INFORMATION TECHNOLOGY CENTER
THE UNIVERSITY OF TOKYO

低精度演算を活用した計算科学シミュレーション

中島研吾（東京大学情報基盤センター）

THE UNIVERSITY OF TOKYO

計算機の中で数はどのように表現されているのか？

- 2進数で処理
  - （0,1）例：001011
  - 1ビット（bit）：1桁
- 1バイト（byte）
  - 処理の単位，8ビット
- 実数
  - 倍精度型：8バイト，64ビット（2進数64桁）
  - 単精度型：4バイト，32ビット（2進数32桁）
    - 有限の桁しか記憶できない
- **計算機内では正規化（Normalized）**
  - **符号部：sign**
  - **指数部：exponent**
  - **仮数部：fraction：2進法の小数第一位が1**

$$10 = 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 1 \times 0 = (1010)_2$$

$$100 = 2^6 \times 1 + 2^5 \times 1 + 2^2 \times 1 = (1100100)_2$$

$$(1111)_2 = 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1 = 15$$

$$(.0101)_2 = 2^{-1} \times 0 + 2^{-2} \times 1 + 2^{-3} \times 0 + 2^{-4} \times 1 = 0.3125$$

$$0.5 = 2^{-1} \times 1 = (0.1)_2$$

$$-0.1 = -(0.000110011.....)_2$$

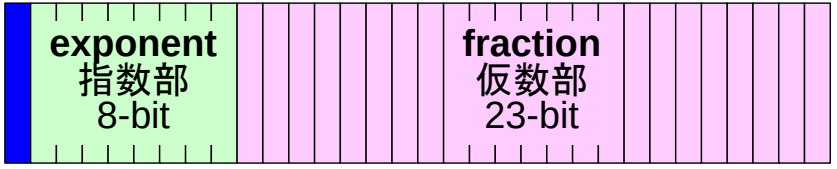
$$= (-1)^1 \times 2^{-3} \times (0.1100110011...)_2$$

sign 符号部　exponent 指数部　fraction 仮数部

単精度実数（32bit）と倍精度実数（64bit）
IEEE 754形式，符号部（sign ■）は1-bit

単精度： $(-1)^{符号部} \times 2^{指数部-127} \times (1+仮数部)$

最小値： 1.18×10^{-38}
最大値： $3.40 \times 10^{+38}$

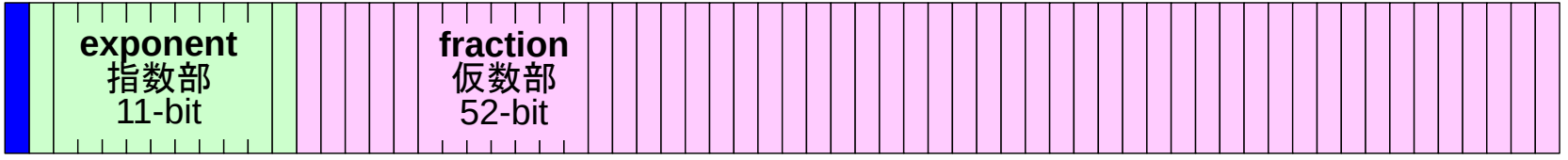


$$-0.1 = -(0.000110011.....)_2$$
$$= (-1)^1 \times 2^{-3} \times (0.1100110011...)_2$$

sign 符号部　exponent 指数部　fraction 仮数部

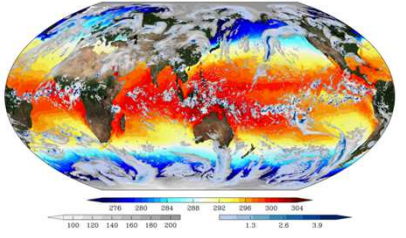
倍精度： $(-1)^{符号部} \times 2^{指数部-1023} \times (1+仮数部)$

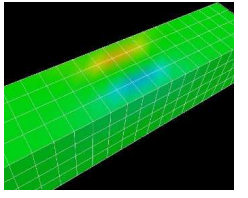
最小値： 2.23×10^{-308}
最大値： $1.80 \times 10^{+308}$

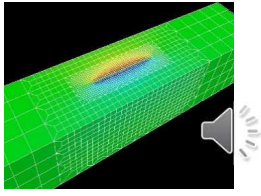


低精度演算の活用

- スパコンは様々な計算科学シミュレーションに使用されている。
  - 一般に，倍精度（64bit）演算によって実施されている
- 単精度（32bit）演算では，扱える数の範囲が狭いが，倍精度演算と比較して変数当たりのデータ量が半分であり，より効率的な計算が可能
  - 計算量・計算時間・消費電力・メモリ容量・I/O削減
  - 計算精度・安定性が低下する可能性もある
- 我々は不必要に高精度な計算を実施して貴重な時間やエネルギーを無駄にしている・・・かも知れない
- 昨今，低精度演算の活用は盛ん
  - 混合精度演算の研究，様々なアプリケーションへの適用は既に行われている
  - Approximate Computing：省電力のための研究
    - 元々は画像処理

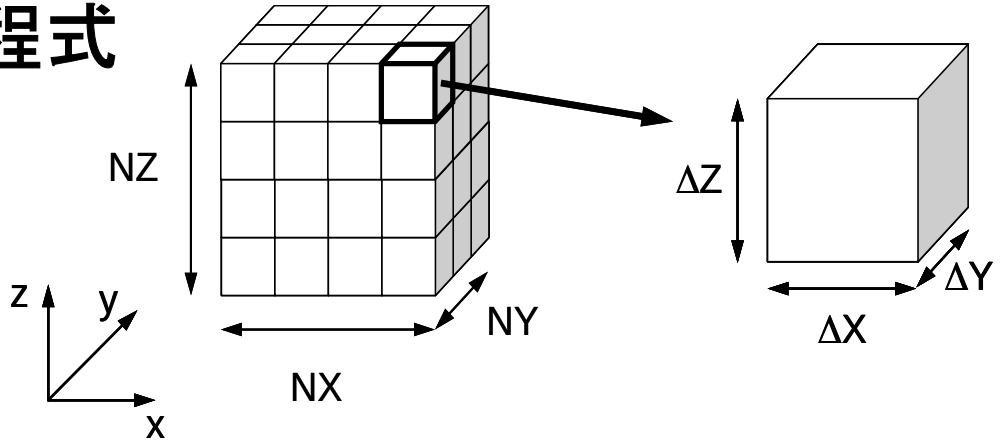




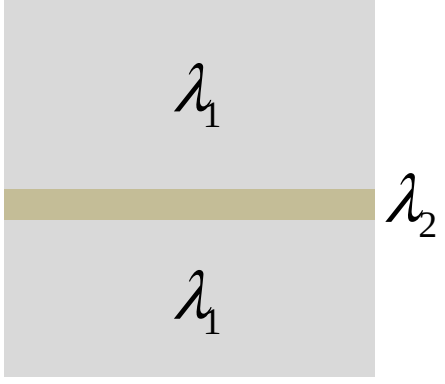


三次元定常熱伝導方程式

- 有限体積法(FVM)プログラム
  - 規則正しい形状，7点差分
  - 一層だけ異なる熱伝導率λ_2
  - 前処理付き反復法:ICCG法
  - Fortran 90 + OpenMP



- 問題設定
  - λ_1/λ_2の値:係数行列の条件数に比例
    - 大きいほど解きにくい(収束しにくい)
  - 倍精度演算・単精度演算
  - N=128^3
- 計算機環境
  - 東大Reedbush-U(Intel Xeon BDW)，1ノード，36コア
- 計算時間・消費電力(W)・消費エネルギ(J)測定

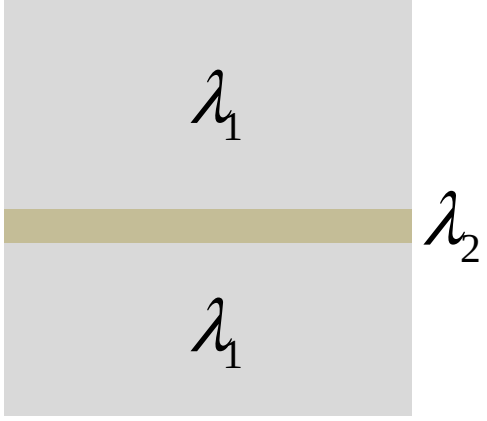


$$\nabla \cdot (\lambda \nabla \phi) + f = 0$$

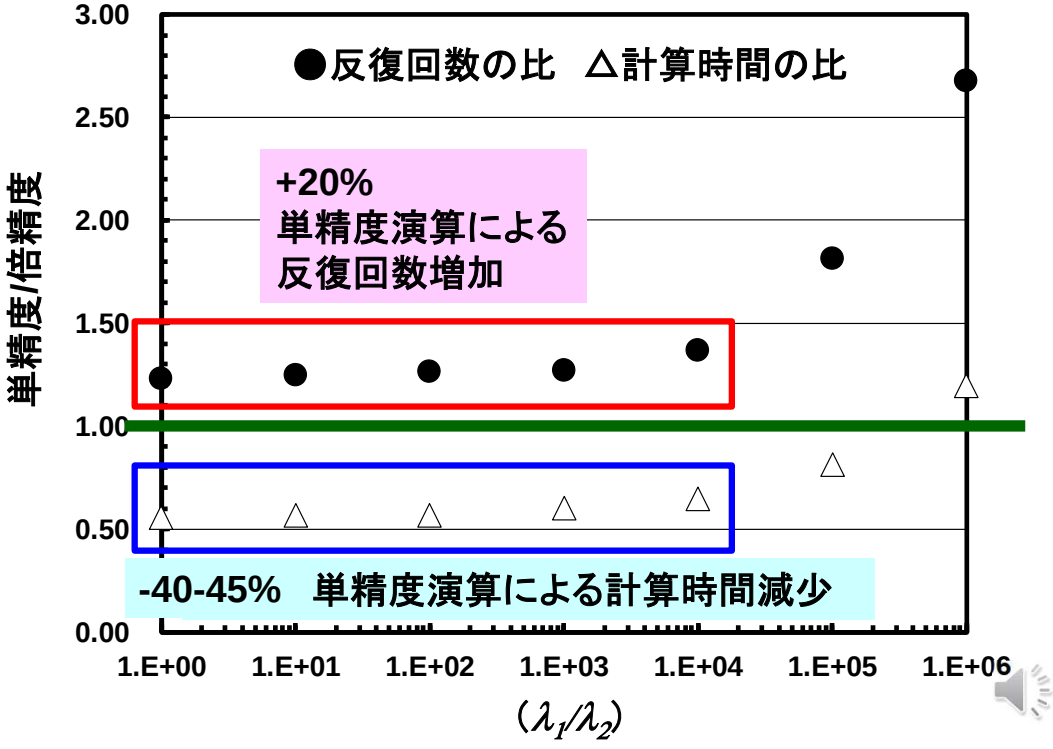
(λ_1/λ_2)とICCG法反復回数・計算時間の関係

単精度／倍精度の比：1より小⇒単精度が反復回数少ない・計算時間速い

(λ_1/λ_2)小：単精度でもOK・速い，(λ_1/λ_2)大：単精度は不安定・答えも不正確

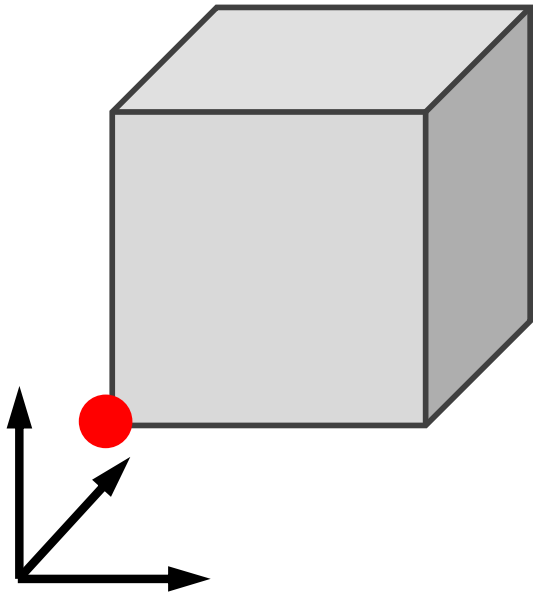


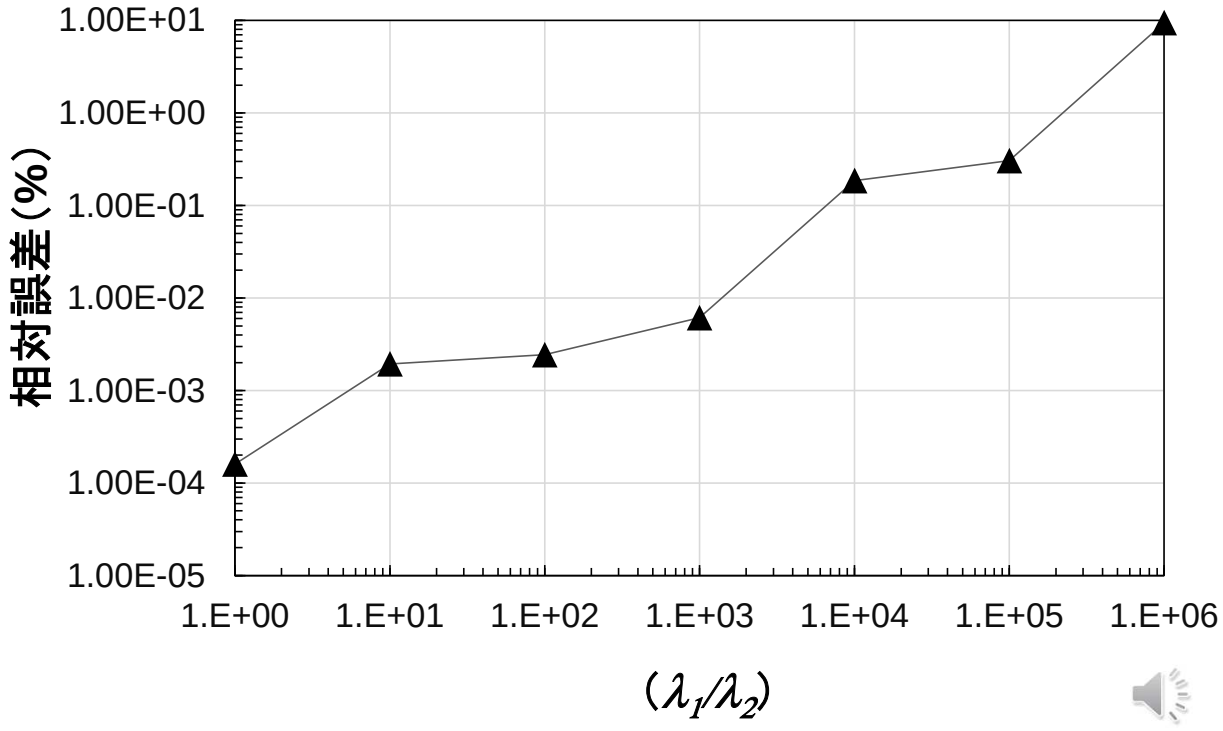
$$\nabla \cdot (\lambda \nabla \phi) + f = 0$$



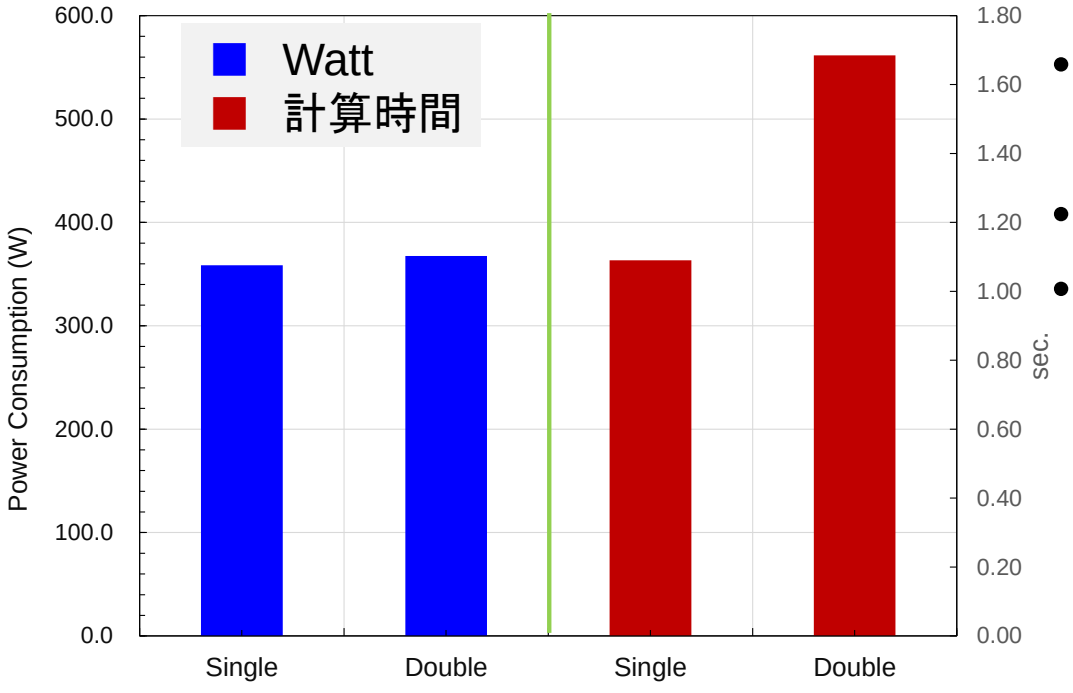
●点での相対誤差（倍精度による結果との相違）

(λ_1/λ_2)大：見かけ上収束しているが誤差大きい⇒精度保証によっても確認



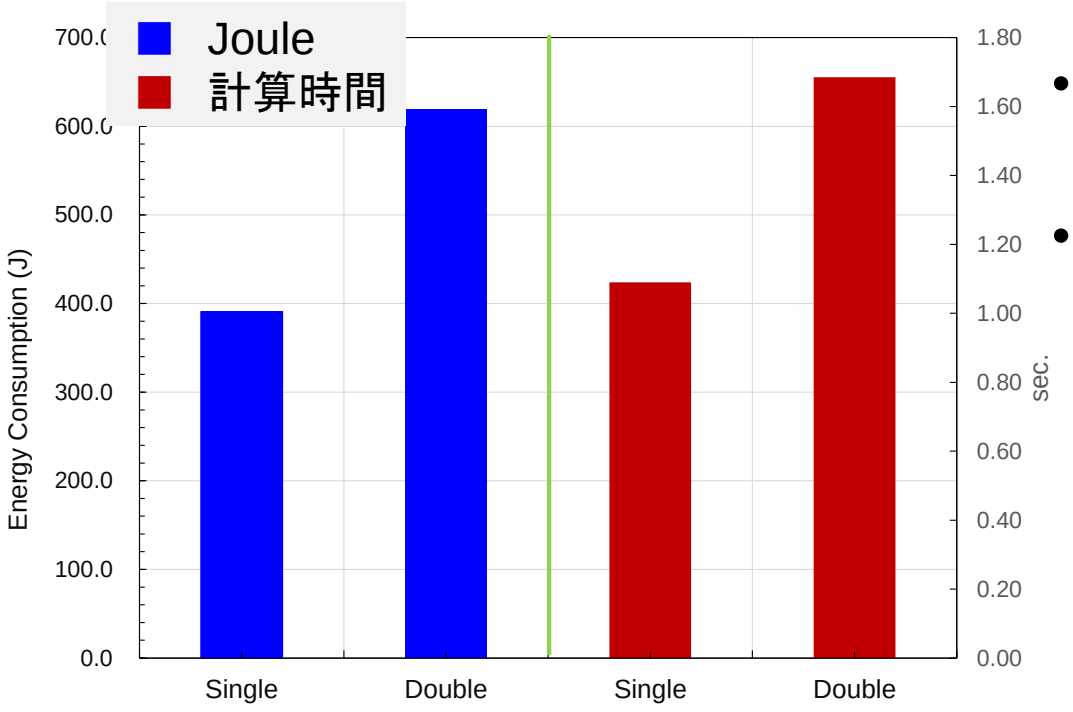


ICCG法計算時間･消費電力(W)



- 倍精度⇒単精度
- 計算時間減少
- 消費電力(W)不変

ICCG法計算時間・消費エネルギ(J)



- 倍精度⇒単精度
- 計算時間に比例して消費エネルギ(J)減少

参考文献

① 坂本龍一，近藤正章，中島研吾，藤田航平，市村強，HPCアプリケーションにおける低精度演算の積極的利用による電力効率改善の検討，情報処理学会研究報告（2018-HPC-167-19），2018

② 中島研吾，坂本龍一，星野哲也，有間英志，塙敏博，近藤正章，低精度演算とアプリケーション性能，情報処理学会研究報告（2020-HPC-174-5），2020

③ 中島研吾，荻田武史，塙敏博，河合直聡，伊田明弘，星野哲也，低精度・混合精度演算による高性能・高信頼性疎行列ソルバー，情報処理学会研究報告（2020-HPC-175-2），2020

✓ 精度保証

まとめ

- 条件の良い問題では倍精度（64bit）⇒単精度（32bit）とすることによって反復法の反復回数は増加するものの，計算時間は大幅に低下する
  - 条件が悪くなると単精度は反復回数増加，答えも不正確（精度保証）
  - 計算機環境，問題設定，実装方法によって計算時間の変動の割合は異なる
- 消費電力（W）は精度により不変，消費エネルギ（J）は計算時間に比例

Thank you for watching