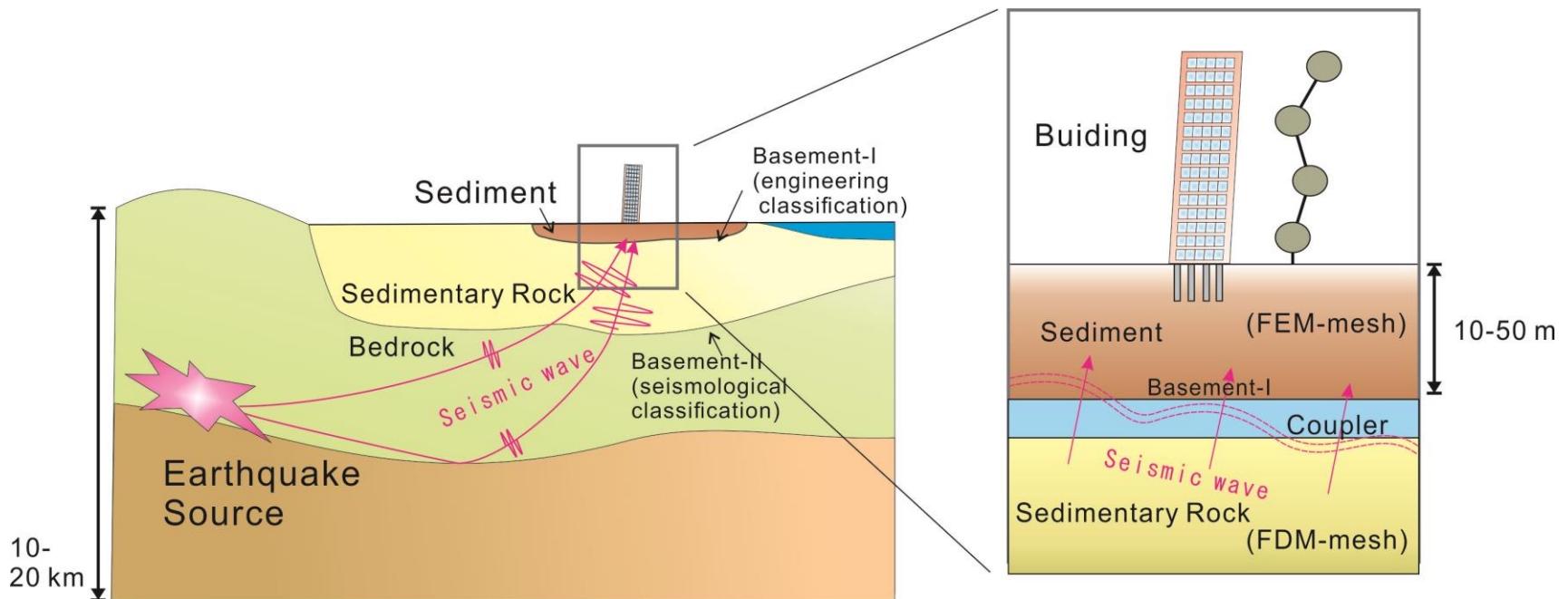
- パフォーマンステストは8ノード(128コア, 16コア/ノード)を使って評価
 - P128T1, P64T2, P32T4, P16T8, P8T16 (Pはプロセス、Tはスレッド)
- モデルサイズ
 - 256³, 512³グリッドポイント
- すべてのモデルにおいて、最小の計算時間は、P64T2 による並列のときであった
 - Pure MPIよりも **P64T2** の方が2倍高速化していた



ppOpen-APPL/FDMの実例
大規模連成計算
(地震動)

大規模連成計算の取り組み

- ppOpen-HPCプロジェクト内で開発されているppOpen-MATH/MP couplerを用いて差分法(地震動→ ppOpen-APPL/FDM)と有限要素法(構造解析→ ppOpen-APPL/FEM)を連成し、大規模連成計算をおこなっている



FDM: Seismic Wave Propagation → ppOpen-MATH/MP → FEM: Building Response

地震動計算



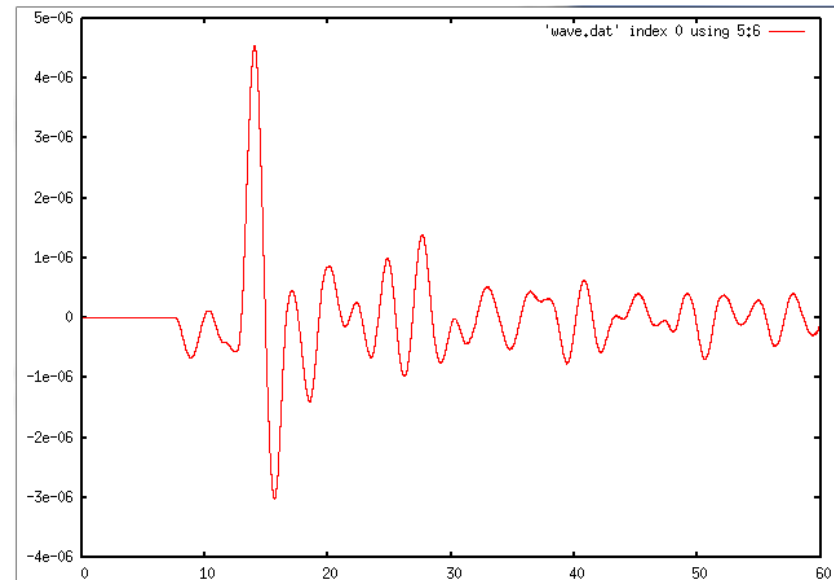
淡路を震源とした地震を想定し、ポートアイランドにある計算科学研究機構 京コンピュータの建屋への影響をシミュレーションをおこなう

計算領域(黒点線): $60\text{km} \times 60\text{km} \times 30\text{km}$
 計算格子間隔: $\Delta x = \Delta y = 40\text{ m}, \Delta z = 20\text{ m}$
 計算ステップ: 6万ステップ ($\Delta t = 0.001, 60\text{s}$)
 計算グリッド: $1536 \times 1536 \times 1600$

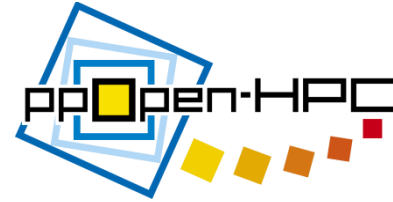
Google mapから引用



計算科学研究機構 (AICS)
京コンピュータ建屋



X方向の変位



ppOpen-APPL/FDM 利用方法と演習

はじめに

/home/c31003/shareに20150327-lectureディレクトリがありますのでコピーしてください

* 以下のファイルがインストールされていることを必須としています

Putty (in windows)

WinSCP (in windows)

FUJITSU Software Development

Paraview

ディレクトリとファイルを確認

./ppohFDM_0.2.0以下を確認 (以下このディレクトリをディレクトリとして説明)

演習で使うコードは、./ppohFDM_0.2.0/src 以下にあります

```
[c31003@oakleaf-fx-6 ppohFDM_0.2.0]$ ls
doc                LICENSE_ppohMATH-VIS  ppohMATH-VIS-install
etc                Makefile               ppohMATH-VIS-lib
examples          Makefile~              README_ppoh-APPL-FDM
INSTALL_ppohAPPL-FDM  Makefile.in           src
LICENSE_ppohAPPL-FDM  Makefile.in~          tools
[c31003@oakleaf-fx-6 ppohFDM_0.2.0]$ cd src/
[c31003@oakleaf-fx-6 src]$ ls
seismic_2D  seismic_3D
[c31003@oakleaf-fx-6 src]$ cd seismic_3D/
[c31003@oakleaf-fx-6 seismic_3D]$ ls
1. ppohFDM-ppohVIS  2. parallel
```

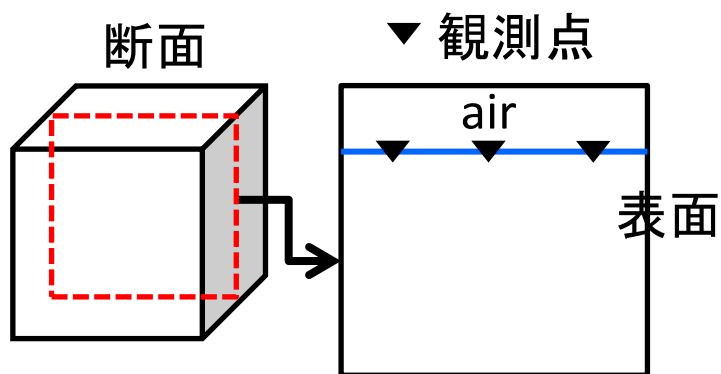
ppohVISライブラリが実装

並列コード

入力パラメータ設定(1)

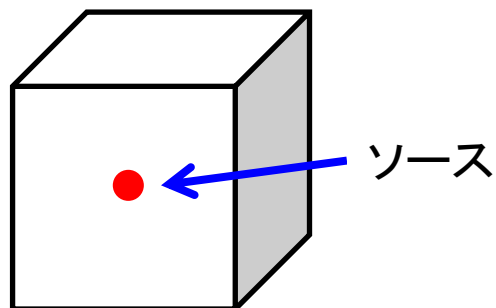
• 3つの入力パラメータファイル

1. 観測点 (X, Y, Z (km))の設定



station.dat	
4	# ステーション数
10.0 10.0 0.0	# ステーション1: X, Y, Z(km)を設定
40.0 10.0 0.0	
10.0 40.0 0.0	
40.0 40.0 0.0	

2. ソースパラメータを設定



source.dat	
16.0 16.0 16.0	# ソースの位置: X(km), Y(km), Z(km)
0.0 45.0 90.0	# 断層パラメータ
0.5 1.0	# ソース時間: at, t0 (s)

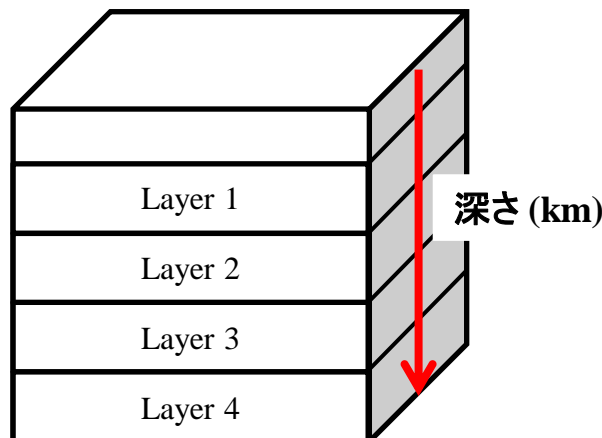
入力パラメータ設定(2)

3. 地下構造モデル(深さ(depth), 密度($R0$), P波の速度(VP), S波の速度(VS))を設定

- medium.dat

medium.dat

4	# 地下構造の層の数
20 2.3 3.0 1.7	# 深さ(km), 密度 (t/m ³), P波速度(km/s), S波速度 (km/s)
30 2.3 3.3 2.3	
40 2.7 5.0 3.3	
50 2.7 6.0 4.0	



計算パラメータ設定



計算パラメータ (m_param.f90)

(計算モデルとタイムステップ)

- モデルサイズ: NX, NY, NZ
- 格子間隔: DX, DY, DZ
- タイムステップ: NTMAX
- 時間間隔: DT

(MPI: 3次元分割)

- 分割: IP, JP, KP
- プロセス数: NP

```
!-- << Model Size and Grid Width >>
```

```
integer, parameter :: NX = 128
```

```
integer, parameter :: NY = 128
```

```
integer, parameter :: NZ = 128
```

```
integer, parameter :: KFS = 25
```

```
integer, parameter :: NX1 = NX+1
```

```
integer, parameter :: NY1 = NY+1
```

```
integer, parameter :: NZ1 = NZ+1
```

```
integer, parameter :: NTMAX = 2000
```

```
integer, parameter :: NWRITE = 10
```

```
real(PN), parameter :: DX = 0.5_PN
```

```
real(PN), parameter :: DY = 0.5_PN
```

```
real(PN), parameter :: DZ = 0.5_PN
```

```
real(PN), parameter :: DT = 0.025_PN
```

```
integer, parameter :: NDUMP = 5
```

モデルサイズ

タイムステップ

格子間隔

時間間隔

```
!--<< Parallel >>
```

```
integer, parameter :: IP = 2
```

```
integer, parameter :: JP = 2
```

```
integer, parameter :: KP = 2
```

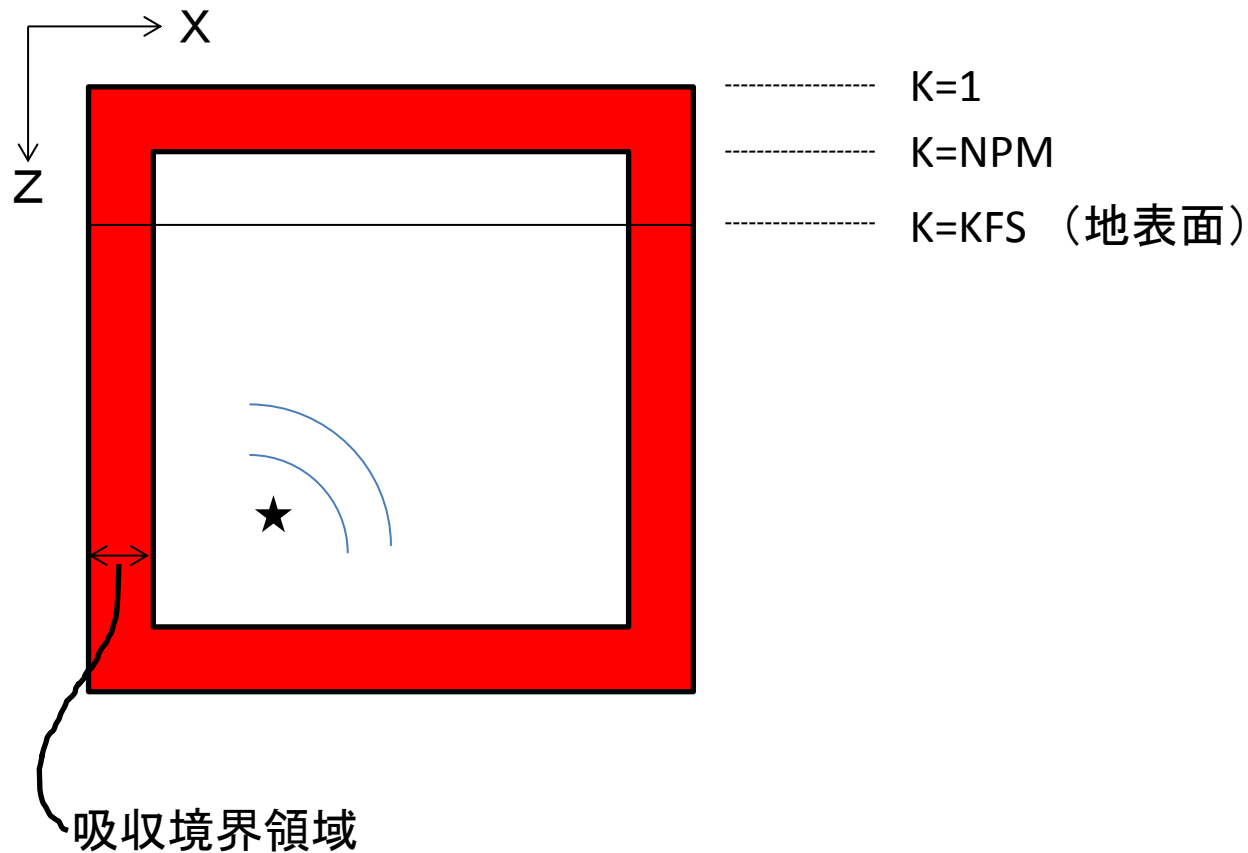
```
integer, parameter :: NP = IP*JP*KP ! Number of process
```

```
integer, parameter :: NL = 4 ! Order of the fd scheme
```

MPI領域分割

(注意) ./examples/seismic_3D-exampleにパラメータファイルが用意されています。

注意

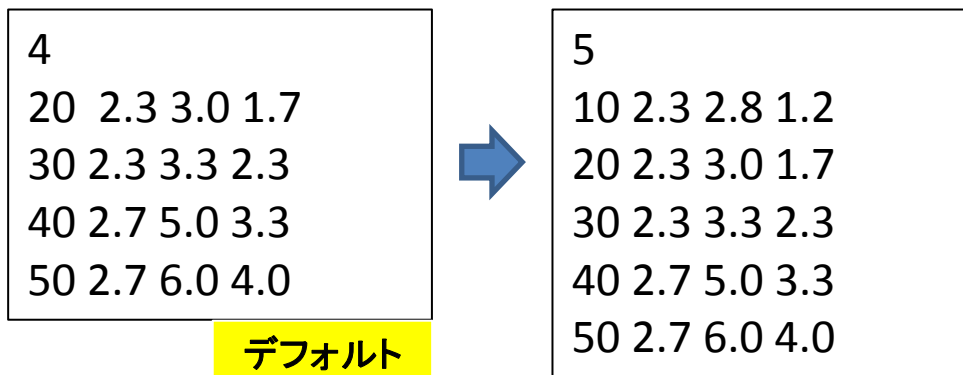


本コードでは、吸収境界条件を設定しています。吸収境界は`m_param.f90`内のNPM変数において、20 gridsに設定しています。また、地表面のリファレンスとしてKFS変数を定義しています。サンプルでは、KFSは5gridsとしています。

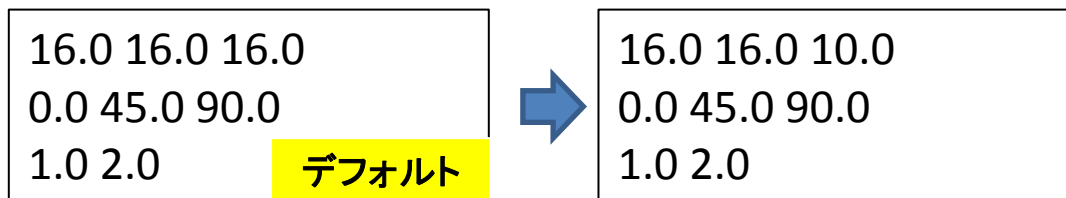
演習(1)

1. 媒質の物理値を変化させ、波動場の観測せよ。パラメータは以下の通り

- 計算パラメータの設定 (m_param.f90)
 - モデルサイズ(NX*NY*NZ): 128*128*128
- 媒質の設定 (medium.dat)



- ソースの設定 (source.dat)



演習(1)(cont.)

– 観測点の設定(station.dat)

```
4
10.0 10.0 0.0
40.0 10.0 0.0
10.0 40.0 0.0
40.0 40.0 0.0
```

– control.dat

```
[Refine]
AvailableMemory = 2.0
MaxVoxelCount = 10000
MaxRefineLevel = 20
[Simple]
ReductionRate = 0.0
```

- ラベルを設定する(seism3d3n.f90のL107)
- 出力間隔を設定する(seism3d3n.f90のL358)
- 出力パラメータの設定(seism3d3n.f90のL364)

- 上記のパラメータとデフォルトパラメータとの違いを確認せよ
 1. 媒質のパラメータを変更したときの違い
 2. ソースのパラメータを変更したときの違い
 3. control.datのMaxVoxelCountとMaxRefineLevelの値を変更したときの違い確認せよ

演習(1)(cont.)



• コンパイル

1. % ./make
2. % ./make install
3. % ./make seism3d-ppohVIS
4. % ./make install

(注意)

Makefile.inにMATH/VISライブラリのコンパイル先が書かれている。各ユーザはディレクトリを場所を絶対パスで書く

• 実行

- % cd ./src/seismic_3D/1.ppohFDM-ppohVIS
- %pjsub job

```
#!/bin/sh
```

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

```
#PJM -L "node=**"
```

```
#PJM -L "elapsed=**:*:*:"
```

```
#PJM -g **
```

```
#PJM --mpi "proc=**"
```

m_param.f90で設定した
NPとproc数を同じにする

```
mpiexec ./seism3d3n
```

**はユーザがパラメータに応じて変更

演習(1)



- ppOpen-MATH/VISで出力されたinpファイルをparaviewを使って可視化
 - ./src/seismic_3D/1.ppohFDM-ppohVIS/ppohVISにデータが出力されていることを確認
- Paraviewを使って可視化する

演習(2) pureMPI並列



1. 並列数とモデルサイズを変更して計算時間の変化を計測せよ

- m_param.f90
 - モデルサイズ: 256*256*256 grid points
 - NP: 8, 16, 32, 64, ...
- プロファイル情報を取得するために、計測した部分にcall fapp_start, call fapp_stopで挟む
 - seism3d3n.f90に実装済み

!! Velocity Update

```
call fapp_start("region2",1,1)
```

```
call ppohFDM_update_vel ( 1, NXP, 1, NYP, 1, NZP )
```

```
call fapp_stop("region2",1,1)
```

演習(2) pureMPI並列



• Job ファイル

```
#!/bin/sh

#PJM -L "rscgrp=**"
#PJM -L "node=**"
#PJM -L "elapsed=**:*:*:"
#PJM -g **
#PJM --mpi "proc=**"

fapp -C -d prof -L 1 -lhwm -Hevent=Statistics mpiexec lpgparm -p 256MB -d 256MB -h 256MB -s 256MB
-t 256MB ./seism3d3n
```

**はユーザがパラメータに応じて変更

- プロファイルを保存するディレクトリが必要
 - ./src/seismic_3D/2.parallel/prof

(注意)

計測にあたりseism3d3n.f90のL175, L316にあるI/Oに関連するサブルーチンをコメントアウトする

演習(3) Hybrid parallel computing

1. 並列数とモデルサイズを変更して計算時間の変化を計測せよ

- m_param.f90
 - モデルサイズ: 256*256*256 grid points
 - 使用するノード数: 8ノード(128コア) or 4ノード(64コア)に固定
 - IP, JP, KPの値を変化させる(プロセス数)
 - Job文のexport OMP_NUM_THREADSの値を変化させる
- Jobファイル

```
#!/bin/sh
```

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

```
#PJM -L "node=**"
```

```
#PJM -L "elapsed=**.*.*"
```

```
#PJM -g **
```

```
#PJM --mpi "proc=**"
```

```
export OMP_NUM_THREADS=**
```

```
fapp -C -d prof -L 1 -lhwm -Hevent=Statistics mpiexec lpgparm -p 256MB -d 256MB -h 256MB -s 256MB -t 256MB ./seism3d3n
```

**はユーザがパラメータに応じて変更

- 古村孝志, 地震波伝播と強震動の大規模並列FDMシミュレーション, 東京大学情報基盤センタースーパーコンピューティングニュース, Vol11, pp.35-63, 2009.
- ./doc以下にあるマニュアルを参照
 - ppOpen-APPL/FDM ver0.2.0 user guide
 - ppOpen-APPL/FDM ver0.2.0 reference guide