

イントロダクション

中島 研吾

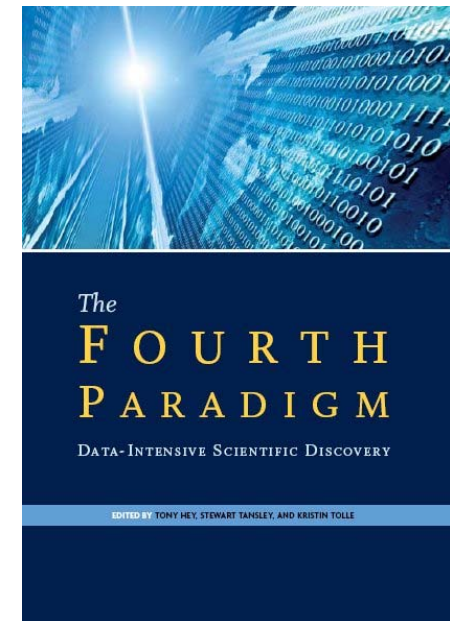
東京大学情報基盤センター

並列計算の意義

- たくさんの計算機を使って
- より速く
- より大規模に
- より複雑に
 - 連成, 連結シミュレーション

科学

- 理論 (Theoretical Science) と実験 (Experimental Science) が2本の柱
- 計算科学 (Computational Science)
 - 第3の柱 (The third pillar of science) , 第3の科学
 - 計算機シミュレーション
- 第4の科学
 - 大規模データに基づく予測と発見
 - 実験・観測, 計算
 - Big Data

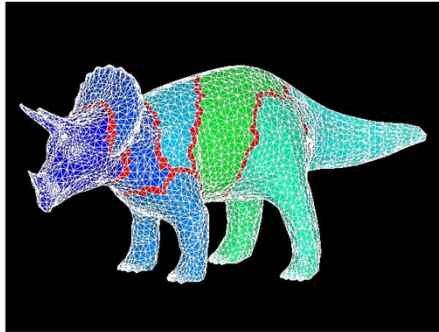


計算科学 (Computational Science)

- 大型計算機（スーパーコンピュータ，スパコン）を駆使して，実験が不可能な現象，事象をモデルにより表現
 - 宇宙科学，地球科学
 - 人体シミュレーション
 - ナノサイエンス，マテリアルサイエンス
- 解明されていないことは多い
 - 実測データが無いと検証できないこともある
 - 観測・実験との協力の重要性
 - 計算機のパワー不足
 - 最高速のスパコンでもまだまだ足りない分野はある

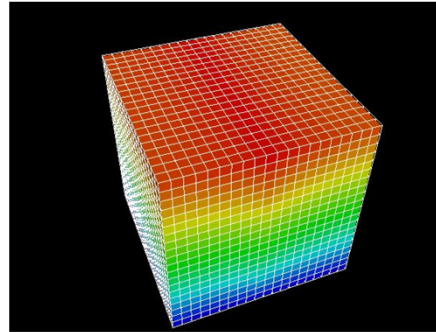
科学技術計算の手法

- 空間を細かい格子，粒子に分割
 - 細かい（粒子が多い）ほど精度高い，計算量多い



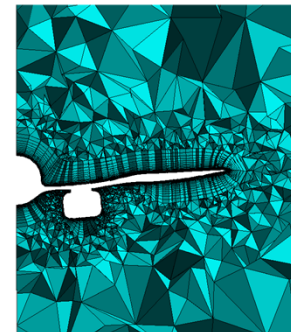
有限要素法

Finite Element Method
FEM



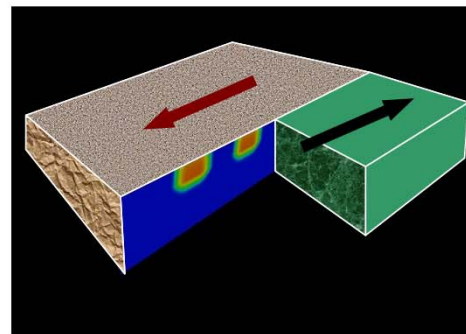
差分法

Finite Difference Method
FDM



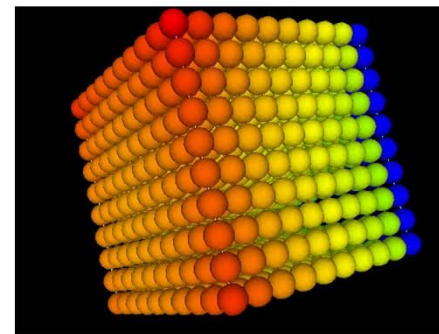
有限体積法

Finite Volume Method
FVM



境界要素法

Boundary Element Method
BEM

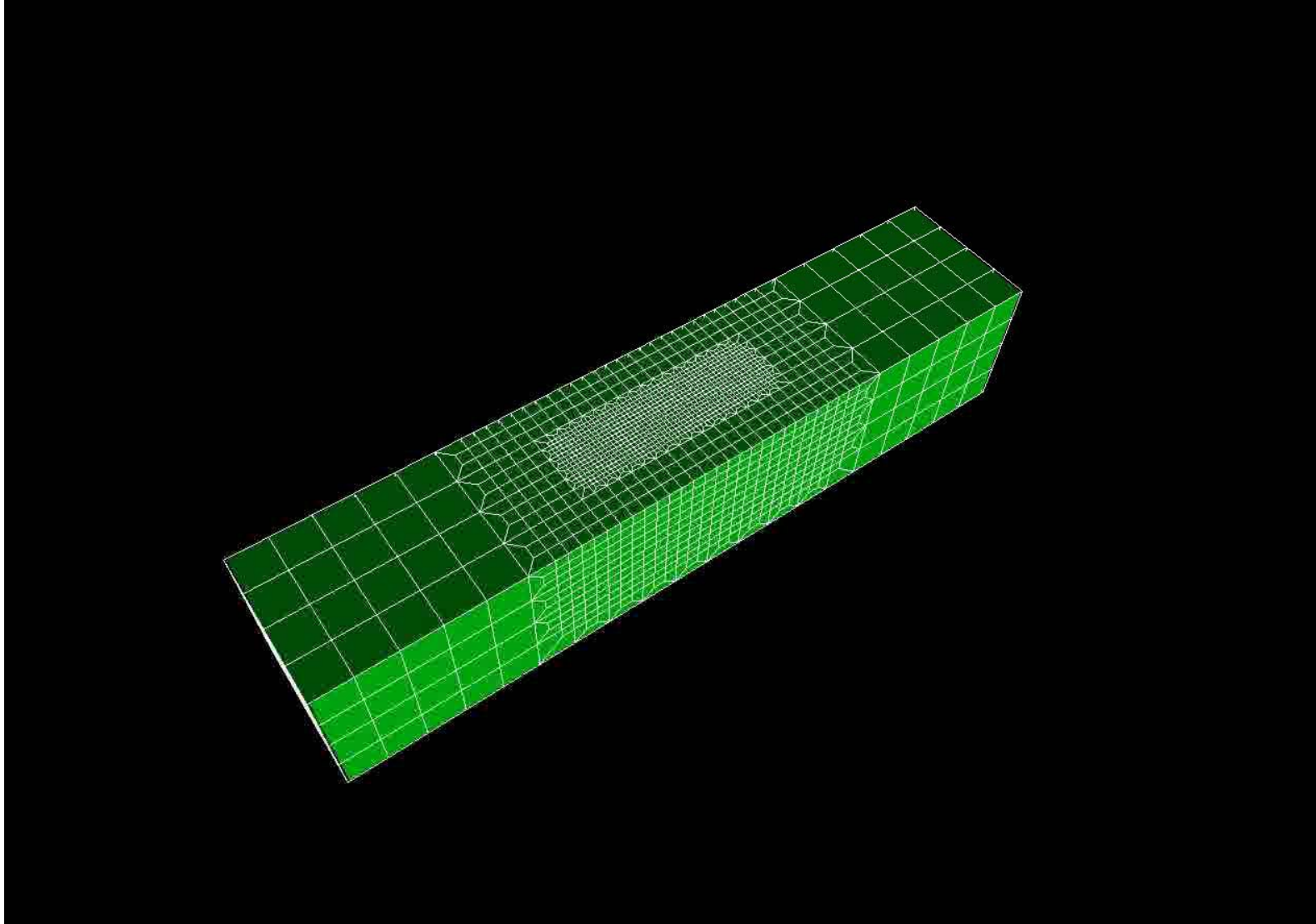


個別要素法

Discrete Element Method
DEM

有限要素法の例：横ずれ断層

- 変化量の大きいところでは細かい格子が必要



科学技術計算の真髄：SMASH

- 幅広い知識と視野の必要性
- 分野間の協力の必要性も示している
- 分野間の協力には密接なコミュニケーションが不可欠，そのためには多少の知識はなければならない
- 「科学」の無い科学技術計算はあり得ない
 - したがって，対象アプリケーション無しに科学技術計算の講義は成立しない
 - 全てをカバーすることはできないため，ここでは並列有限要素法とその周辺を扱う
 - 他分野にも容易に拡張可能な考え方である
- 並列化に重要なのはアプリ自身を深く理解すること

Science

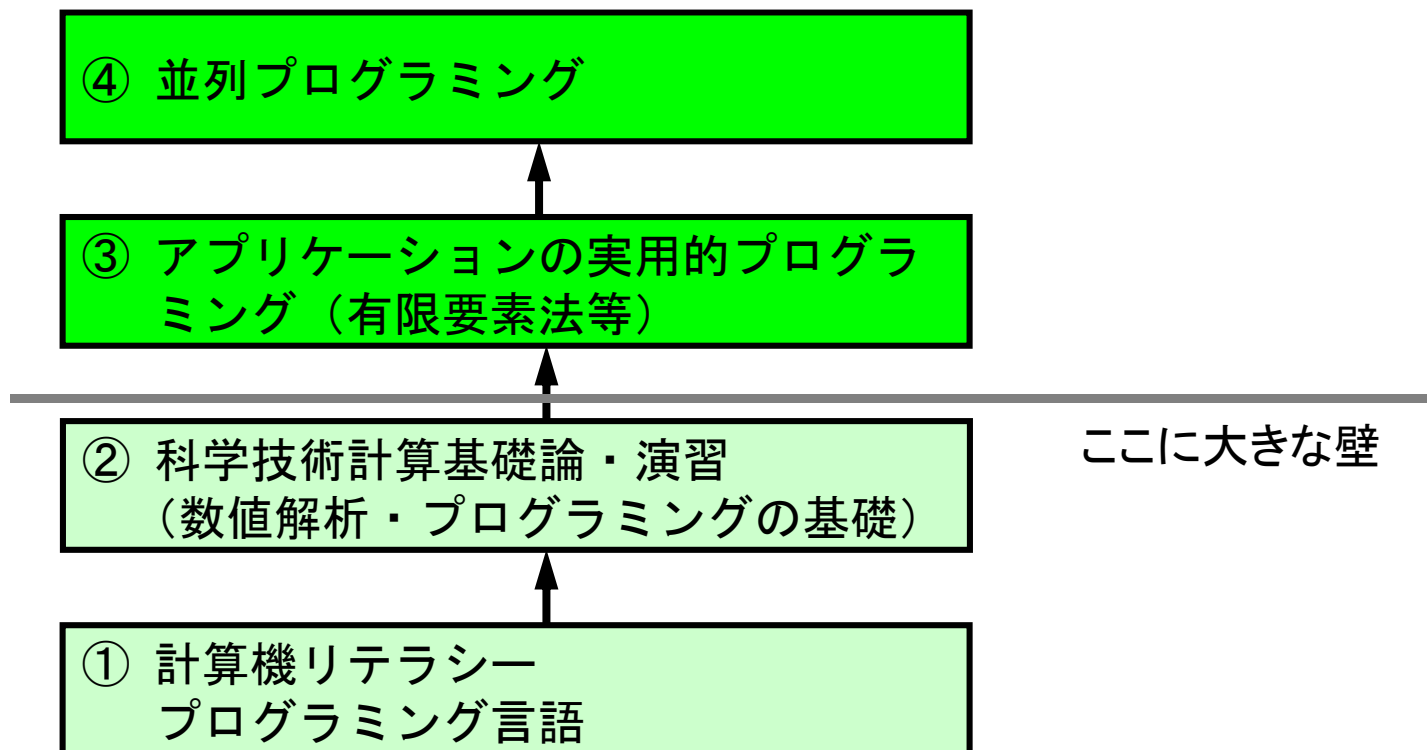
Modeling

Algorithm

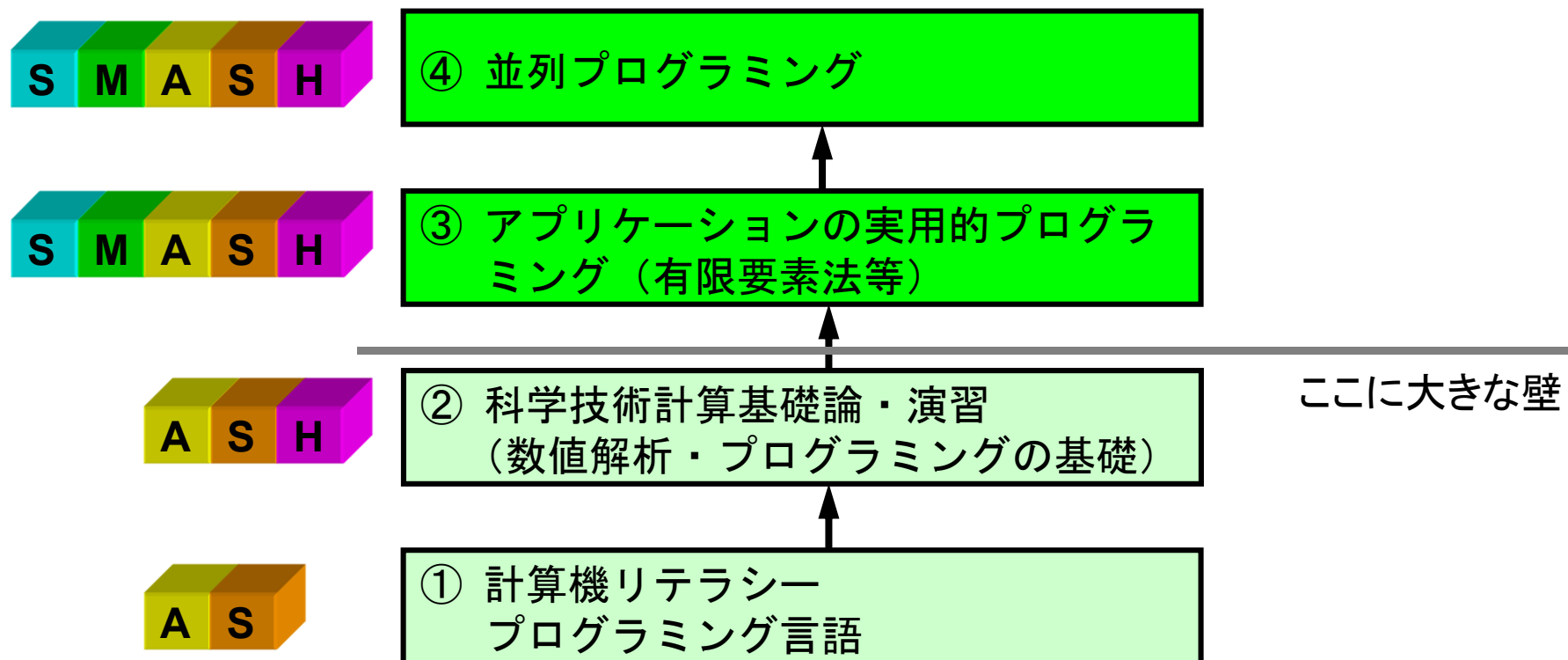
Software

Hardware

並列プログラミングへの道



並列プログラミングへの道



本Summer Schoolの目的 (1/2)

- 科学技術計算における様々な手法・アルゴリズムを理解し，各アルゴリズムにおいて並列性を引き出す様々な考え方を学ぶ
 - まずは，元のアプリケーションの中身（アルゴリズム，実装）をよく理解していること，そうすれば「並列化」は難しくくない
- 並列計算プログラミング技法の習得
 - MPI, OpenMP
 - 情報の探し方の習得：これは重要
- SMASH
 - ASHが中心となるが，Science, Modelingを抜きにして，科学技術計算プログラミングは語ることは不可能である
 - できるだけ誰でも理解できる「SM」を選択

本Summer Schoolの目的 (2/2)

- 本Summer Schoolでは，有限要素法による熱伝導解析プログラムをMPIを使用して並列化するための手順，特に並列分散データ構造に関する考え方を中心に説明します。
 - <http://nkl.cc.u-tokyo.ac.jp/seminars/2015-AICSsummer/>

背景

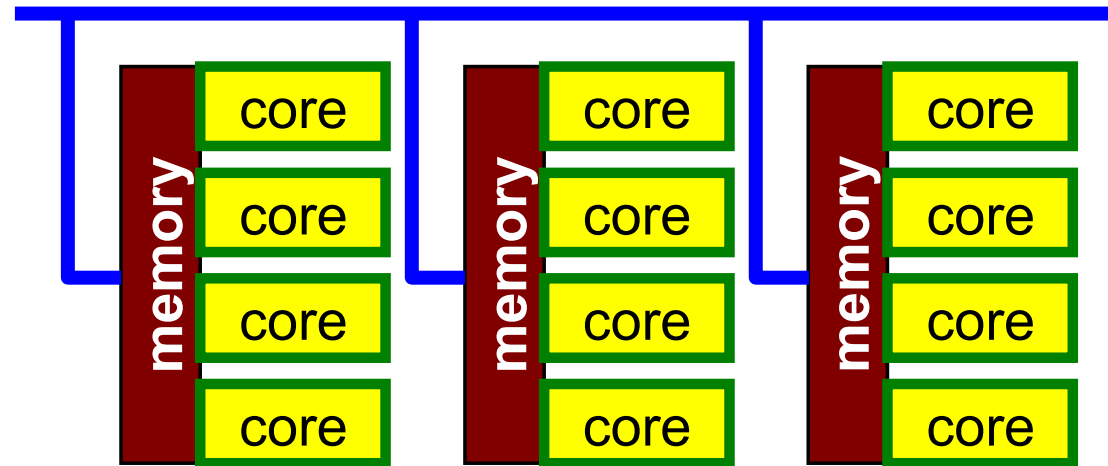
- 大規模並列計算機
- ポストペタスケール, エクサスケール時代の到来
- 並列プログラミングの重要性
 - 体系的に学習する機会は少ない
 - アプリケーションの観点から教えている材料は少ない (ほとんど無い)
- 計算機アーキテクチャの多様化, 複雑化
 - Multicore, Manycore, GPU
 - MPI+X(+Y)
 - X : OpenMP, CUDA, OpenACC, OpenCL
 - Y : Vectorization
- Co-Designの重要性

Hybrid並列プログラミングモデルは 必須

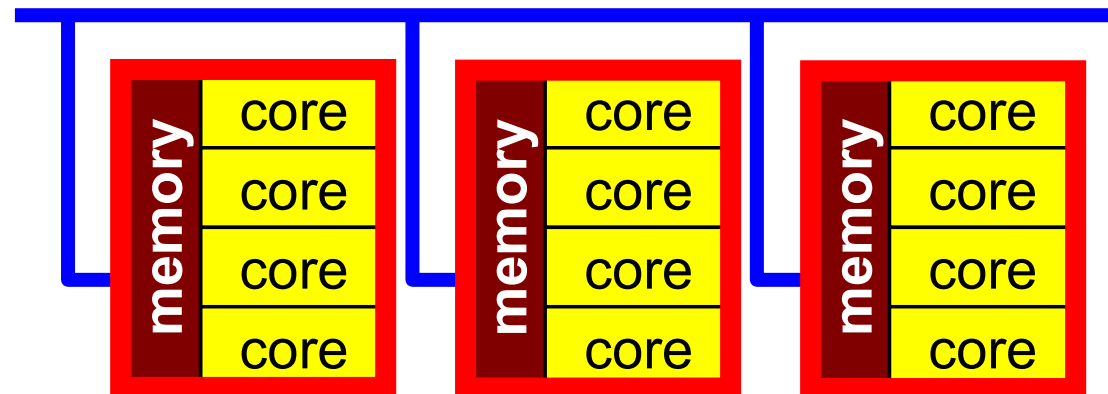
- Message Passing
 - MPI
- Multi Threading
 - OpenMP, CUDA, OpenCL, OpenACC
- 「京」, 「FX10」でもHybrid並列プログラミングモデルが推奨されている
 - 但し MPI+自動並列化 (ノード内)

Flat MPI vs. Hybrid

Flat-MPI: Each PE -> Independent



Hybrid: Hierarchical Structure



本Summer Schoolの位置づけ

- 有限要素法にターゲットを絞る
 - 疎行列, 共役勾配法
- MPI, 簡単なMPI+OpenMPハイブリッド並列プログラミングまで教える
- 計算機科学の人がアプリケーションの勉強をするのにも適している
- 本来10日かける内容を5日で実施
 - よく復習してください。
- 前処理付き反復法
 - 実は非常に重要な研究開発分野なのだが, 本講習会では簡単なもののみ教える。
 - より詳しい内容はSpring Schoolで (2016詳細不明)