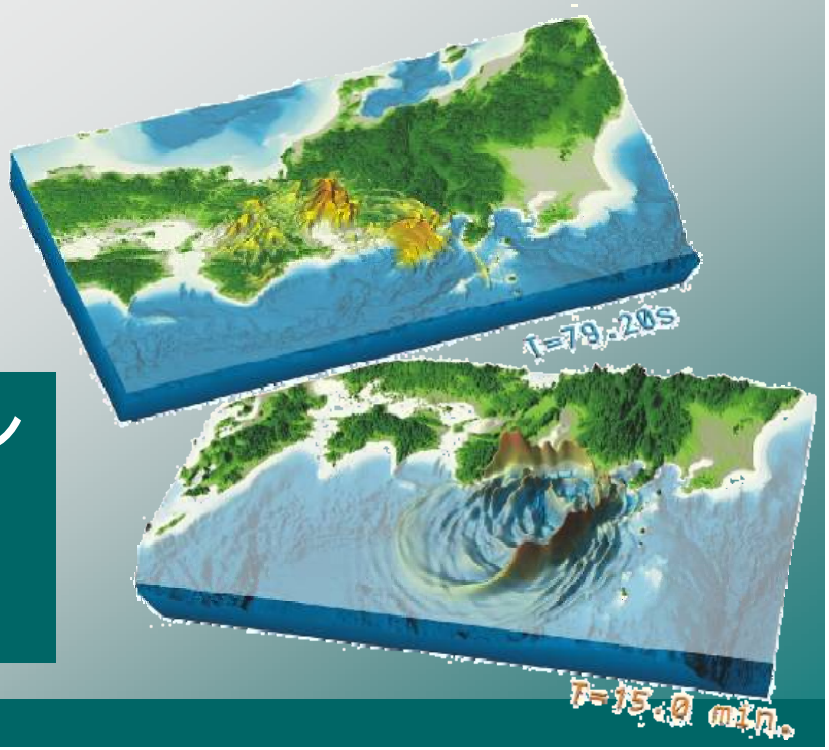
T2K（東大）共同研究プロジェクト利用報告会　2010

地震と津波発生伝播の大規模 3 次元シミュレーション

古村 孝志

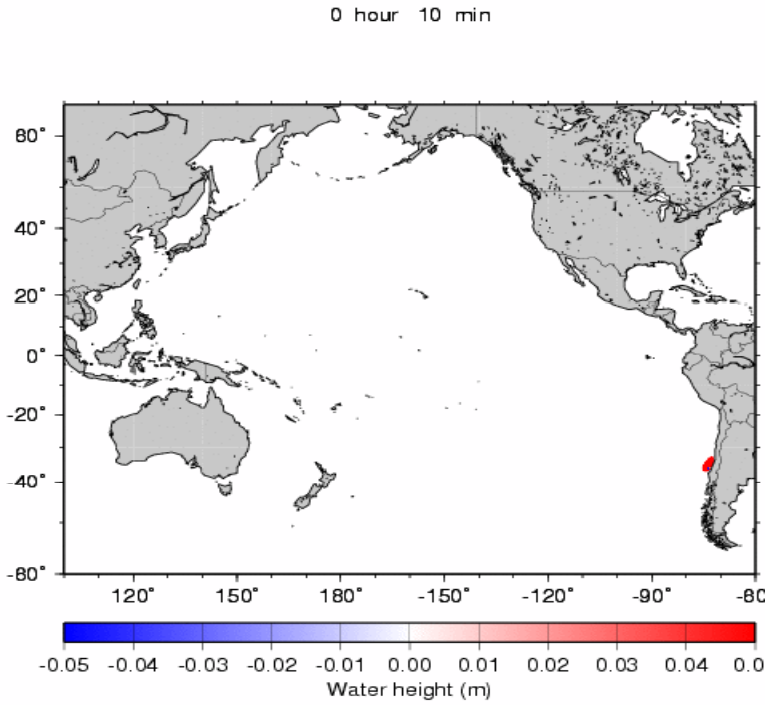
東京大学 情報学環 総合防災情報研究センター/地震研究所

① 南海トラフ地震・津波シミュレーション
② 地震と津波の高精度評価
③ 大規模並列計算に向けた課題

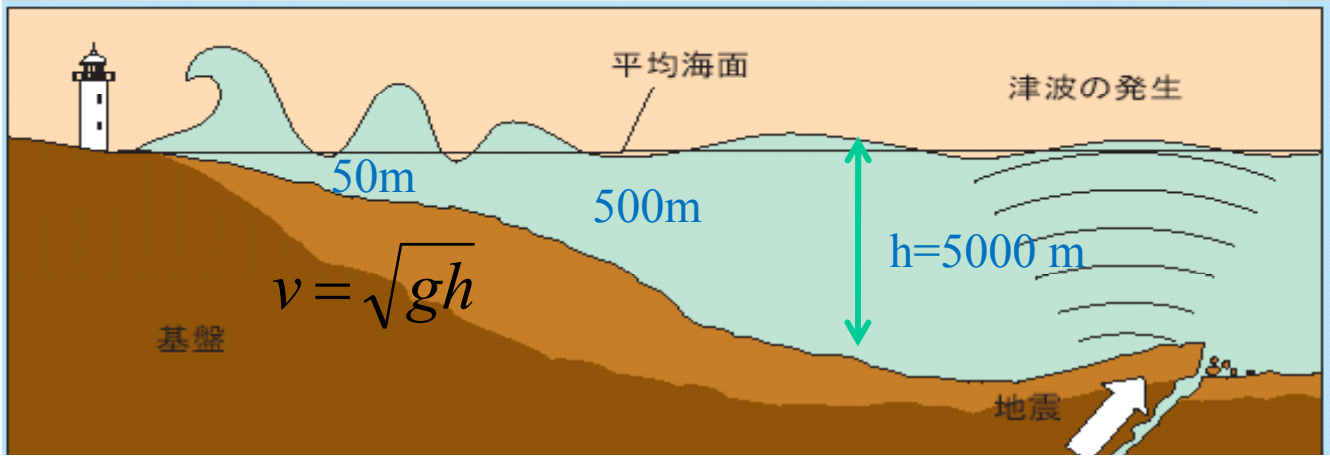


津波と被害

発生年	地震名	M	死者数
1896（明29）	明治三陸地震	8.5	21,259
1933（昭8）	昭和三陸地震	8.1	3,064
1944（昭19）	東南海地震	7.9	1,223
1946（昭21）	南海地震	8.0	1,443
1960（昭35）	チリ地震	9.5	142
1983（昭58）	日本海中部地震	7.7	104
1993（平5）	北海道南西沖	7.8	230

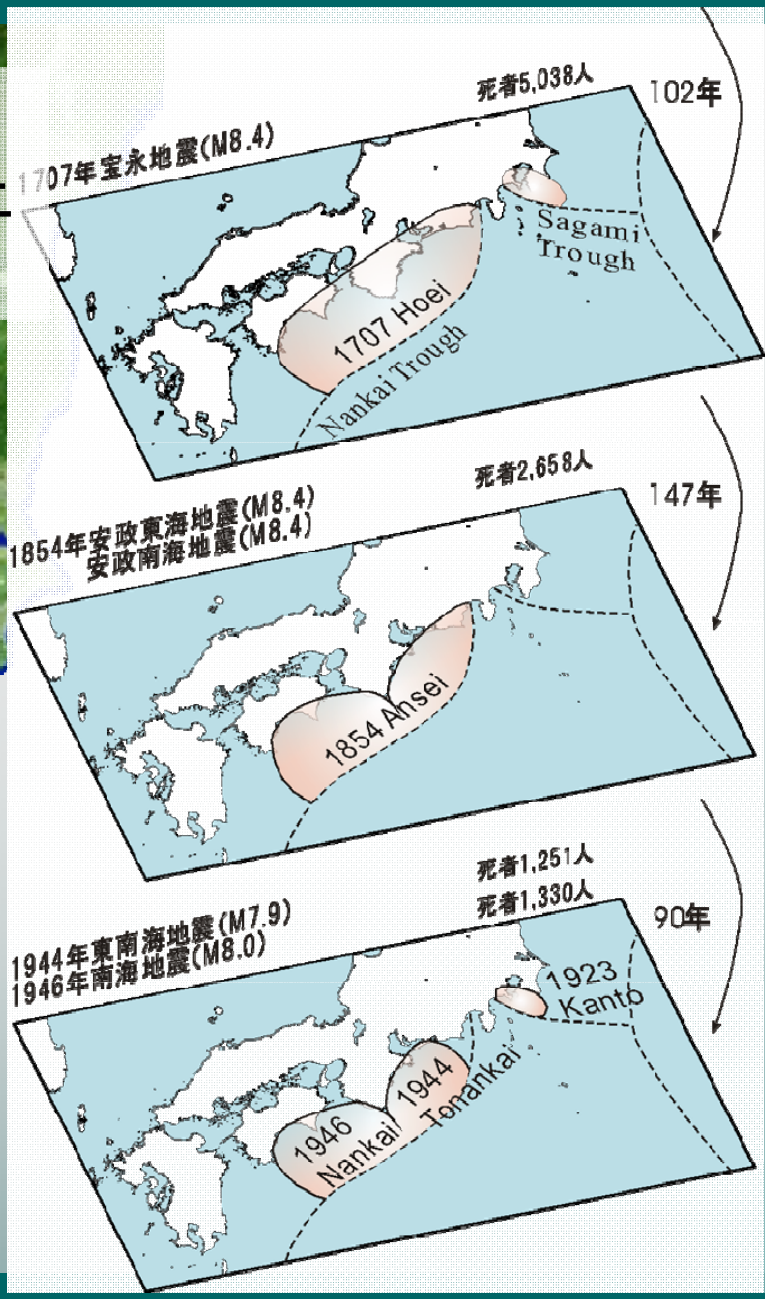


2010年チリ地震（M8.8)津波
東大地震研佐竹健治教授による



南海トラフ地震による強震動と津波

今後30年以内の地震発生確率
・南海・東南海・東海地震（M8.0）：　40-80%以上
・宮城沖地震（M7.5）：　99%以上



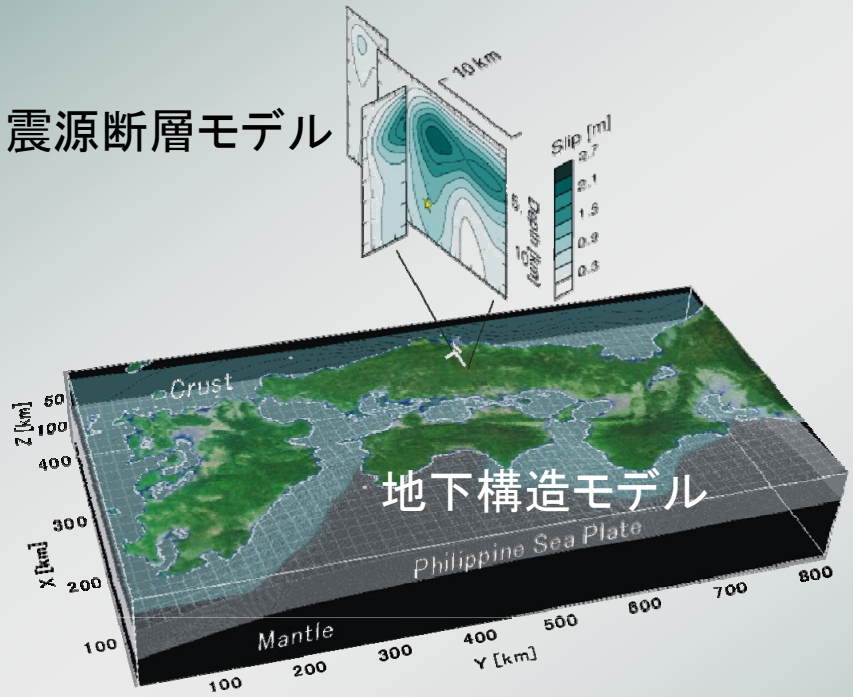
地震波伝播の大規模並列シミュレーション

1. 地下構造・震源断層モデル

2000年鳥取県西部地震

Yagi and Kikuchi (2001)

震源断層モデル

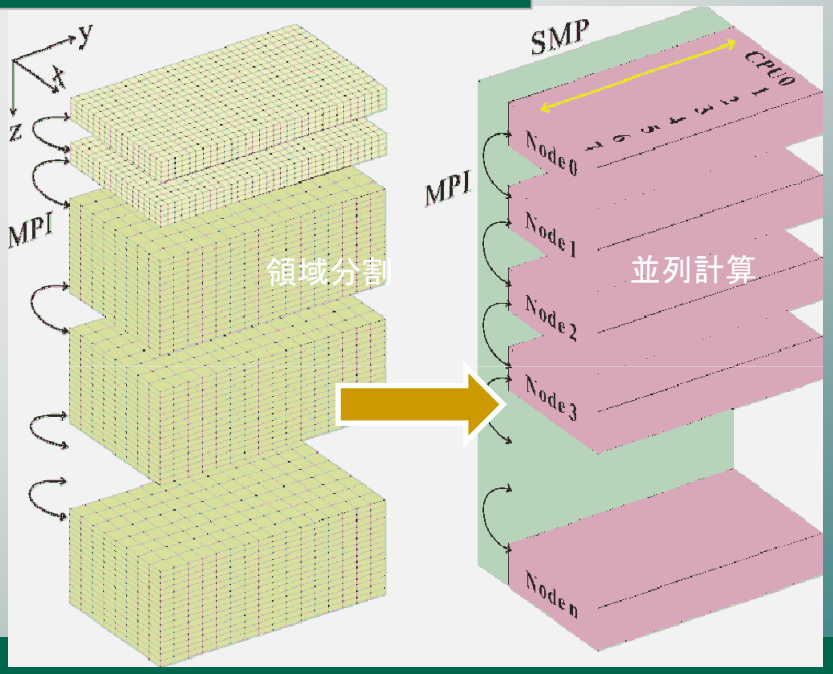


2. 運動方程式の差分計算

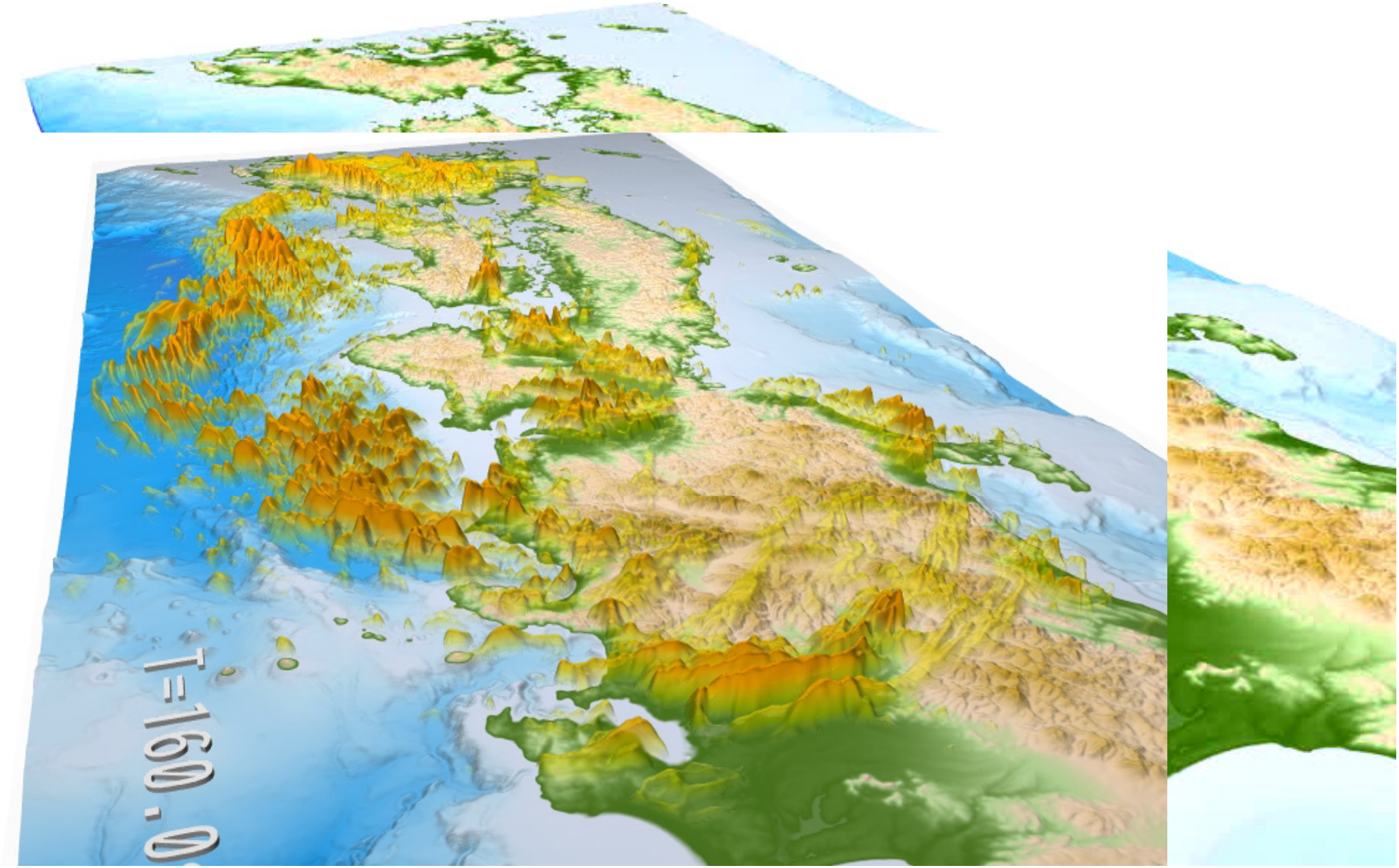
$$\rho \ddot{U}_p = \frac{\partial \sigma_{xp}}{\partial x} + \frac{\partial \sigma_{yp}}{\partial y} + \frac{\partial \sigma_{zp}}{\partial z} + f_p$$

$$\sigma_{pq} = \lambda \left(\frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} + \frac{\partial U_z}{\partial z} \right) \delta_{pq} + \mu \left(\frac{\partial U_p}{\partial q} + \frac{\partial U_q}{\partial p} \right)$$

3. 大規模 並列計算



1707年宝永地震による強震動(ES2による計算)



地震→津波　連成シミュレーション

運動方程式（釣り合いの式）

$$\rho \ddot{U}_p = \frac{\partial \sigma_{xp}}{\partial x} + \frac{\partial \sigma_{yp}}{\partial y} + \frac{\partial \sigma_{zp}}{\partial z} + f_p$$

応力ー歪み構成方程式 (Hooke's Law)

$$\sigma_{pq} = \lambda \left(\frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} + \frac{\partial U_z}{\partial z} \right) \delta_{pq} + \mu \left(\frac{\partial U_p}{\partial q} + \frac{\partial U_q}{\partial p} \right)$$

ナビエ･ストークス式（非圧縮性流体）

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{\partial p}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right] + g_x$$

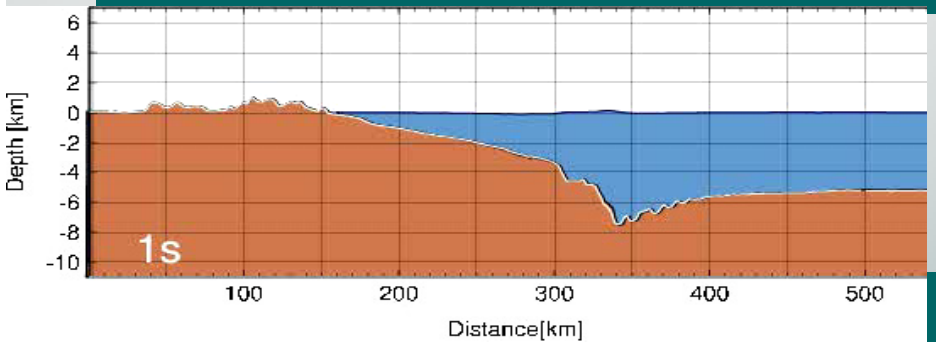
$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{\partial p}{\partial y} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right] + g_y$$

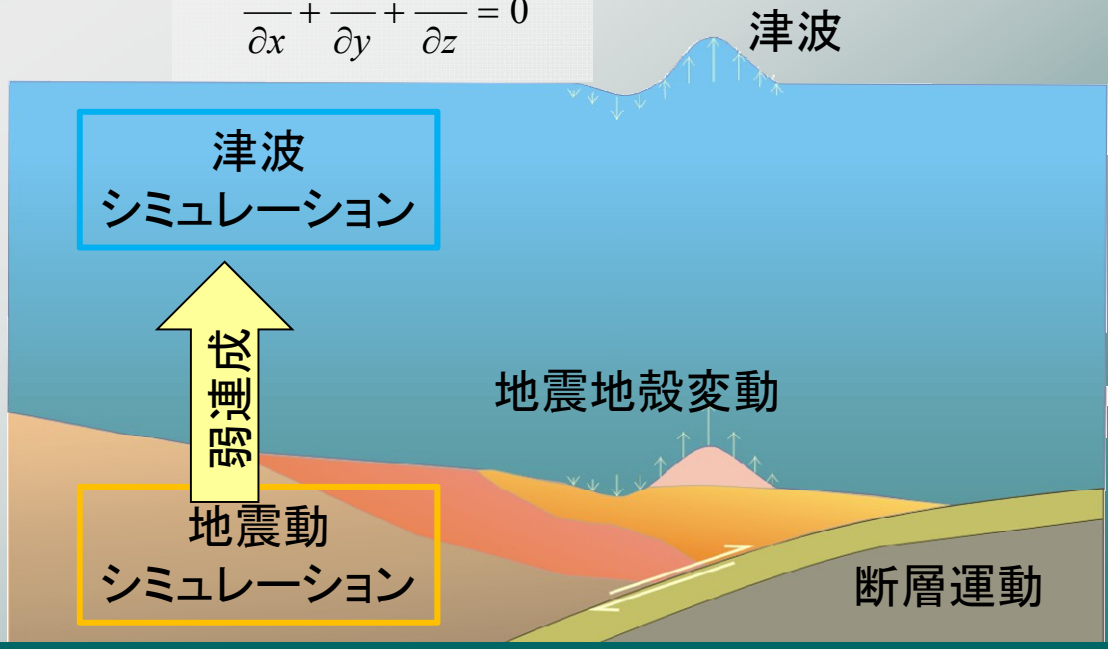
$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{\partial p}{\partial z} + \nu\left[\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right] + g_z$$

連続の式

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

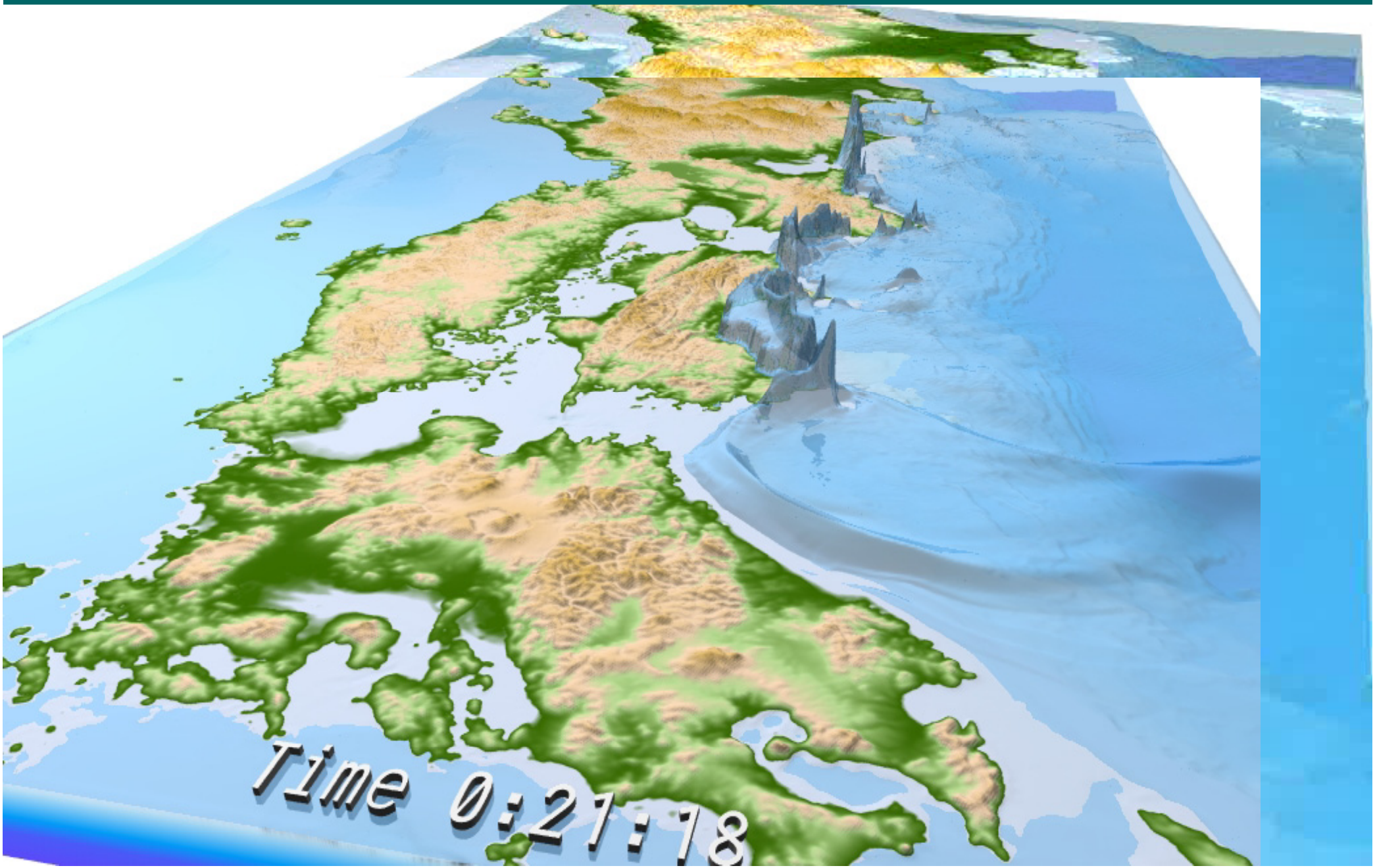
★地震ー津波シミュレーション
- 地震動計算により、地震地殻変動を評価
- 津波計算により、津波の発生と伝播を評価





Saito and Furumura (2009), Furumura and Saito (2009)

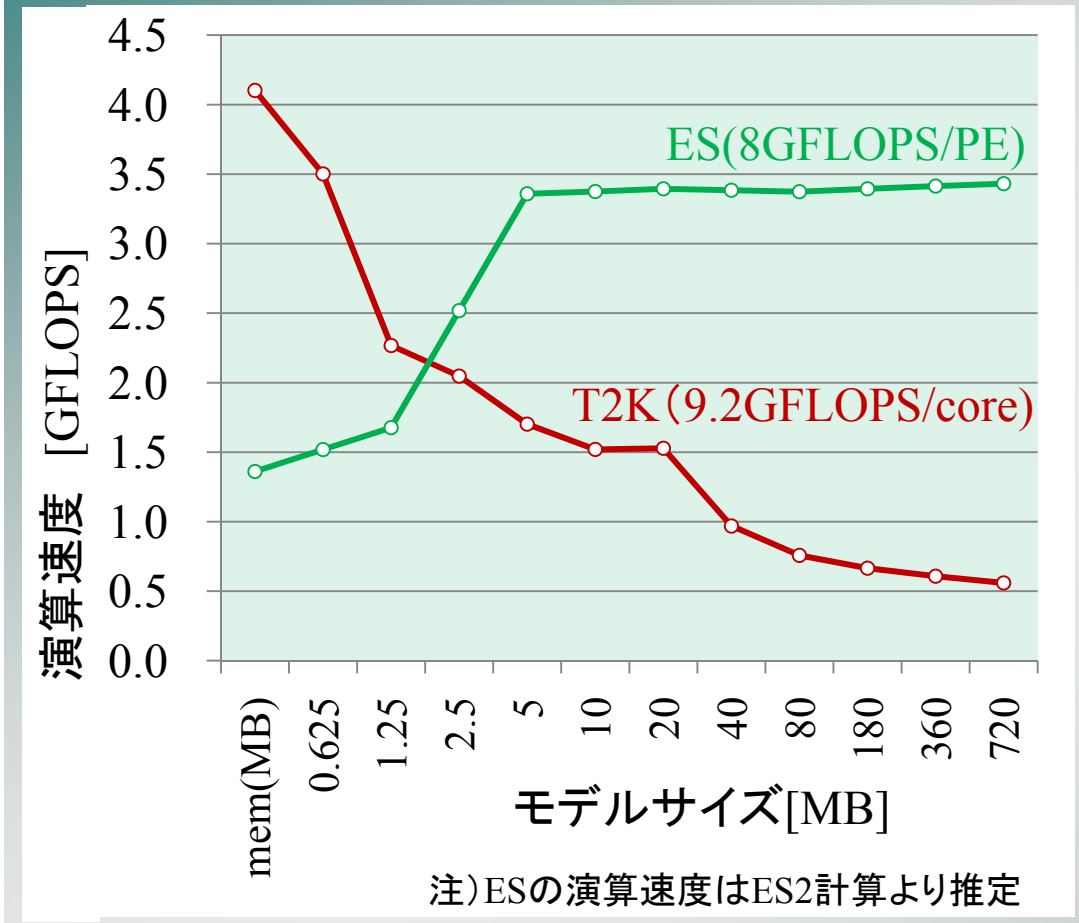
1707年宝永地震による津波（ES2による計算）

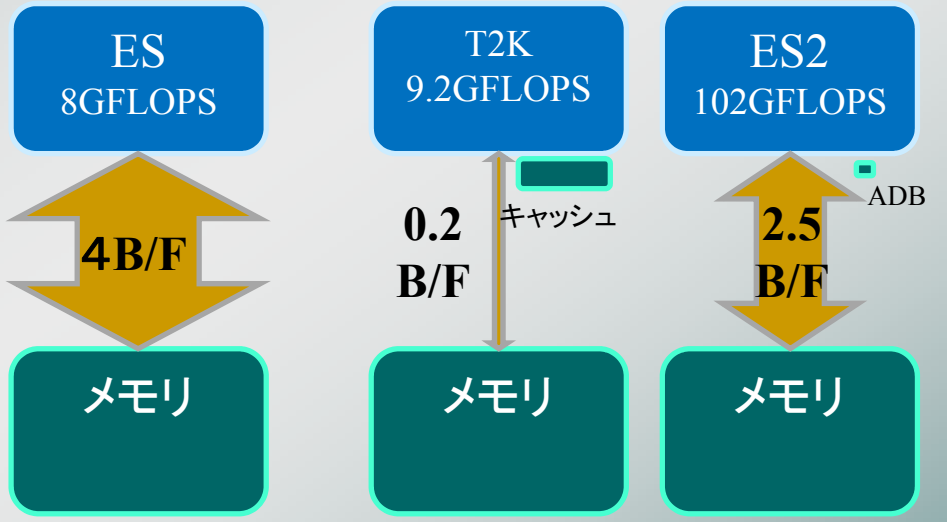


T2Kによる大規模並列計算

地震動計算の効率（ベクトルvsスカラーCPU）

演算速度





【実効性能】
- T2Kは、ESの1/6程度
- 6倍の並列化が必要

【T2K】小モデルはキャッシュ効果が大きい

チューニング①　計算の結合･順序の変更

```
CALL    DIFFx ( Vx, DxVx,  NXP,NYP,NZP,
   :     NXP0,NXPB,NYP0,NYP1,NZP0,NZP1, DX, MOP )

CALL    DIFFy ( Vx, DyVx,  NXP,NYP,NZP,
   :     NXP0,NXPB,NYP0,NYP1,NZP0,NZP1, DY, MOP )

CALL   DIFFz  ( Vx, DzVx,  NXP,NYP,NZP,
   :     NXP0,NXPB,NYP0,NYP1,NZP0,NZP1, DZ, MOP )

-------------------------


 SUBROUTINE  DIFFz (V, DzV, NX,NY,NZ,
   :                NX0,NX1,NY0,NY1,NZ0,NZ1, DZ, M )
      ....
      ....
 DO K = NZ0, NZ1
 DO J = NY0, NY1
 DO I = NX0, NX1
            DzV (I,J,K) = (V(I,J,K+1)-V(I,J,K  ))*R40
   :                    - (V(I,J,K+2)-V(I,J,K-1))*R41
 END DO * 3
 RETURN
 END
```

【旧】同じ変数（Vx）に対する同種の演算を3回実行

```
      DO K = NZP0, NZP1
      DO J = NYP0, NYP1
      DO I = NXP0, NXPB
         DxVx(I,J,K) = (Vx(I,J,K)-Vx(I-1,J,K)  )*R40
   :                 - (Vx(I+1,J,K)-Vx(I-2,J,K))*R41
         DyVx(I,J,K) = (Vx(I,J+1,K)-Vx(I,J,K ) )*R40
   :                 - (Vx(I,J+2,K)-Vx(I,J-1,K))*R41
         DzVx(I,J,K) = (Vx(I,J,K+1)-Vx(I,J,K  ))*R40
   :                 - (Vx(I,J,K+2)-Vx(I,J,K-1))*R41
       END DO*3
```

【新】同じ変数（Vx）に対する演算を一つに結合、データロードを効率化

	修正前[s]	修正後[s]	性能比
T2K	650.8	624.8	1.04
ES2	7.2	7.1	1.14

チューニング②　計算順序の変更

```
DO K = NZP0, NZP1
 DO J = NYP0, NYP1
 DO I = NXP0, NXPB
    DxVy(I,J,K) = (Vy(I+1,J,K)-Vy(I,J,K)  )*R40
    :                - (Vy(I+2,J,K)-Vy(I-1,J,K))*R41
END DO*3

DO K = NZP0, NZP1
 DO J = NYP0, NYP1
 DO I = NXP0, NXPB
      DyVx(I,J,K) =  (Vx(I,J+1,K)-Vx(I,J,K ) )*R40
    :                - (Vx(I,J+2,K)-Vx(I,J-1,K))*R41

END DO*3
```

【旧】方程式通りに変数を用意

```
DO K = NZP0, NZP1
DO J = NYP0, NYP1
DO I = NXP0, NXPB
     DxyVxy (I,J,K) = (Vy(I+1,J,K)-Vy(I,J,K)  )*R40
   :                - (Vy(I+2,J,K)-Vy(I-1,J,K))*R41
   :                + (Vx(I,J+1,K)-Vx(I,J,K ) )*R40
   :                - (Vx(I,J+2,K)-Vx(I,J-1,K))*R41
END DO*3
```

【新】必ず足して使う2変数（DxVy+DyVx）を、1変数（DxyVxy）に再定義し、演算を効率化（全部で6カ所）

（注）プログラムが、方程式と直接対応しなくなり、読みにくくなる

Seism 3D3, 2048x1024x1024, 16x8x8CPU, flat MPI

	修正前[s]	修正後[s]	性能比
T2K	624.8	536.7	1.16
ES2	7.2	7.2	1.01

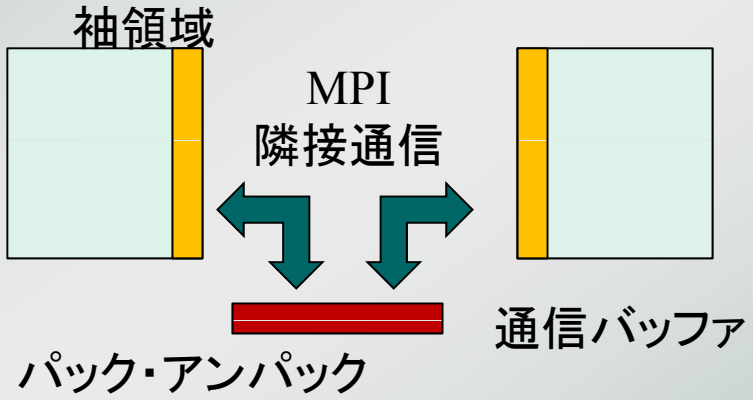
チューニング③　MPI 通信バッファの自作

```
CALL MPI_TYPE_CREATE_SUBARRAY(3,  ISIZE, ISUBSIZE,
      ISTART,  MPI_ORDER_FORTRAN,  MPI_REAL,  ITYPE_K, IERR)
 CALL MPI_TYPE_COMMIT(ITYPE_K, IERR)
 ----------------------------------------------------------------

CALL MPI_ISEND ( Vx, 1, ITYPE_K, KUP, 1,
                    MPI_COMM_WORLD, ITAG1, IERR )
```

【旧】MPI2の「派生データ型」宣言を利用した通信バッファ操作

【新】通信バッファを自作、パック・アンパックを明示的に記述

```
DO  K = 1, NL2
DO  J = 1,  NYP
DO  I = 1,  NXP
      SBUFF_x (I,J,K) = Vx (I,J,NZP-NL2+K)
END DO
END DO
END DO
IBSIZE = NXP*NYP*NL2

CALL MPI_ISEND ( SBUFF_x,  IBSIZE,  MPI_REAL,
:       KUP, 1,   MPI_COMM_WORLD, ITAG1, IERR )
```

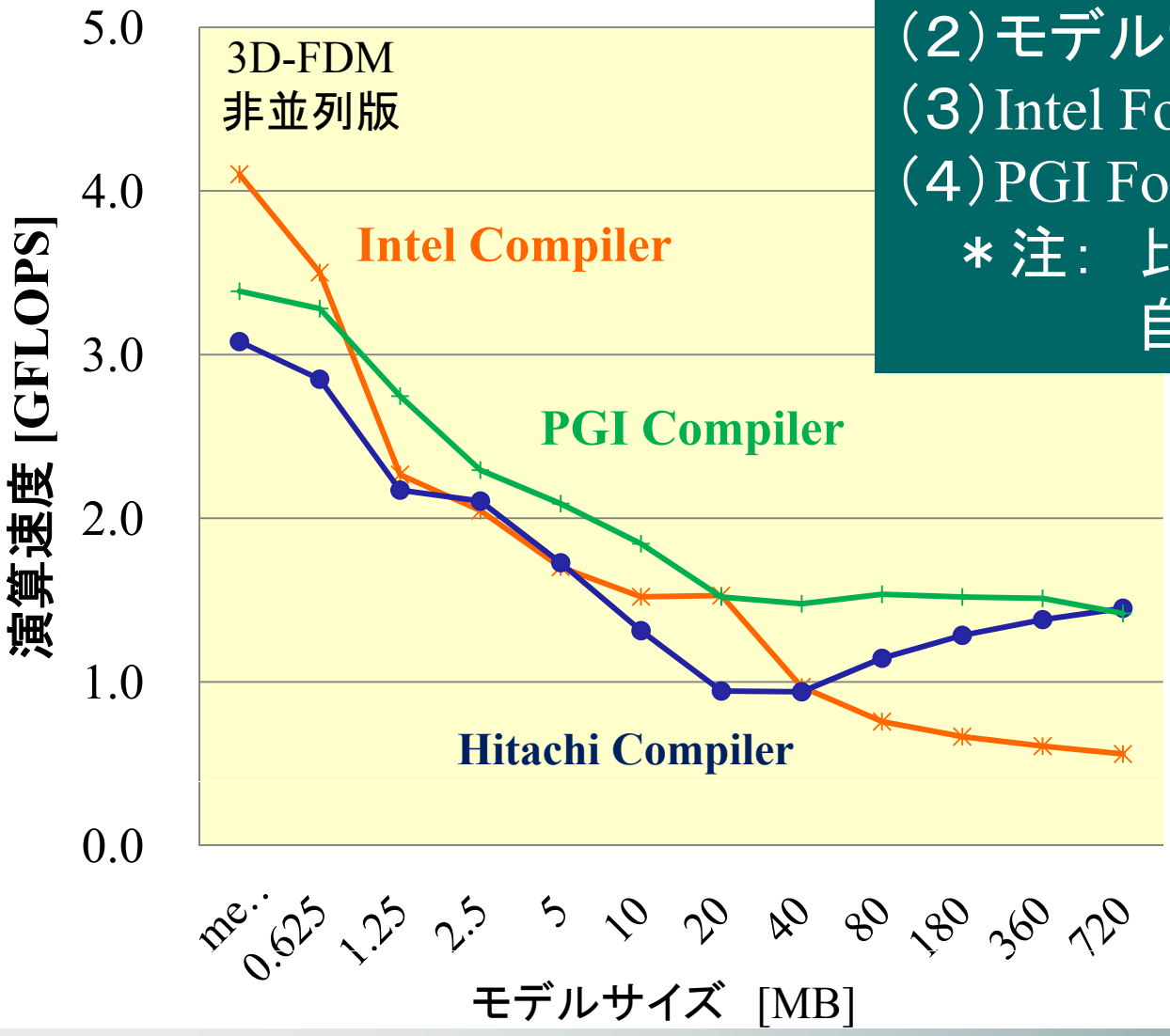


Seism 3D3, 2048x1024x1024, 16x8x8CPU, flat MPI

	修正前[s]	修正後[s]	性能比
T2K	650.8	624.8	1.04
ES2	9.7	7.2	1.34

チューニング（番外）：各社の最適化？コンパイラ

（1）コンパイラにより性能が3割変化
（2）モデルサイズにより性能が変動
（3）Intel Fortranは高速（変動大）
（4）PGI Fortranも高速（速度安定）
＊注：　比較は2008年時点のもの
　　自動並列化性能も要比較



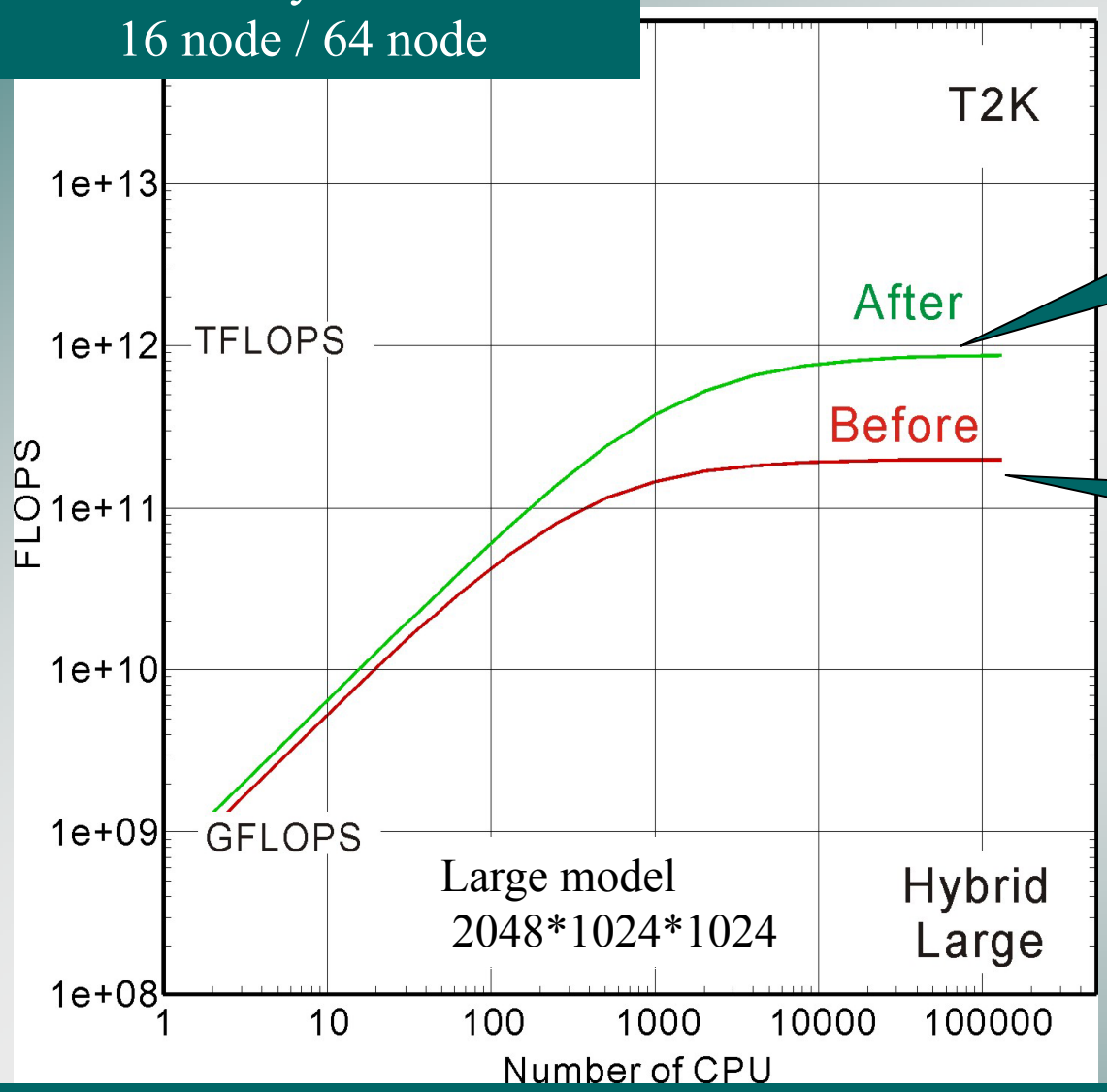
（PGI）　% pgf90 –fast –O3
　　–tp=barcelona64

（Intel）　% ifort　–O3 -fast

(Hitachi) % f90 –Oss –noparallel

チューニング結果（まとめ） 並列計算効率

Thread/MPI Hybrid 並列計算
16 node / 64 node



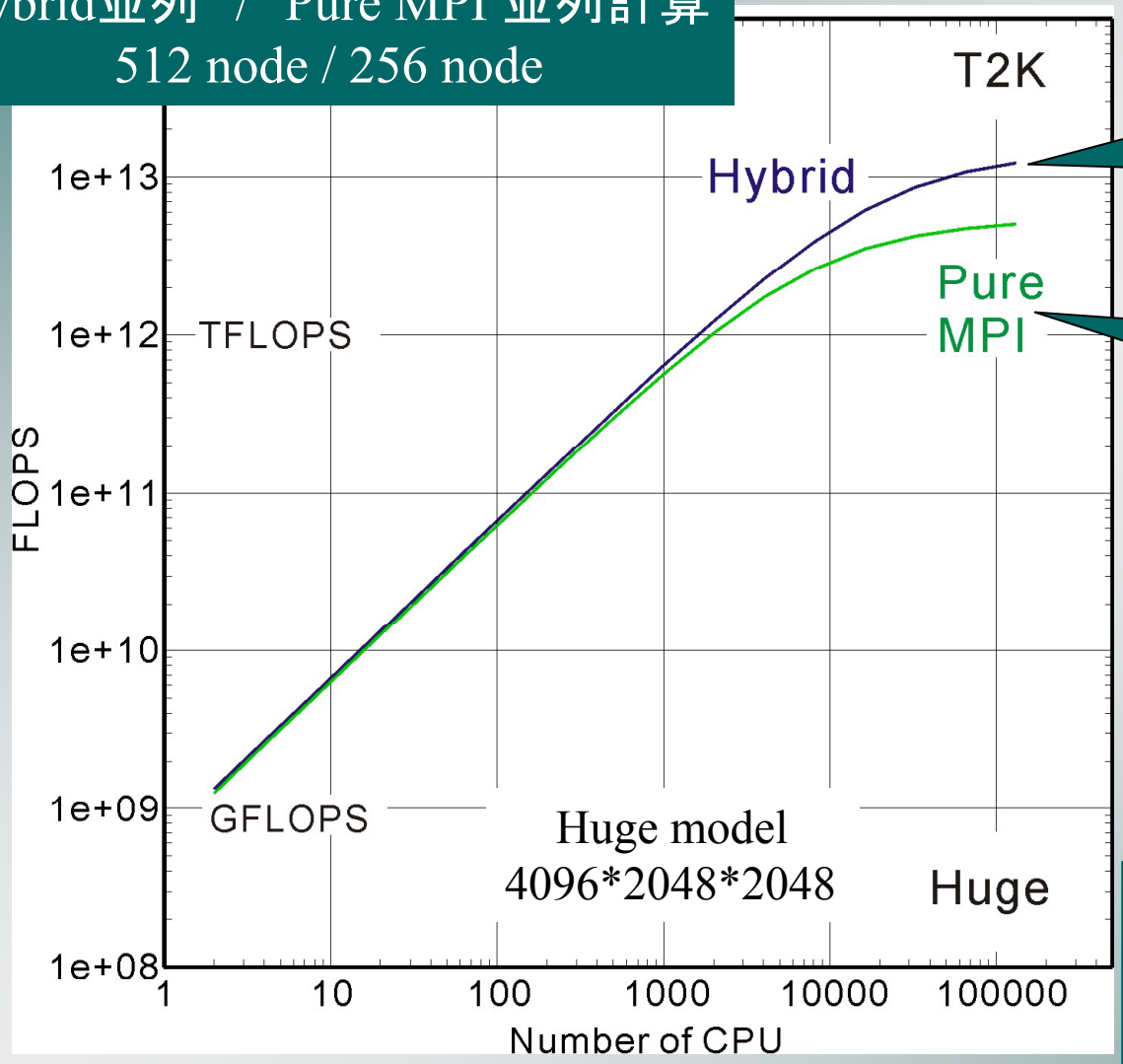
Peak 性能比: 7.1 %
並列化率: 99.92 %

Peak 性能比: 5.8 %
並列化率: 99.72 %

チューニングにより、
- 演算速度
- 並列計算効率　が大幅向上

チューニング結果（まとめ2） Hybrid vs Pure MPI

Hybrid並列 / Pure MPI 並列計算
512 node / 256 node



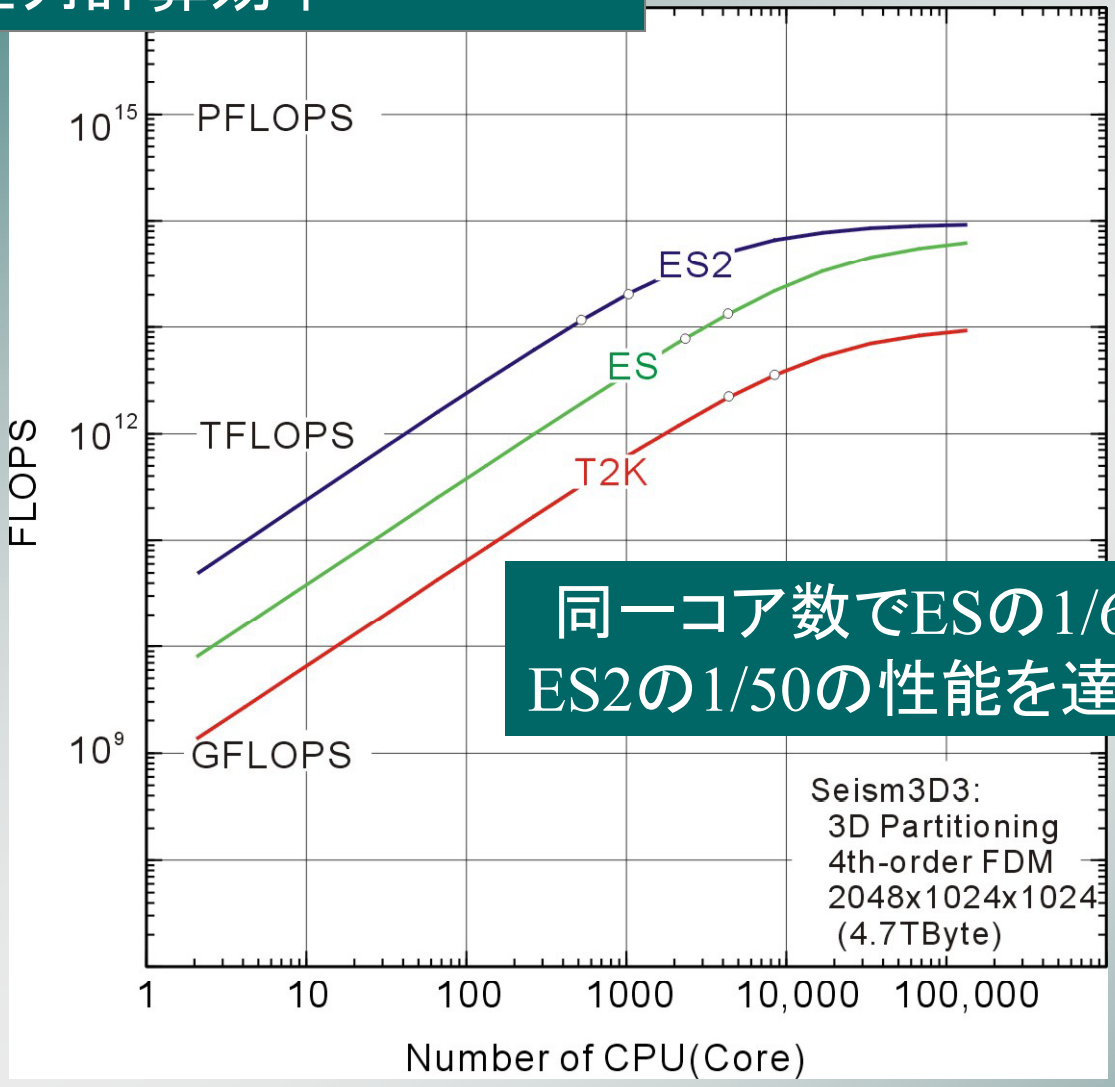
【Hybrid】
Peak 性能比: 6.96 %
並列化率: 99.99 %

【Pure MPI】
Peak 性能比: 6.21 %
並列化率: 99.98 %

Multi, Many Core CPUでは
Thread / MPI Parallel
のハイブリッド並列化が必須

大規模地震動シミュレーションにおける演算速度

地震動シミュレーション：並列計算効率



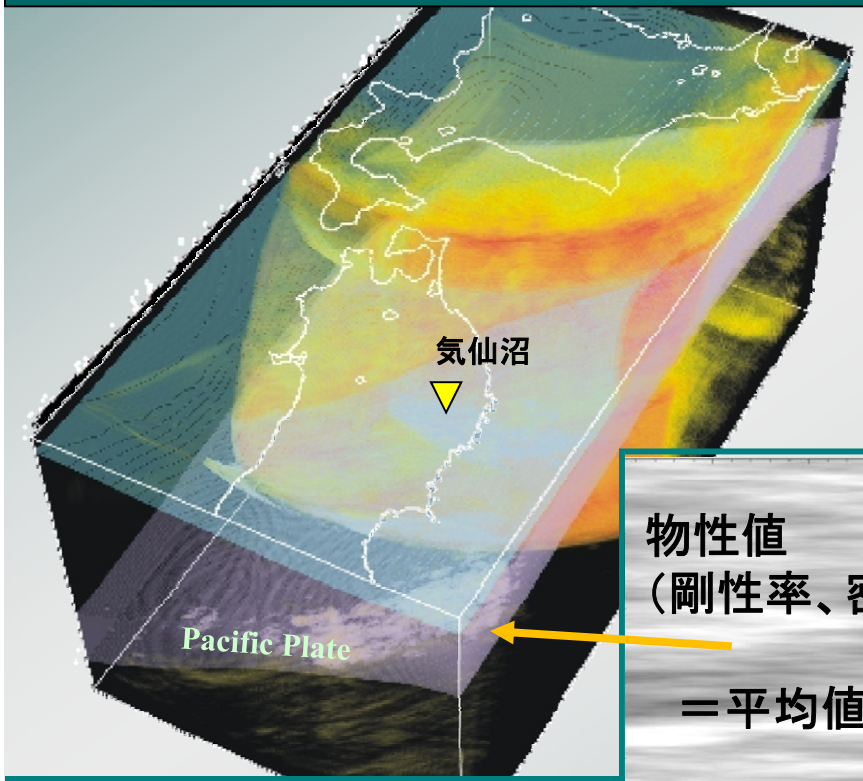
同一コア数でESの1/6,
ES2の1/50の性能を達成

次世代スパコンによる大規模地震・津波シミュレーションに向けて

出展：理研次世代スパコン開発本部HP

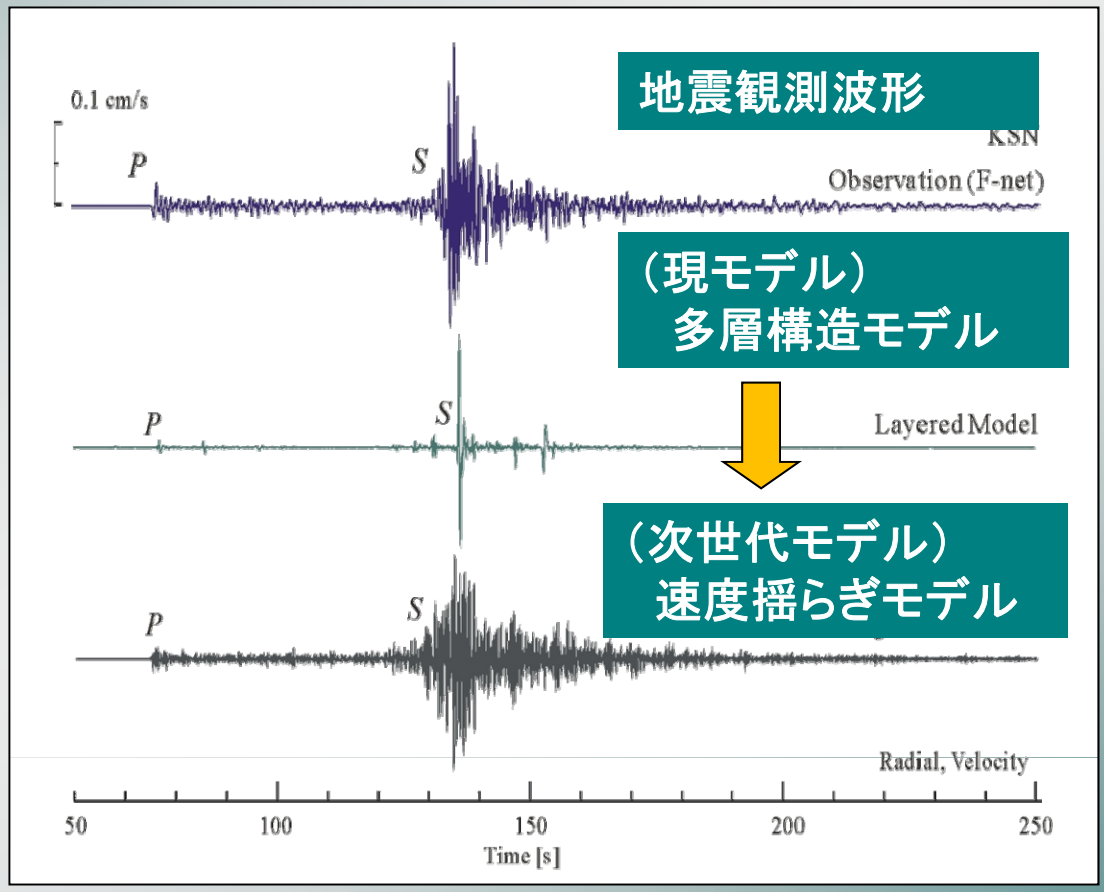
広帯域、短周期（T<1s）地震動の高精度評価

短周期地震動の散乱評価
- 短波長(<100m)不均質構造のモデル化
- 高分解能計算



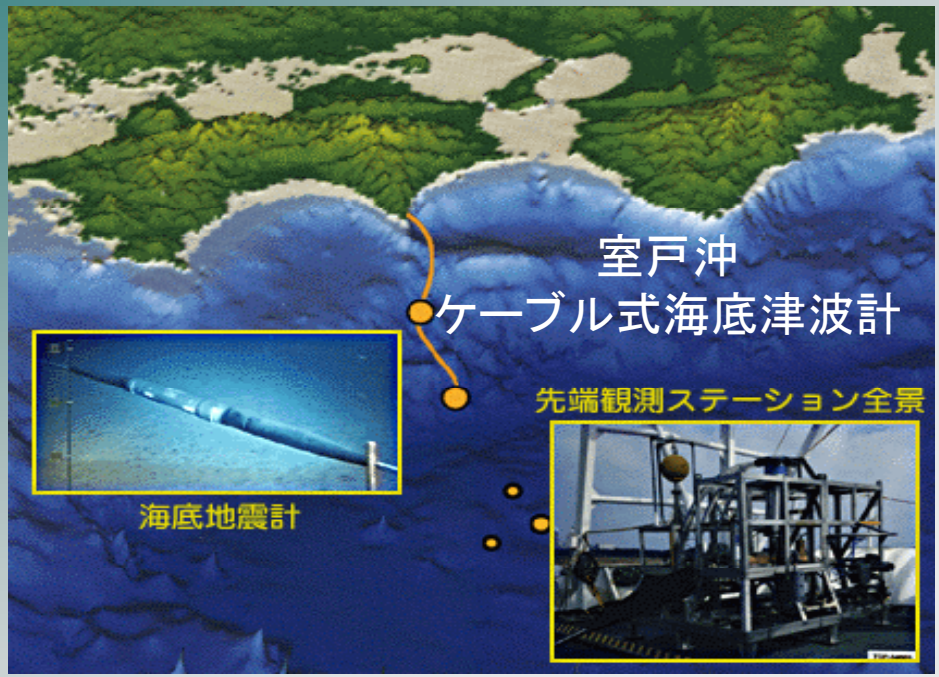
物性値
（剛性率、密度etc）:

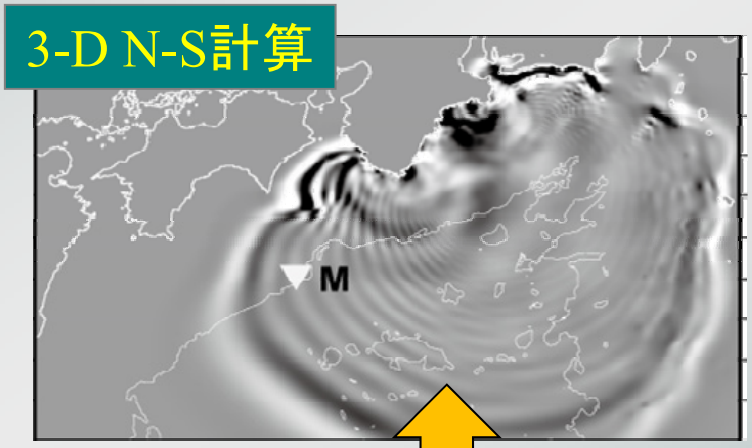
＝平均値＋揺らぎ

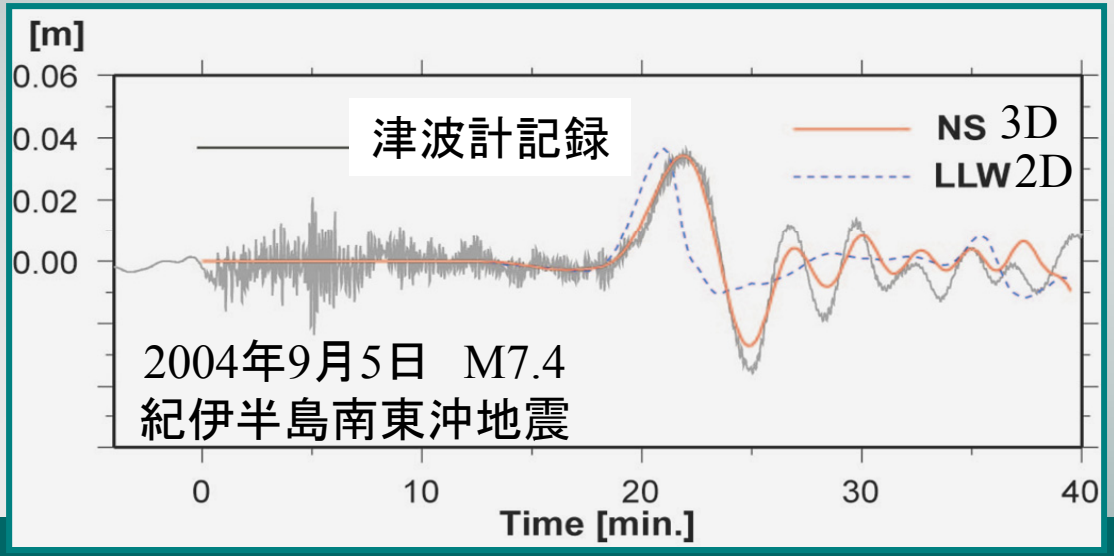


計算規模:
- 分解能40m（現在の1/5）
- 計算規模 5^4=625倍

津波発生伝播シミュレーション：2004年紀伊半島沖地震(M7.4)



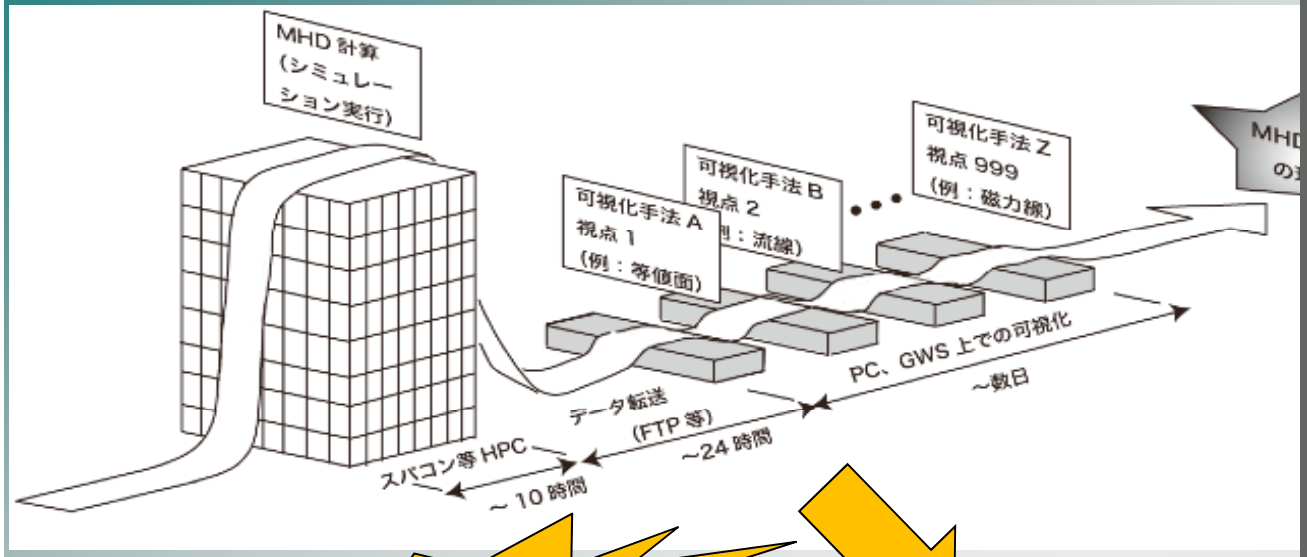


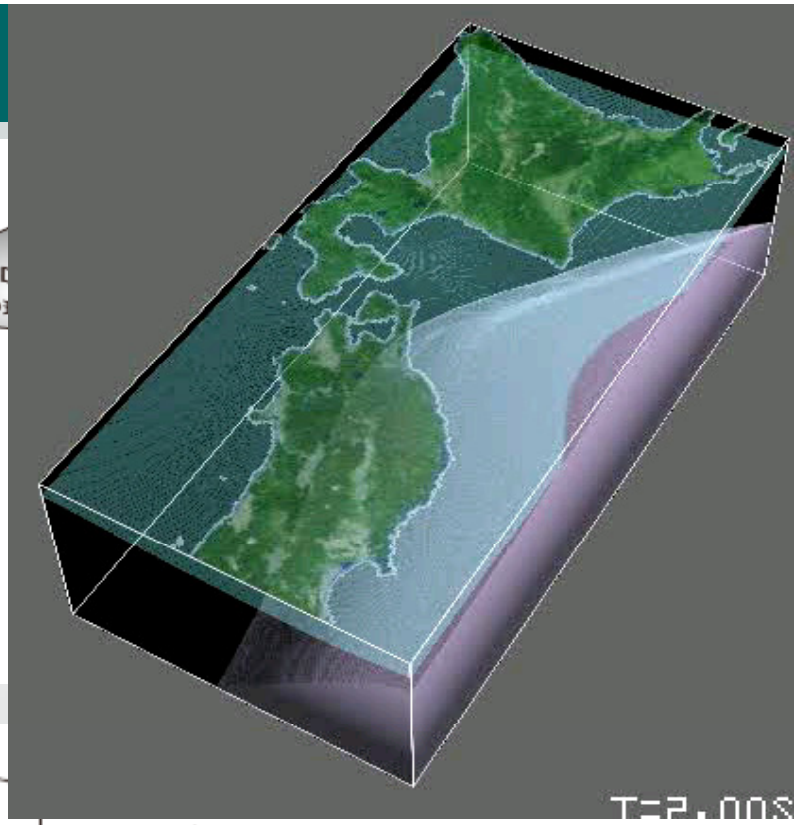


Saito and Furumura (2009)

- 沖合津波計の実現、高精度(3D)津波波形計算が必要
- 2D→3D計算：300倍の規模

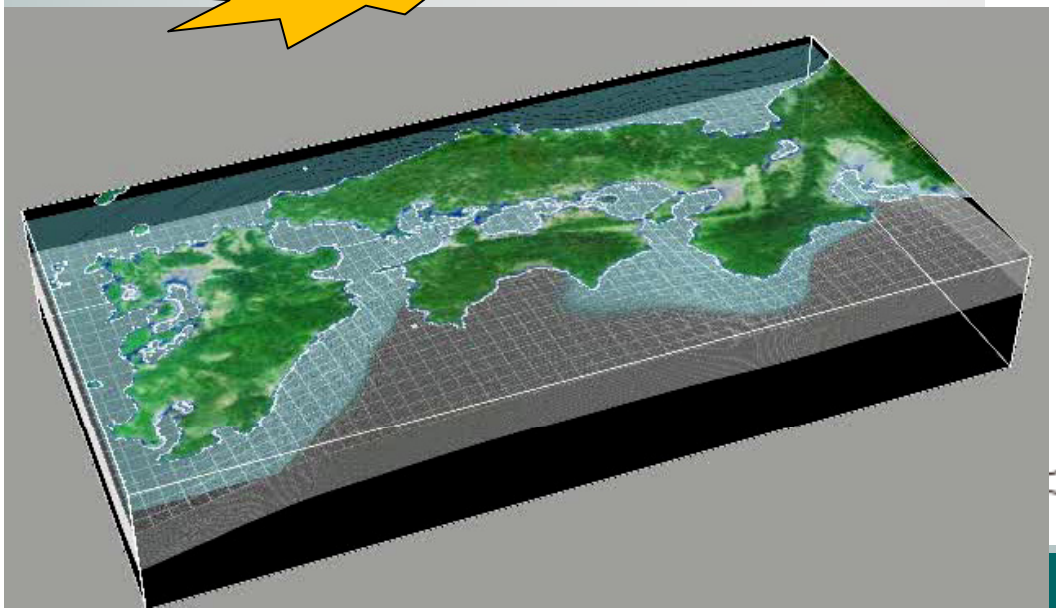
大規模計算と同時Volume可視化

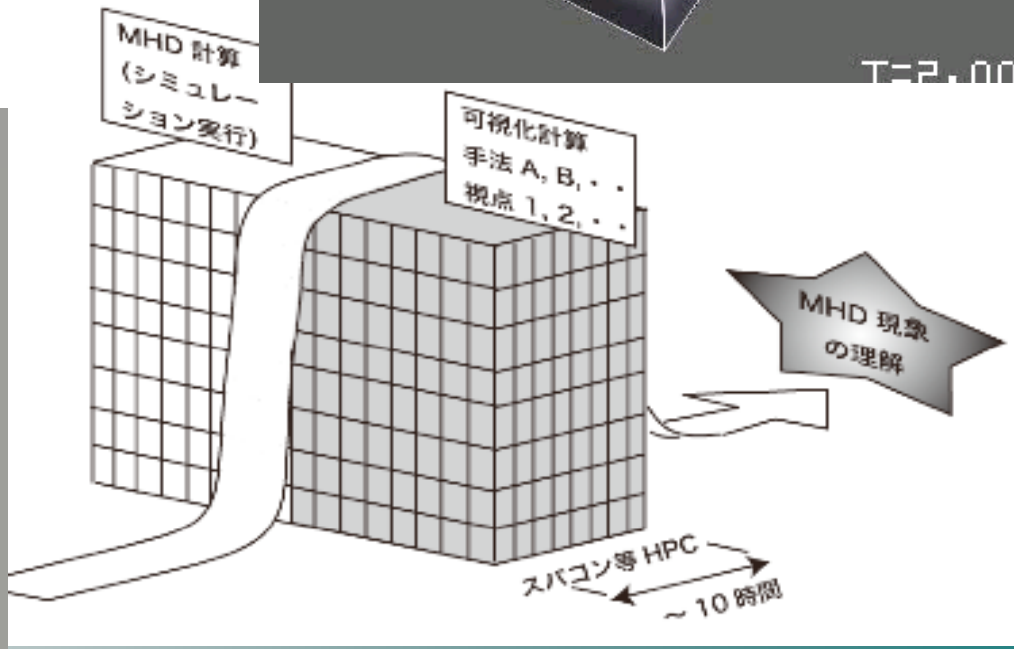




同時可視化

数十万コアの有効利用



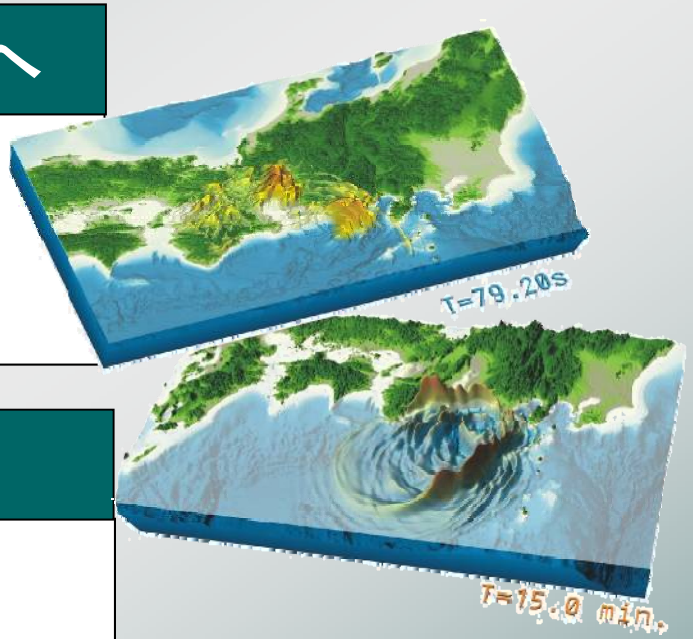


図： 神戸大、陰山聡教授による

T2Kによる大規模地震津波計算　ーまとめと課題ー

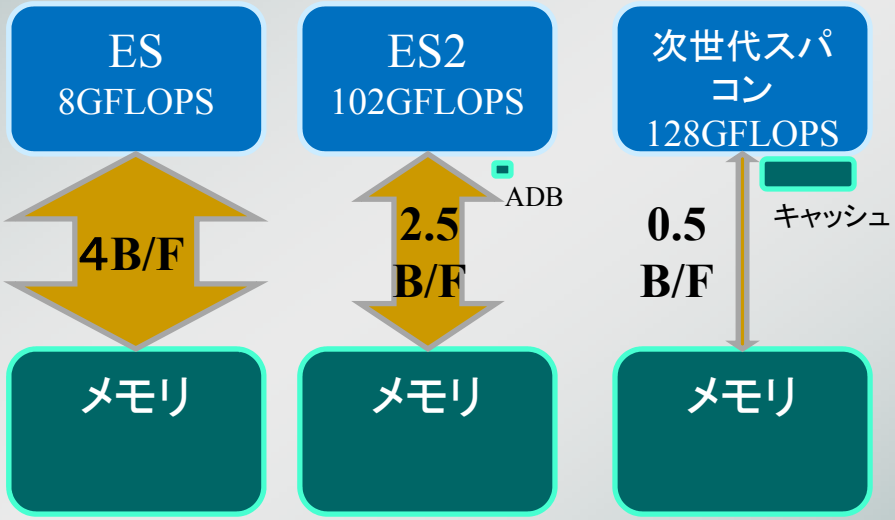
（1）地震と津波の高精度予測・災害軽減へ

- 広帯域（f>5Hz）シミュレーション（計算量は625倍）
- 高精度津波観測に対応する高精度津波計算（300倍）
- 地震、水中音波、地殻変動、津波の連成計算



（2）大規模並列計算の求めるもの

- 高精度化以上の成果（突然見えてくる世界）
- 次(次)世代スパコンに向けた新たな計算物理の創出



（3）次世代スパコンに向けた課題

- メモリ速度の低下（スカラー、ベクトル共通）
- ユーザ主体のコード（高度）チューニング
- 大容量可視化（4次元データ）