

「TSUBAME」を用いた格子ボルツマン法による
大規模ラージエディ・シミュレーション

小野寺 直幸
東京工業大学
学術国際情報センター

「TSUBAME」を用いた格子ボルツマン法による 大規模ラージエディ・シミュレーション

小野寺 直幸, 青木 尊之, 下川辺 隆史

東京工業大学・学術国際情報センター
〒152-8550 東京都目黒区大岡山 2-12-1
onodera@sim.gsic.titech.ac.jp

自動車の空力解析や都市の気流解析等を高精度に実施するためには、物体壁面の乱流境界層と計算対象全体を捉えた大規模計算が不可欠である。東京工業大学のスーパーコンピューター TSUBAME は、アクセラレータとして GPU を 4000 台搭載することで、5.76 PFLOPS の非常に高い演算性能を達成している。GPU の高い演算性能を引き出すためには、局所的なメモリアクセス且つ演算密度の高いアルゴリズムを選択する必要がある。

格子ボルツマン法 (LBM: Lattice Boltzmann method) はステンシルに基づく計算手法であり、単純なアルゴリズムで局所的なメモリアクセスを行う。また、演算密度においても、Navier-Stokes 方程式を解く差分法と比較して高いため、GPU を用いた大規模並列計算に適している。LBM を用いた大規模計算の例として、SC10 のゴードン・ベル賞を受賞した計算があげられる。しかしながら、通常の LBM では高いレイノルズ数の計算は不可能であり、都市の気流計算等のレイノルズ数が 100 万程度の大規模乱流解析は行われたことがない。

本研究では、モデル定数を局所的に決定できるコヒーレント構造スマゴリンスキー・モデルを LBM に導入することに成功し、大規模な LES 計算を可能とした。また、本計算では並列計算の大きなオーバーヘッドとなる GPU 間の通信と分割領域内の計算をオーバーラップさせる最適化手法を導入することで、計算の高速化を図った。本計算の演算性能として、TSUBAME のほとんど全ノードの 3,968 台の GPU を用いた計算で 1.142 PFLOPS (1 秒間に 2.4×10^{12} 格子点の更新が可能) の非常に高い演算性能を達成した。また、大規模計算の解析例として、「自動車周りの流れのラージエディ・シミュレーション」(図 1)、「格子ボルツマン法を用いた東京都心部 10km 四方の 1m 解像度気流計算」(図 2) を実施した。図 2 の東京都心部の気流計算においては、東京都心部の 10 km 四方のエリアに対し、実際の建造物のデータを元に 1 m 間隔の格子を用いた $10,080 \times 10,240 \times 512$ 格子の超大規模乱流計算を実施することで、高層ビル背後の発達した渦によるビル風や幹線道路に沿って流れる「風の道」等を再現した。

以上より、GPU に適したアルゴリズムである格子ボルツマン法によるラージエディ・シミュレーションを実施することで、非常に大規模解析が可能であることを示した。本発表の最後には、今後のエクサスケールに適した計算アルゴリズムの提案や、気液二相流等の大規模解析が困難な問題に対しての研究の方針について述べる予定である。

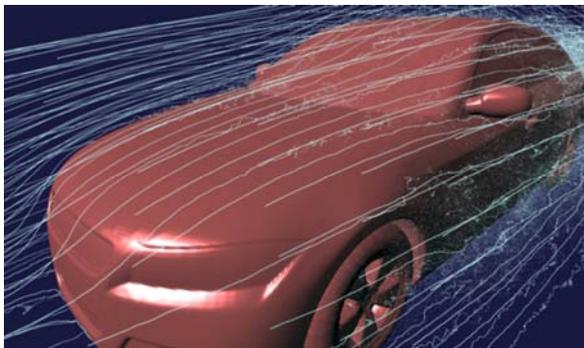


Figure 1. Snapshot of turbulent flow around a car body with $3,072 \times 1,536 \times 768$ mesh.



Figure 2. Snapshot of wind flow using particles with a $10,080 \times 10,240 \times 512$ mesh on a $10 \text{ km} \times 10 \text{ km}$ urban area of Tokyo.

2015年2月26日(木)
第26回CCSEワークショップ

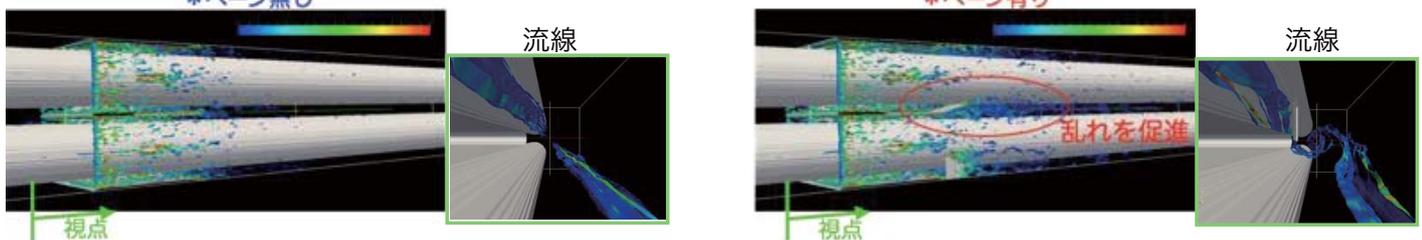
「TSUBAME」を用いた
格子ボルツマン法による大規模ラージエディ・シミュレーション

小野寺 直幸、青木 尊之、下川辺 隆史
東京工業大学 学術国際情報センター

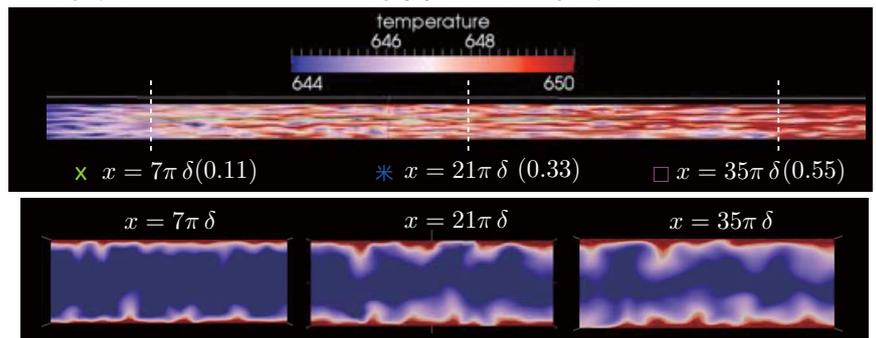
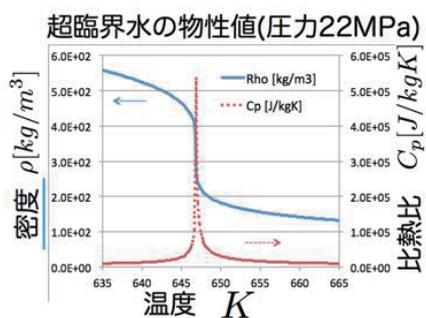


原子力機構(2011-2012)での研究内容

- 燃料棒周りの構造物の最適化と乱流解析

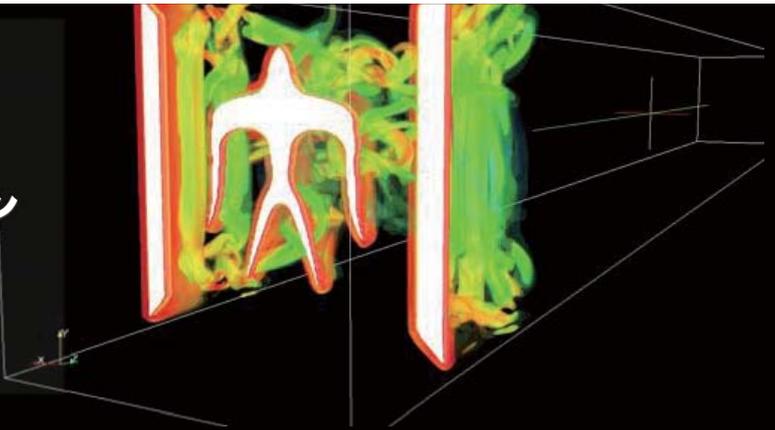


- 超臨界圧流体の伝熱劣化現象の解明のためのLES解析手法の開発

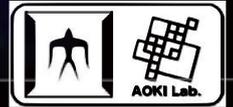
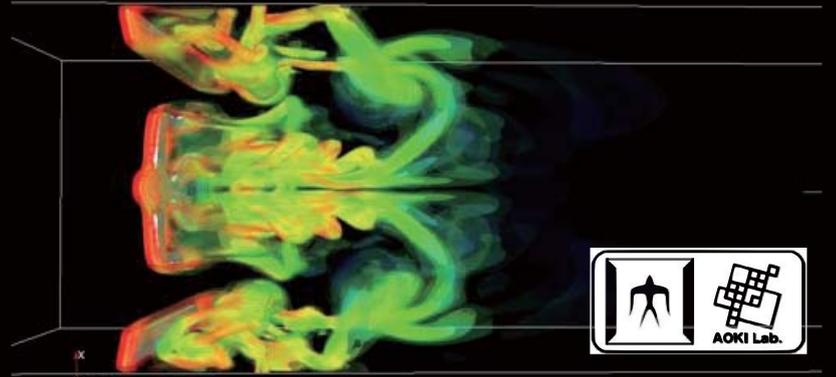


・ 研究テーマ

- ステンシル計算に基づく流体解析
- 乱流のラージエディ・シミュレーション
- 気液二相流解析
- GPUを用いた大規模計算



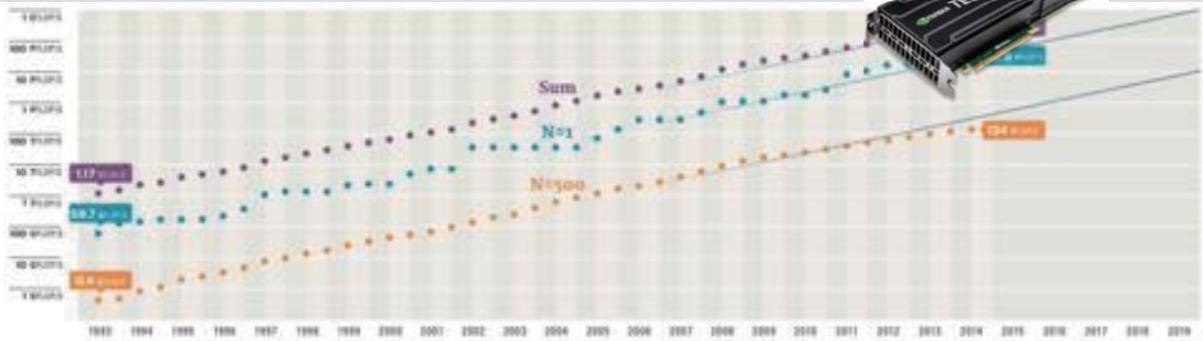
GPUを用いた流体計算
 2009年～
 GTX285(2GBMem)
 1,000,000～
 (160x160x256, 4 GPU)



FIND OUT MORE AT
www.top500.org

Rank	NAME	SPECS	SIZE	COUNTRY	CORES	Peak FLOPS	POWER
1	Tianhe-2 (Milkyway-2)	NVIDIA, Intel Ivy Bridge (12C, 2.2 GHz) & Xeon Phi (57C, 1.1 GHz), Custom interconnect	NCC Guangzhou	China	3,120,000	33.9	128

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
15	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.93GHz, Infiniband QDR, <u>NVIDIA K20x</u> NEC/HP	76032	2785.0	5735.7	1399

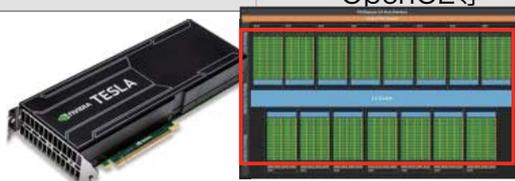


<http://www.top500.org/lists/2014/06/>

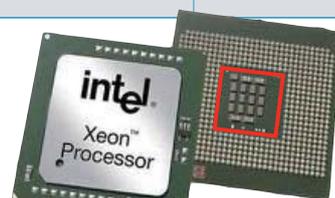
GPUとCPUの比較

NVIDIA Tesla K20X	
演算性能	1.31 TFlops (倍精度) 3.95 TFlops (単精度)
メモリバンド幅	250 GB/sec
メモリ量	6 GB (GDDR5)
CUDA コア数	2688 個
プログラミング言語	CUDA、OpenACC、OpenCL等

Intel Xeon E5-2697 v2	
演算性能	0.52 TFlops (倍精度)
メモリバンド幅	59.7 GB/sec
メモリ量	128 GB (DDR3)
コア数	2 個 (24コア)
プログラミング言語	C、C++、Fortran等



CUDAコア



CPUコア

流体の数値計算手法

- 格子ボルツマン法
有限個の方向を持つ
流体現象を再現する

$$f_i(x + c_i \Delta t, t + \Delta t)$$

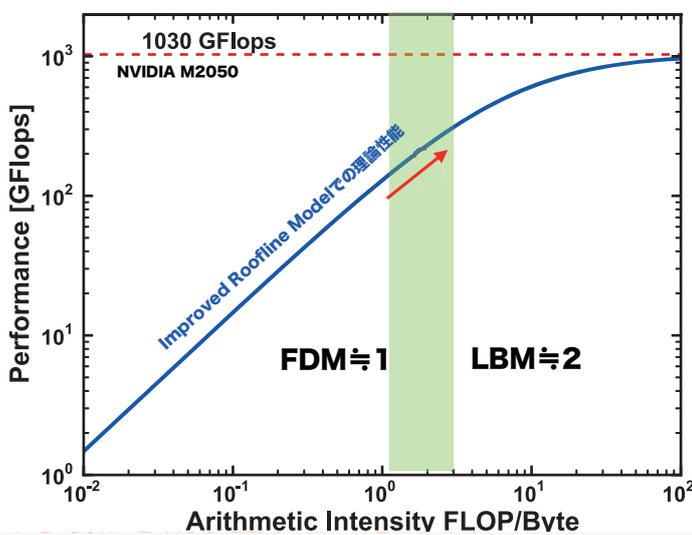
⇒方程式がシンプル

- Navier-Stokes方
流体の速度・圧力・

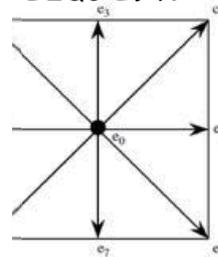
$$\bar{u}_i^* = \bar{u}_i^n + \left(-\frac{\partial \bar{u}_i^n}{\partial x} \right)$$

$$\frac{\partial^2 \bar{p}^{n+1}}{\partial x_i^2} = \frac{\partial \bar{u}_i^*}{\partial x_i}$$

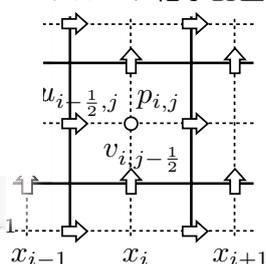
$$\bar{u}_i^{n+1} = \bar{u}_i^* - \frac{\partial \bar{p}^{n+1}}{\partial x_i} \Delta t$$



D2Q9モデル

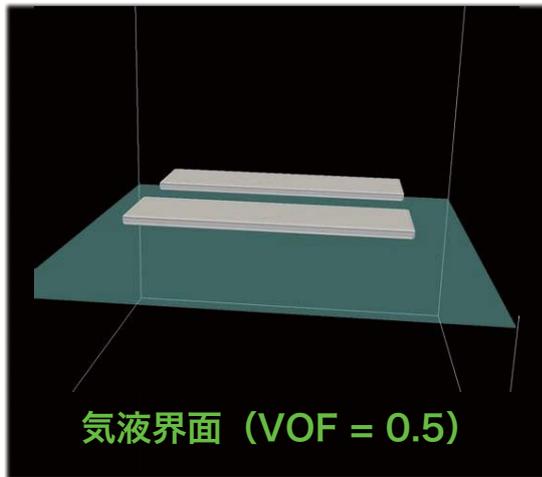


ツガード格子配置

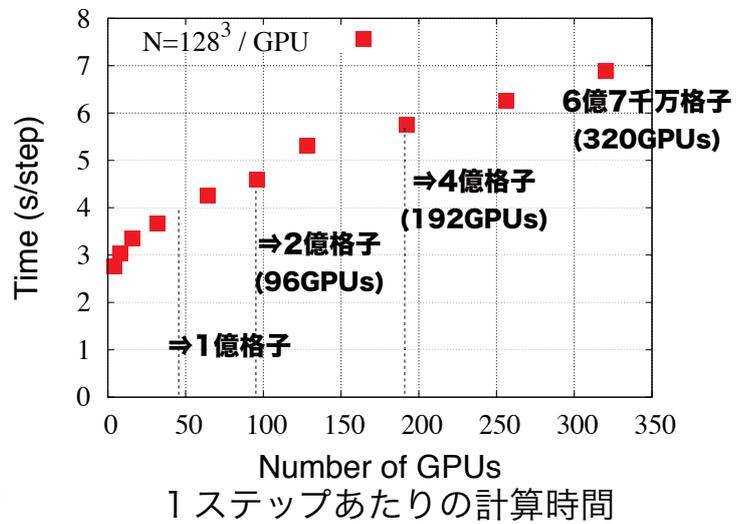


⇒方程式が複雑
データのRead/W Rooflineモデルを用いた理論性能

行列の反復計算を含む問題での計算速度の悪化



気液界面の攪拌計算(密度比 1:1000)

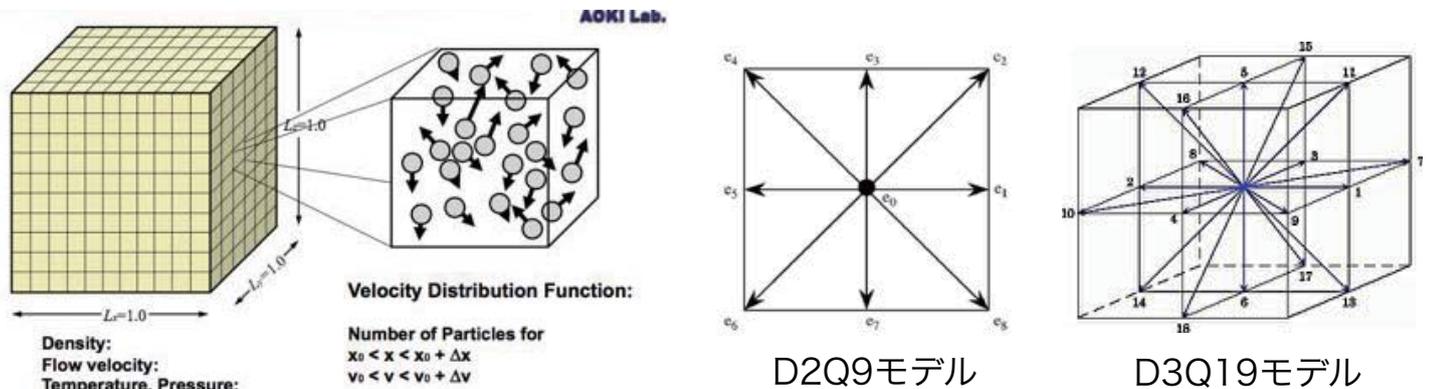


問題サイズが大きくなるに従って、Poisson方程式の反復解法の収束性が悪化し、計算速度が低下

格子ボルツマン法による流体解析

・ 格子ボルツマン法とは、

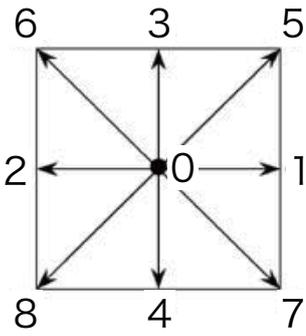
格子ボルツマン法 (こうしボルツマンほう) とは、**流体を多数の仮想粒子の集合体 (格子気体モデル) で近似し**、各粒子の衝突と並進とを粒子の速度分布関数を用いて逐次発展させることにより、その**モーメント**を計算することによって、流体の熱流動場を求めるシミュレーション法のことである。(Wikipediaより抜粋)



東京工業大学 青木先生の講義資料より抜粋

格子ボルツマン法による流体解析

- ・ 格子ボルツマン法の特徴
- ・ 各仮想粒子(f_i)がそれぞれの速度方向(c_i)を持つ
- ・ 質量・運動量・エネルギーの保存則が成り立つ

 <p>D2Q9モデル</p>	速度方向ベクトル $c_0 = (0, 0)$ $c_1 = (c, 0)$ $c_2 = (-c, 0)$ $c_3 = (0, c)$: : :	質量 $\rho = \sum_i f_i$
	運動量 $\rho \vec{u} = \sum_i c_i f_i$	
	エネルギー $\frac{1}{2} \rho \vec{u}^2 + \rho e = \sum_i \frac{1}{2} (c_i f_i)^2$	

格子ボルツマン法による流体解析

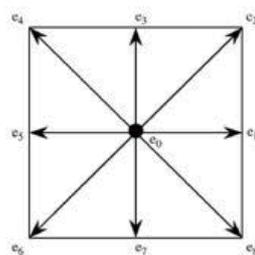
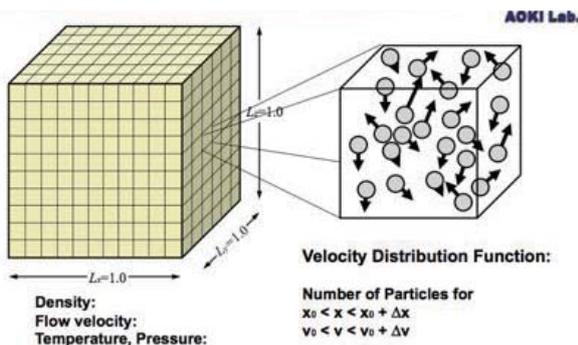
- ・ ボルツマン方程式
- ・ 有限個の方向を持つ速度分布関数の時間発展方程式を計算

$$f_i(x + c_i \Delta t, t + \Delta t) = f_i(x, t) - \frac{1}{\tau} (f_i(x, t) - f_i^{eq}(x, t)) + F_i$$

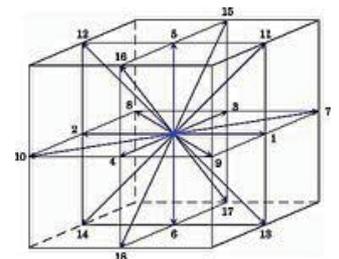
streaming (並進移動)

collision (衝突・減衰)

forcing term (外力)



D2Q9モデル



D3Q19モデル

乱流の計算手法の分類

モデル	特徴	計算負荷	解析精度・解像度
DNS	すべての変動を計算	大	高い ・非常に多くの格子点数が必要
LES	格子解像度以下の成分をモデル化	中	中～高い ・格子解像度を増やすことでDNSに近づく
RANS	変動成分をモデル化	小	低い ・幅広い現象が計算可能である反面、精度が低い ・モデルの影響が大きい

ラーゼディ・シミュレーション

・ フィルター化されたNavier-Stokes方程式

物理量を格子で解像できる成分(GS)と格子解像度以下の成分(SGS)に分割し、SGS成分をモデル化

$$u = \bar{u} + u'$$

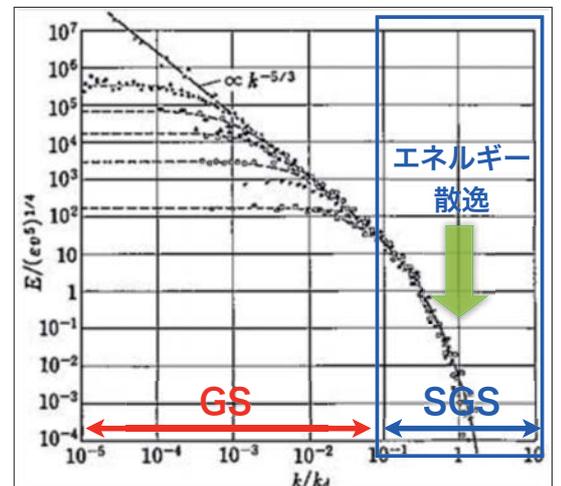
$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \tau_{ij}}{\partial x_j}$$

・ 渦粘性モデルによるSGS応力

SGSのエネルギー散逸を分子粘性と同様にモデル化

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j = -2\nu_{SGS} S_{ij}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right)$$



コルモゴロフの相似則

格子ボルツマン法によるラージエディ・シミュレーション

- Boltzmann 方程式

$$f_i(x + c_i \Delta t, t + \Delta t) = f_i(x, t) - \frac{1}{\tau} (f_i(x, t) - f_i^{eq}(x, t)) + F_i$$

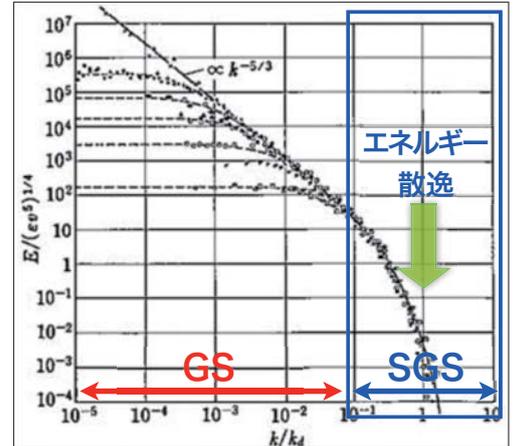
collision (衝突・減衰) + SGSによる渦粘性

- サブグリッドスケールモデルの適用

渦粘性モデルでは、SGSの渦粘性を分子粘性に加える

$$\tau = \frac{3\nu_*}{c^2 \Delta t} + \frac{1}{2} \quad (\text{LBMの緩和時間})$$

$$\nu_* = \nu_0 + \nu_{SGS} \quad (\text{粘性} = \text{分子粘性} + \text{渦粘性})$$



コルモゴロフの相似則

ラージエディ・シミュレーションの乱流モデル

Smagorinsky モデル、Dynamic Smagorinsky モデル

- Smagorinsky モデル

幅広く使われる代表的なSGSモデル。モデル係数の値を一定値とするため、経験が必要

$$\tau_{ij} = 2\nu_{SGS} S_{ij}$$

$$\nu_{SGS} = C \Delta^2 |S| \quad \boxed{C : \text{定数}}$$

○ 計算がシンプル

△ モデル係数が一定(定数値)のため、壁面近傍などで精度が低下

△ モデル係数の決定に経験則が必要

- Dynamic Smagorinsky モデル

物理量に二種類のフィルターをかけることで、動的にモデル係数が決定可能

$$\nu_{SGS} = C \Delta^2 |S|$$

$$C = \frac{\langle L_{ij} L_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle} \quad L_{ij} = \widehat{\bar{u}_i \bar{u}_j} - \widehat{\bar{u}_i} \widehat{\bar{u}_j} \quad M_{ij} = 2\widehat{\Delta^2} |\widehat{S}| \widehat{S}_{ij} - 2\widehat{\Delta^2} |\widehat{S}| \widehat{S}_{ij}$$

< > : 計算領域全体での平均操作

○ 壁などの物体に対しても適用可能

△ 計算が煩雑

△ 全計算領域での平均化が必要

→ 複雑物体周りの流れに適用できない

→ 大規模計算に不向き

ラーゼディ・シミュレーションの乱流モデル Coherent-structure Smagorinsky モデル

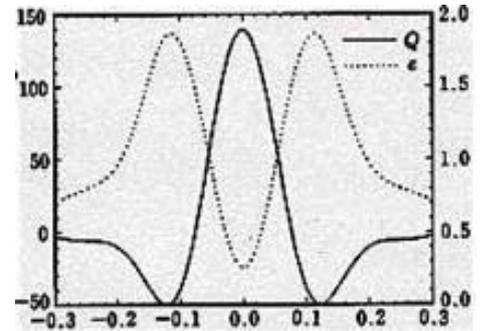
・ Coherent-structure Smagorinsky モデル

速度勾配テンソルの第二不変量を局所的に求めることで、渦粘性係数を決定

$$\tau_{ij} = 2\nu_{SGS} S_{ij}$$

$$\nu_{SGS} = C\Delta^2 |S|$$

$$C = C_1 |F_{CS}|^{3/2} \quad F_{CS} = \frac{Q}{E}$$



速度勾配テンソルの第二不変量(Q)とエネルギー散逸(ε)

- 渦粘性係数を局所的に決定可能
- 複雑物体周りの流れに適用可能
- 局所的なメモリアクセスのため大規模計算に適している

格子ボルツマン法での物体境界面の表現手法

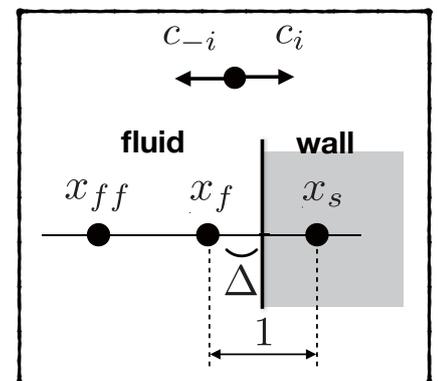
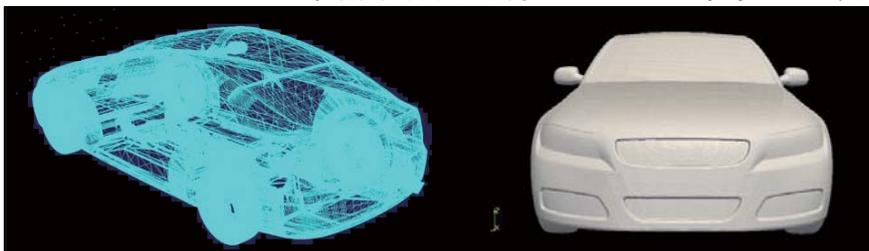
・ Interpolated bounce-back法

壁面に鏡面反射条件を与え、速度分布関数を逆方向に跳ね返す

$$f_{-i}^*(r_w, t) = 2\Delta f_i^*(r_f, t) + (1 - 2\Delta) f_i^*(r_{ff}, t) + F_{-i} \quad \Delta < \frac{1}{2}$$

$$f_{-i}^*(r_w, t) = \frac{1}{2\Delta} f_i^*(r_f, t) + \frac{(2\Delta - 1)}{2\Delta} f_{-i}^*(r_f, t) + \frac{1}{2\Delta} F_{-i} \quad \Delta \geq \frac{1}{2}$$

STLデータから距離関数を作成し、壁面を表現



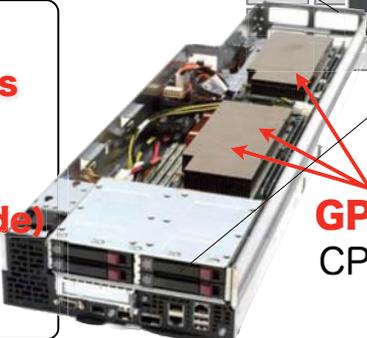
into curved and moving boundary treatments in the
ce Boltzmann method for moving boundaries, (2003)

TSUBAME 2.0 の概要

System (58racks)



Rack (30nodes)

Compute Node
GPU×3
CPU×2

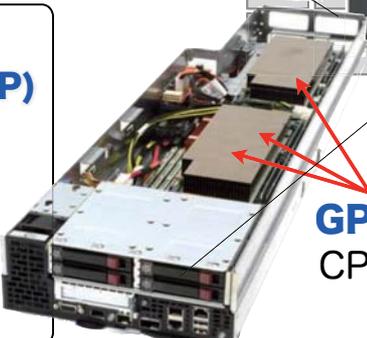
TSUBAME 2.0
Peak performance **2.4 PFlops**
GPU :
NVIDIA Tesla M2050 x 4224
(1408Node 3GPU/Node)
Network :
QDR InfiniBand x 2 (80Gbps)

TSUBAME 2.5 の概要

System (58racks)



Rack (30nodes)

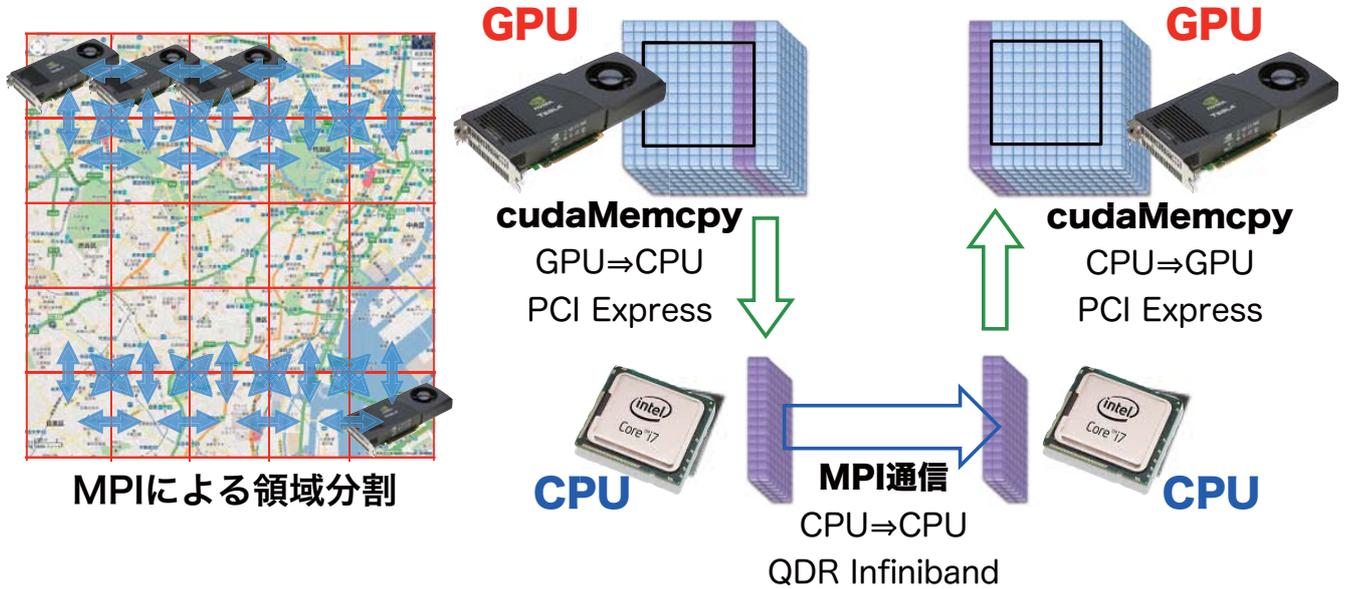
Compute Node
GPU×3
CPU×2



TSUBAME 2.5
Peak performance **17 PFlops(SP)**
GPU :
NVIDIA Tesla K20X (6GB)
(1408Node 3GPU/Node)
Network :
QDR InfiniBand x 2 (80Gbps)

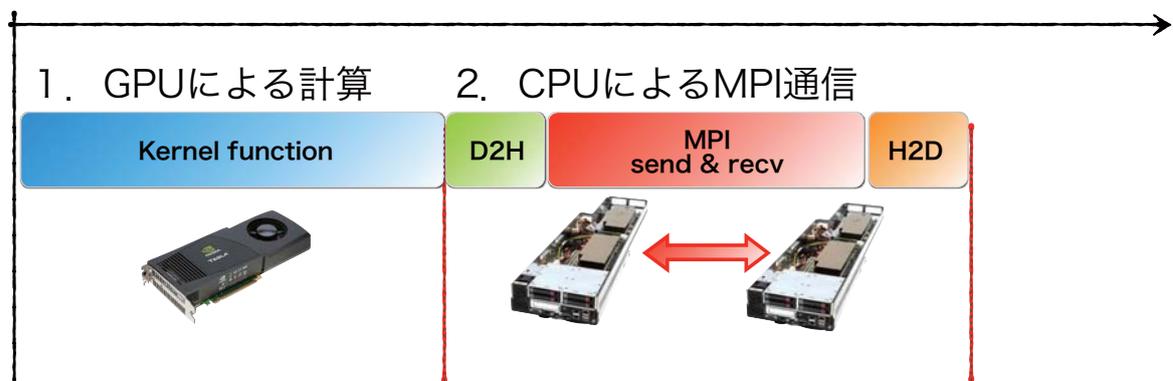
GPUによる並列計算

- ・ MPIライブラリを用いて、各計算領域の端部分を隣接ランクと交換



大規模並列計算での通信のオーバーヘッド

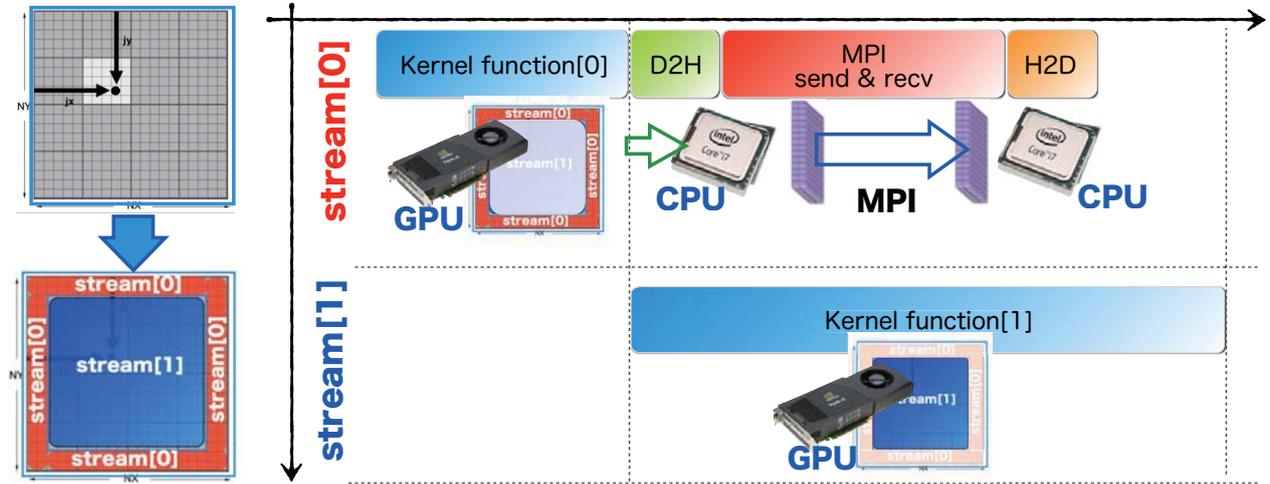
- ・ GPUによる並列計算での計算プロセスの流れ



ノード内の計算速度が速いGPUでは、通信時間の割合が大きくなる
MPIによるデータ転送の時間がオーバーヘッド

オーバーラップ計算による通信の隠蔽

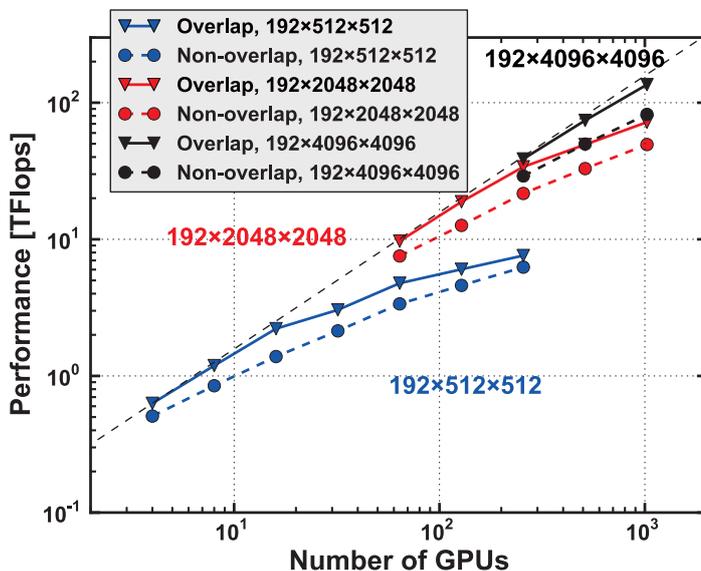
- GPUのカーネル計算とCPUを用いたMPIの通信をオーバーラップ



MPIの通信時間の隠蔽が可能となり、良いスケーリングが期待

Strong scalability on TSUBAME 2.0

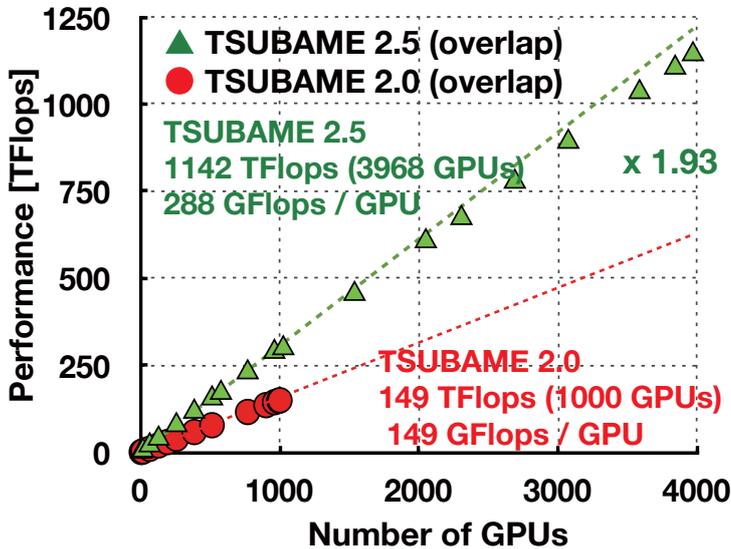
- 全計算量域の格子点数を固定：192x512x512, 192x2048x2048, 192x4096x4096



- オーバーラップ手法の導入により、30%の速度向上
- 良い強スケーリングが得られたことにより、より高速な解析が可能

Weak scalability on TSUBAME 2.0 and 2.5

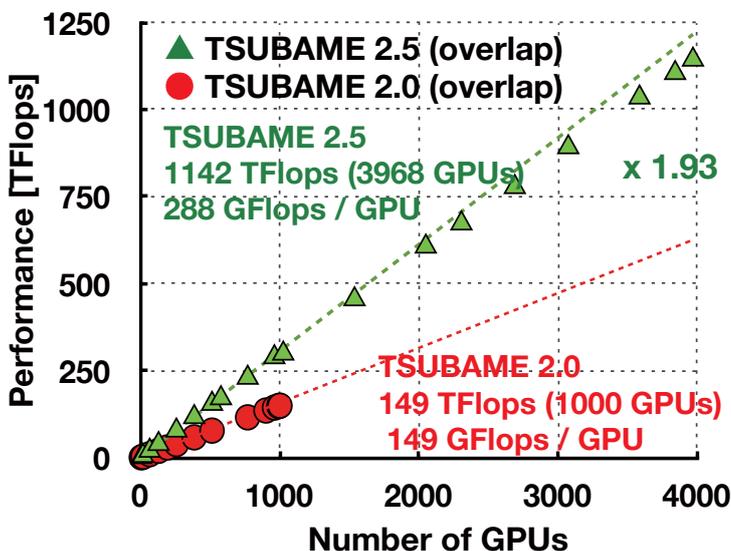
・ プロセスあたりの格子点数を固定：196×256×256 / GPU



- ・ TSUBAMEのアップデートにより、GPUあたり 1.93倍の高速化
- ・ 3968 GPUを用いた計算で、1.14 PFlops(単精度)の非常に高い実効性能を達成

Weak scalability on TSUBAME 2.0 and 2.5

・ プロセスあたりの格子点数を固定：196×256×256 / GPU

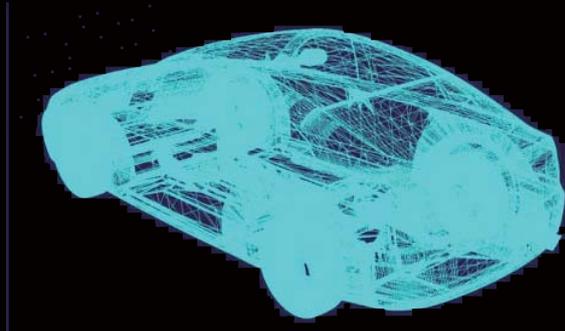


- ・ 1GPUあたり 288 GFlops の実行性能→1秒間で 605×10^6 格子点(約6億格子： 845^3)の更新が可能
- ・ 大規模な問題に対しても、非常に高速に計算を行うことが可能

車体周りの流体計算 物体表面データ形式

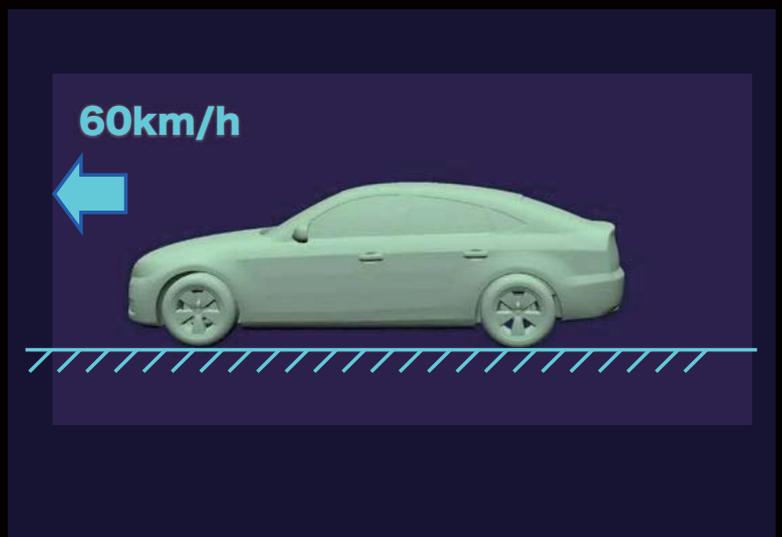
- STLデータと距離を用いた物体表現

STLデータ(三角形の頂点と法線ベクトルからなるデータ)から、物体表面の距離関数を作成



車体周りの流体計算 計算条件

格子点数	3,072×1,536×768
格子解像度	4.2 mm
計算領域	13m × 6.5m × 3.25m
GPU数	288 台
計算時間	約20時間(20万ステップ)

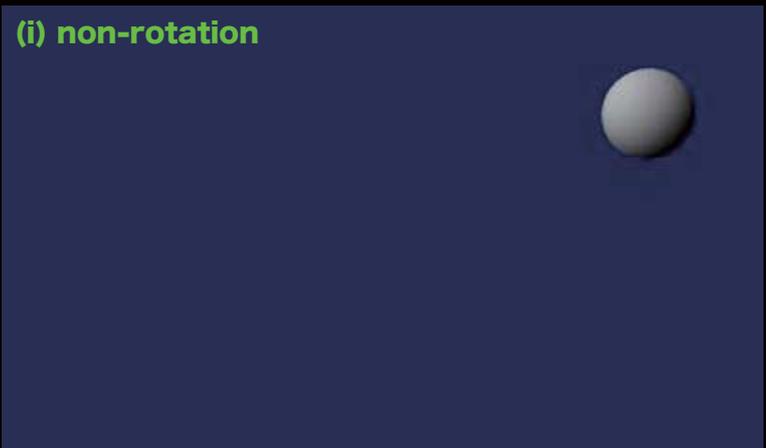




ピンポン球の飛翔計算

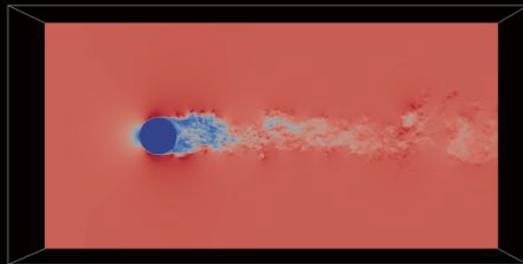
計算条件

格子点数	1,152×576×576
格子解像度	0.4 mm
速度	20 m/s
回転の種類	(i) 無回転
	(ii) 横軸回転

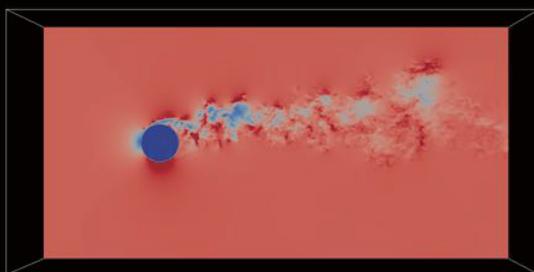


ピンポン球周りのラージエディ・シミュレーション(物体座標固定)

(i) non-rotation

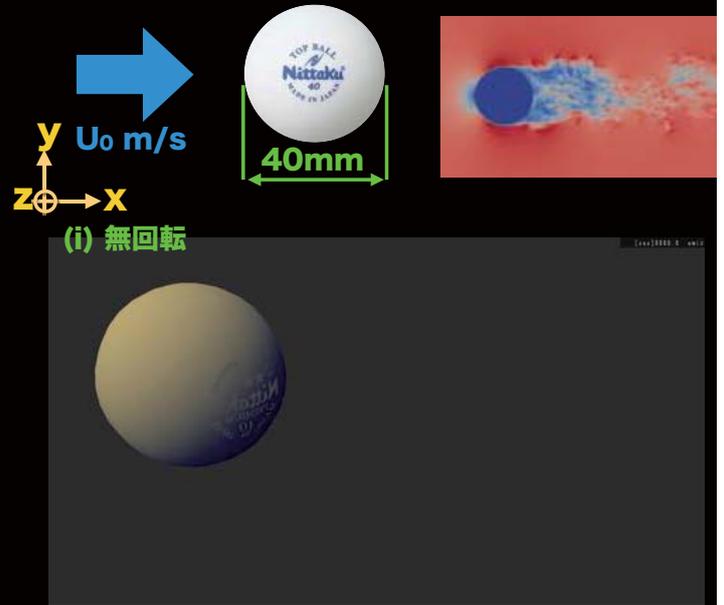
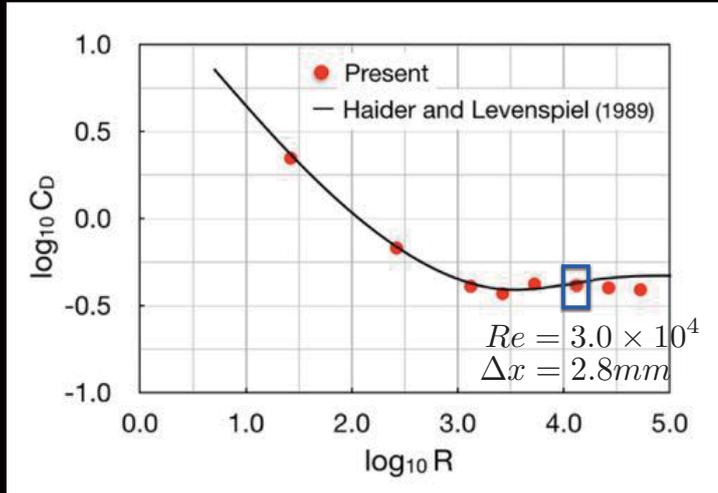


(ii) z-axis rotation



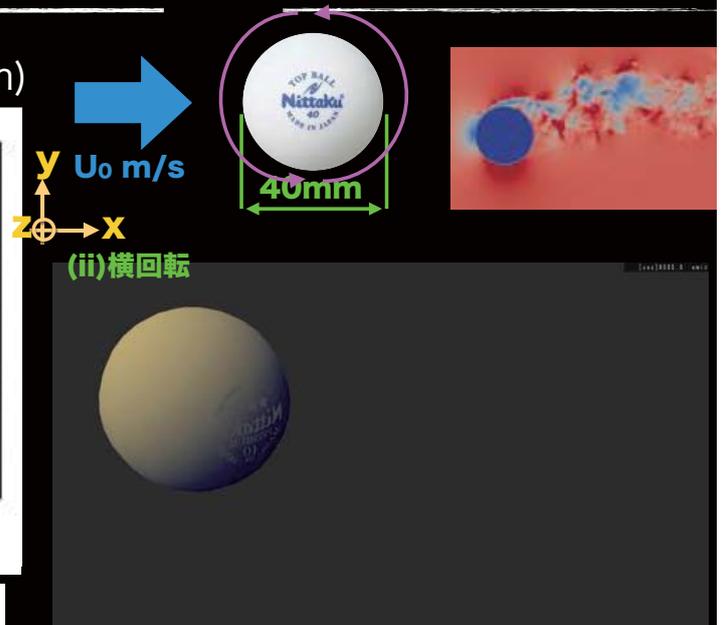
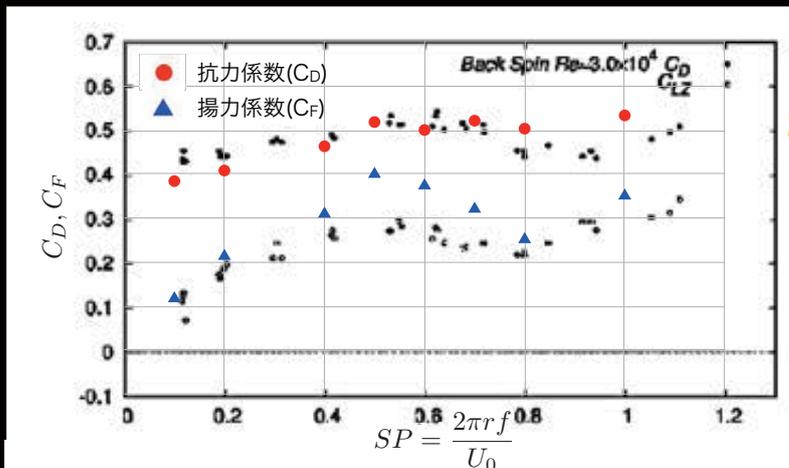
ピンポン球の空力計算 無回転時の抗力係数

球の抗力係数



ピンポン球の空力計算 回転時の抗力・揚力係数

回転時のSP依存性 ($Re = 2 \times 10^4, \Delta x = 2.7mm$)



ながれ33 (2014) 37-45
卓球ボールの空力特性

*電気通信大学大学院 田中 晃 平[†]
電気通信大学大学院 福重 貴之^{††}
電気通信大学大学院 宮崎 武
**理化学研究所 姫野 龍太郎^{†††}

東京都心部の大規模気流計算

東京都心部の大規模気流計算

計算条件

計算領域サイズ	10km × 10km × 500m
格子解像度	1 m立方格子
格子点数	528 億格子 10,080×10,240×512
GPUあたりの格子点 とデータサイズ	160×160×512 52 MB/変数 (単精度)
GPU数	4032 台
境界条件	上空100mで10m/sの北風



東京都心部の大規模気流計算 計算データサイズ

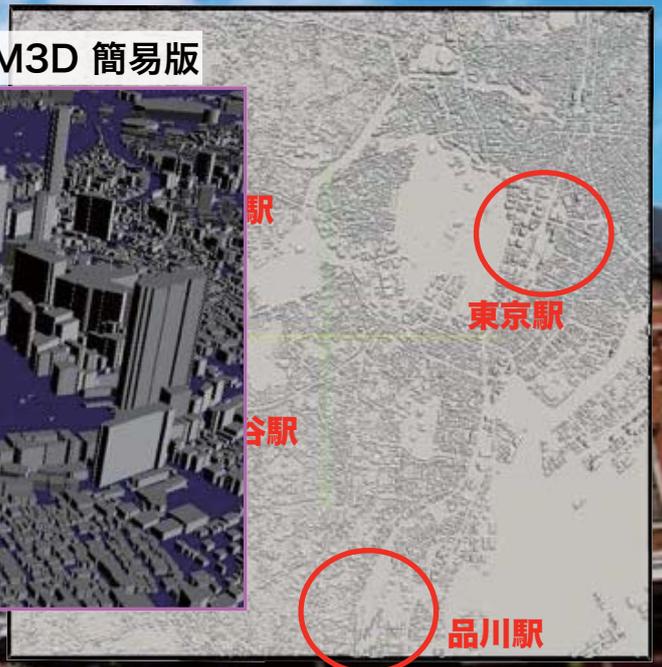
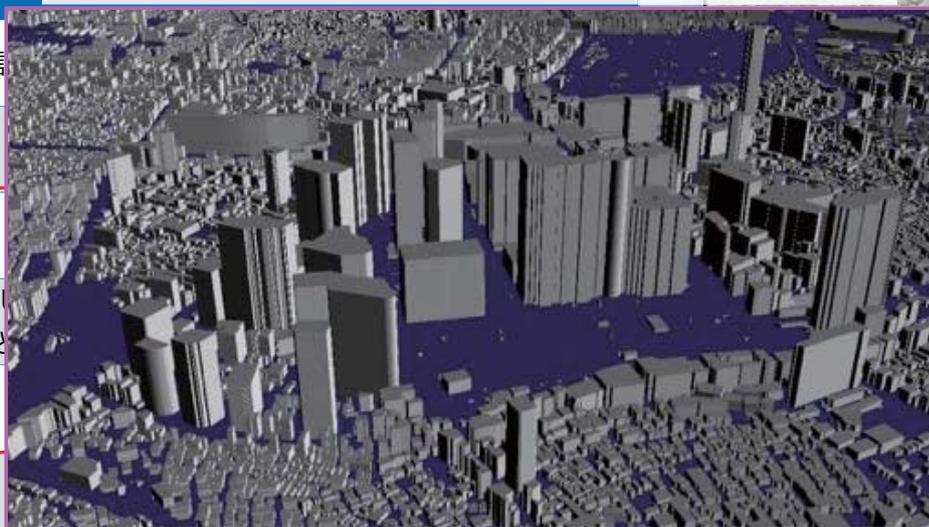
計算条件	
計算領域サイズ	10km × 10km × 500m
格子解像度	1 m立方格子
格子点数	528 億格子 10,080×10,240×512
GPUあたりの格子点 とデータサイズ	160×160×512 52 MB/変数 (単精度)
GPU数	4032 台



データサイズが4032並列では211GB/変数と非常に大きいため、各計算ノード内のローカルSSDを使用した並列I/O処理が必要

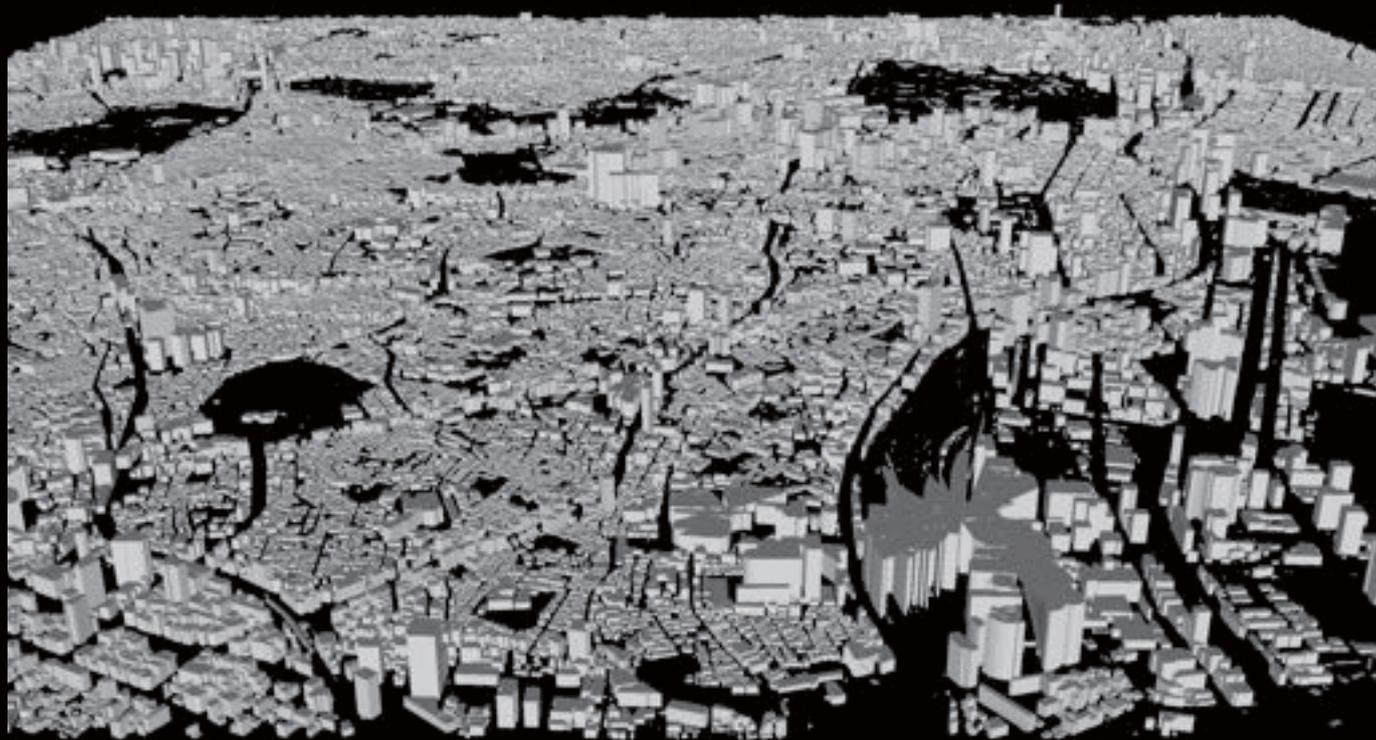
東京都心部の大規模気流計算 建物データ

読み込んだ建物データ (株)パスコ TDM3D 簡易版

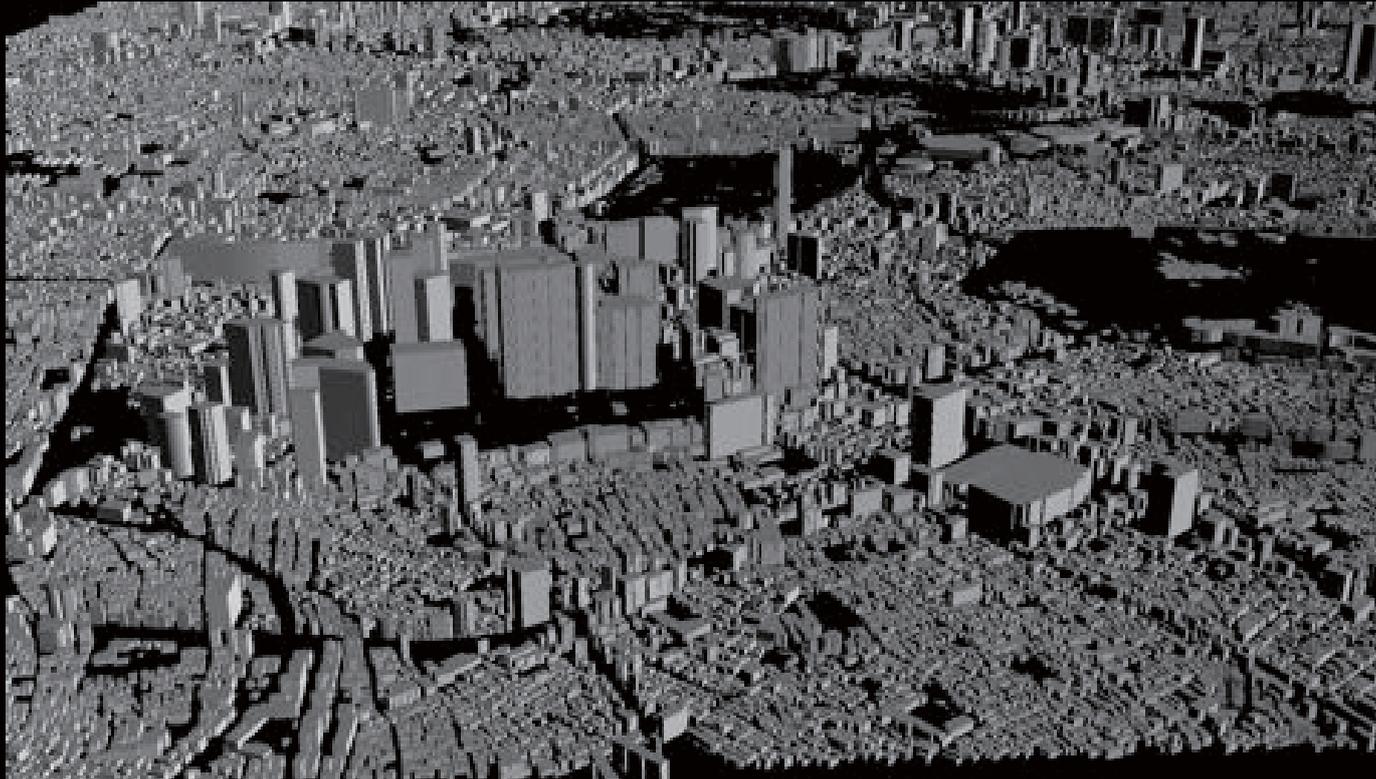


GPU
と

東京都心部1m解像度10km四方気流計算 10,080x10,240x512(4,032 GPUs)



東京都心部1m解像度10km四方気流計算 10,080x10,240x512(4,032 GPUs)



東京都庁前の速度分布 高さ100m

風向き(北風)

640 m

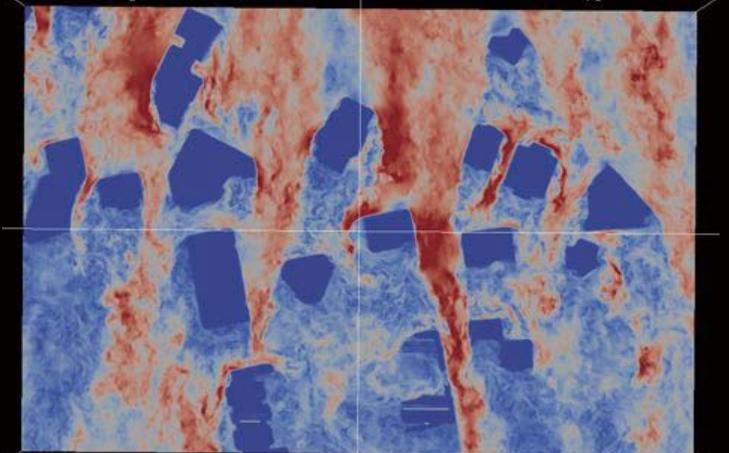


960 m

velocity Magnitude

2.5 5 7.5

0 10



東京都庁前の速度分布 高さ25m

風向き(北風)

640 m

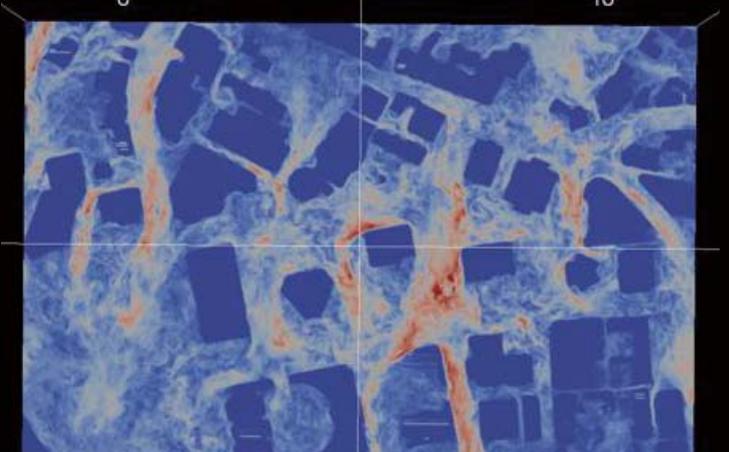


960 m

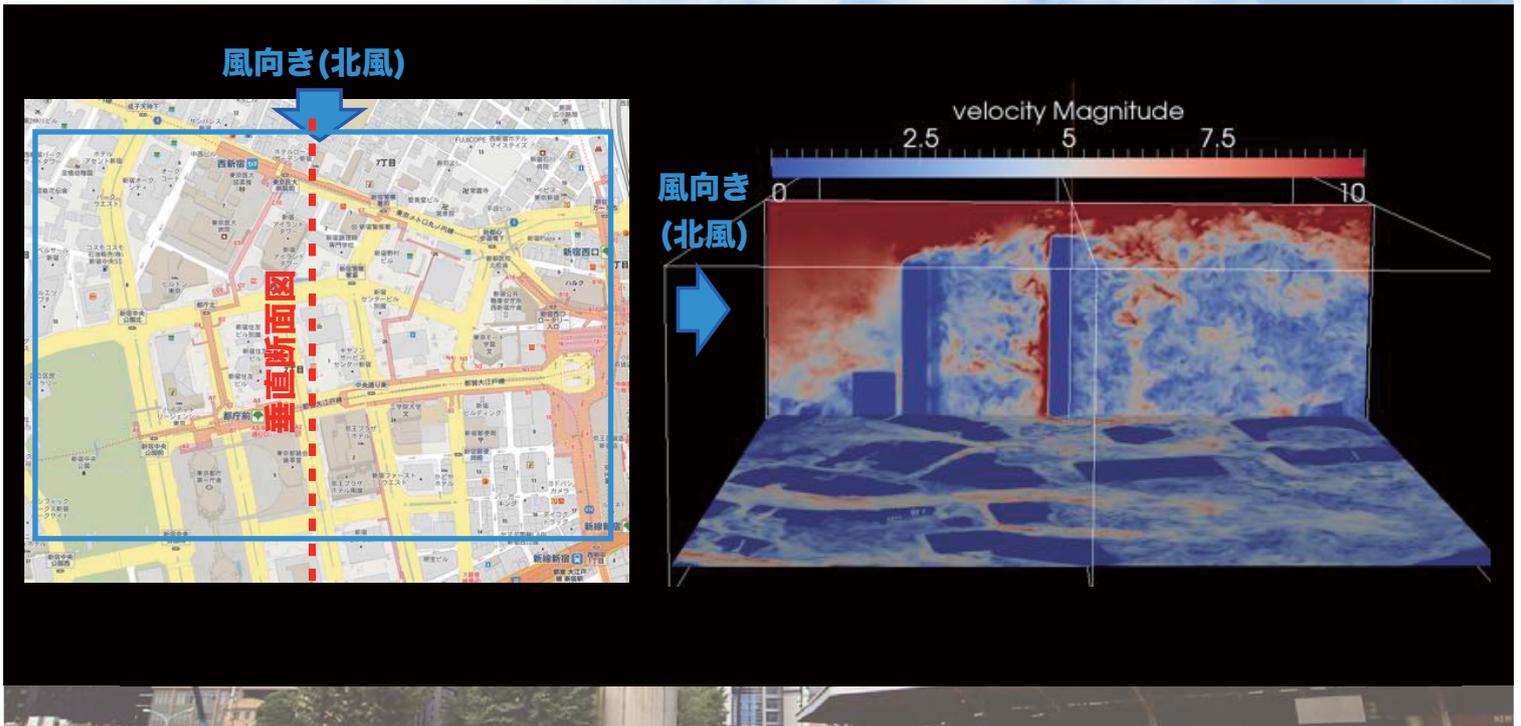
velocity Magnitude

2.5 5 7.5

0 10



東京都庁前の速度分布 垂直断面



まとめと今後の課題

・ まとめ

- ・ TSUBAME 2.5の3968台のGPUを用いた計算で、1.19 PFlops (1秒間に 2.4×10^{12} 格子点の更新が可能)と非常に高い演算性能を達成した。
- ・ 格子ボルツマン法によるラージエディ・シミュレーションを実施し、車体周りの空力解析やピンポン球の飛翔計算など、様々な乱流現象に対しても適用が可能であることを示した。
- ・ 東京都心部 1m解像度の $10,080 \times 10,240 \times 512$ 格子の大規模気流計算に成功した。その結果、都心全域の気流の流れやビル風など局所的な現象を捉えたマルチスケールの解析が可能であることを示した。

・ 今後の課題

- ・ 壁面近傍の乱流境界層や計算対象全体に対して、格子解像度を最適に変化させることが可能な細分化格子 (adaptive mesh refinement)法の適用
- ・ GPU等の、メモリバンド幅と比較して演算性能が高いアーキテクチャに適したアルゴリズムの開発
- ・ 現在の格子ボルツマン法では解析が困難な密度変化が大きな二相流問題に対しての、超大規模計算手法の開発

