

格子ボルツマン法による1m格子を用いた 都市部10km四方の大規模LES気流シミュレーション

小野寺 直幸^{1,a)} 青木 尊之^{1,b)} 下川辺 隆史^{1,c)} 小林 宏充^{2,d)}

概要: 都市は高層ビルが立ち並び複雑な構造をしており、詳細な気流を解析するためには高解像度格子による大規模気流シミュレーションが必要となる。数値計算手法は単純なアルゴリズムで大規模計算に適したD3Q19モデルの格子ボルツマン法を用いた。都市の気流はレイノルズ数が100万を超えるような乱流状態になるため、ラージエディ・シミュレーション(LES)のモデルを導入する必要がある。現在良く使われている動的スマゴリンスキー・モデルでは、モデル定数を決定するために各格子点で広領域の平均操作が必要になり、大規模計算には極めて不向きである。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・モデルを格子ボルツマン法に導入することに成功し、大規模な気流のLES計算を始めて可能にした。実際の建物データに基づき計算対象のエリアを領域分割し、TSUBAMEのFermiコアのGPUを用いて計算を行った。CUDAを用いてコードを実装し、並列計算の大きなオーバーヘッドとなるGPU間の通信を分割領域内の計算とオーバーラップさせることにより、実行性能を30%以上向上させることができた。格子ボルツマン法はメモリアクセスが律速の計算手法であるが、強スケーリング・弱スケーリングとも良い結果が得られ、1000 GPUを用いた計算ではピーク性能の15%となる149 TFLOPSの実行性能が得られた。10,080 × 10,240 × 512格子に対して4,032個のGPUを用い、新宿や皇居を含む10 km四方のエリアを1m格子で計算した。これにより、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」、台風の際の被害などが飛躍的な精度で予測できるようになる。

キーワード: GPU, 大規模計算, 格子ボルツマン法, ラージエディ・シミュレーション

A Large-scale LES Wind Simulation using Lattice Boltzmann Method for a 10km × 10km Area in Tokyo

ONODERA NAOYUKI^{1,a)} AOKI TAKAYUKI^{1,b)} SHIMOKAWABE TAKASHI^{1,c)} KOBAYASHI HIROMICHI^{2,d)}

Abstract: A lot of tall buildings and complex structures in urban large cities make the air flow turbulent. In order to understand the detail of the air flow there, it is necessary to carry out large-scale CFD (Computational Fluid Dynamics) simulations. We have developed a CFD code based on LBM (Lattice Boltzmann Method) using D3Q19 model. Since air flows in large cities are turbulent with a several-millions Reynolds number, a LES (Large-Eddy Simulation) model has to be introduced to solve the LBM equation. The dynamic Smagorinsky model is often used, however it requires to take an average operation for a wide area to determine the model constant. Since it becomes huge overhead for large-scale computations, we applied the coherent-structure Smagorinsky model which does not use any spatial averages and is able to determine the model constant locally. We have generated the computational boundary condition based on the real building data and carried out the domain decomposition for multiple-GPU computing on TSUBAME 2.0. The code is written in CUDA and the GPU kernel function is well tuned to achieve high performance on Fermi-core GPUs. By introducing the overlapping technique between the GPU-to-GPU communication and the GPU kernel computation, we have improved 30% for the large-scale computation. Although the LBM computation is essentially memory bound, we obtained fairly good performances in both the strong and the weak scalabilities. We achieved 149 TFLOPS in single precision, which is compatible with 15% of the peak performance of 1,000 GPUs. We used 4,032 GPUs for the computation with 10,080 × 10,240 × 512 mesh. By executing this large-scale computation, detailed winds behind buildings, so called "wind street" along a big street, the damage of typhoon and other will be revealed with much higher accuracy than before. The LES computation for the area 10km × 10km with 1-m resolution has never been done before in the world.

Keywords: GPU, Large-Scale Computing, Lattice-Boltzmann Method, Large-Eddy Simulation

1. 緒言

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 1m 間隔の格子解像度で詳細な気流シミュレーションを行う。都市部は高層ビルが密集した複雑な形状をしているため、それらに対する気流（風）は乱流状態となり、汚染物質などの拡散過程に大きな影響を与える。これを精度良く予測するには、1m 間隔格子を用いて広範囲に計算する必要があり、大規模計算が不可欠である。

従来の非圧縮性 Navier-Stokes 方程式を半陰解法的に解く数値計算手法では、圧力 Poisson 方程式に対する疎行列の反復計算が必要となり、行列サイズの増加や複雑物体による行列の悪条件化により、計算速度が低下する。格子ボルツマン法 (LBM: Lattice Boltzmann method) は速度の分布関数の時間発展方程式を陽的に解く事で効率的に計算を行うことが可能となる。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 [1] があげられ、複雑物体周りの流体計算についても広く研究されている [2][3]。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、実際の建造物データを用いたレイノルズ数が 100 万オーダーの大規模解析は行われていない。LBM に乱流のラージエディ・シミュレーション (LES: Large-eddy simulation) モデルを適用することで、高レイノルズ数の流体解析を安定に行う事が可能となる [4]。LES は、定常解を求めるレイノルズ平均モデル計算 (RANS) [5] とは異なり、乱流の非定常性に起因する渦を高精度に捉える事が可能である。現在良く使われている動的スマゴリンスキー・モデル [6][7] では、モデル定数を決定するために各格子点で広領域の平均操作が必要になり、大規模計算には極めて不向きである。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 [8] を格子ボルツマン法に導入することで大規模な気流の LES 計算を行う。

2. 格子ボルツマン法

LBM は連続体である流体を格子上を並進・衝突する仮想的な粒子の集合と仮定し、格子上の粒子の分布関数について時間発展方程式を解く手法である。LBM では空間は等間隔の格子によって離散化され、有限個の速度を持つ粒子は並進運動により 1 タイムステップ後に格子点上に位置するため、補間などによる離散化誤差を含まない。衝突過程は BGK モデル [9] を用いて、粒子分布が単一時間で局所平衡状態へと緩和すると仮定する。LBM は非圧縮性 Navier-Stokes 方程式を解く半陰解法とは異なり、圧力 Poisson 方程式に対する疎行列の反復計算を行う必要が無く、複雑物体を含む流れ場や大規模計算に適している [10][11][12]。ここでは、LBM の弱圧縮性流体モデルに対して、離散的粒子速度に 3 次元 19 速度 (D3Q19) モデルを用いた。

時刻 t 、位置 x において、速度 c_i に対応する粒子の分布関数 $f_i(x, t)$ の時間発展は以下の式で表される。

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

Δt は時間刻み、 τ は緩和時間係数、 $f_i^{eq}(x, t)$ は局所平衡状態における粒子分布、 F_i は外力である。D3Q19 モデルの粒子速度 c_i は以下のように表される。

$$c_i = \begin{cases} (0, 0, 0) & i = 0, \\ (\pm c, 0, 0), (0, \pm c, 0), (0, 0, \pm c) & i = 1-6, \\ (\pm c, \pm c, 0), (\pm c, 0, \pm c), (0, \pm c, \pm c) & i = 7-18. \end{cases} \quad (2)$$

並進過程では、粒子は隣接する格子点へと分布関数を保ったまま移動するため、隣接格子間の物理量の複写に相当する。衝突過程は隣接格子を参照しない局所的なメモリ参照になる。平衡分布関数は以下の式で表される。

$$f_i^{eq} = w_i \rho \left(1 + \frac{3c_i \cdot u}{c^2} + \frac{9(c_i \cdot u)^2}{2c^4} - \frac{3u^2}{2c^2} \right) \quad (3)$$

ρ は密度、 u は速度、 w_i は重み係数である。D3Q19 モデルの重み係数 w_i は以下のように表される。

$$w_i = \begin{cases} 1/3 & i = 0, \\ 1/18 & i = 1-6, \\ 1/36 & i = 7-18. \end{cases} \quad (4)$$

密度 ρ と運動量 ρu_i の保存は成り立ち、以下の式で表される。

$$\rho = \sum_i f_i, \quad \rho u = \sum_i c_i f_i. \quad (5)$$

Boltzmann 方程式に Chapman-Enskog 過程と密度一定の条件を与えることで、非圧縮 Navier-Stokes 方程式が導かれる。

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (6)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}. \quad (7)$$

$p = \rho/3$ は圧力、 ν は動粘性係数となる。衝突過程での緩和時間 τ は動粘性係数を用いて以下の式で表される。

$$\tau = \frac{1}{2} + \frac{3\nu}{c^2 \Delta t}. \quad (8)$$

等温モデルでは外力項は以下の式で表される。

¹ 東京工業大学 東京都目黒区大岡山 2-12-1
² 慶應義塾大学 神奈川県横浜港北区日吉 4-1-1
a) onodera@sim.gsic.titech.ac.jp
b) taoki@sim.gsic.titech.ac.jp
c) shimokawabe@sim.gsic.titech.ac.jp
d) hkobayas@phys-h.keio.ac.jp

$$F_i = -3w_i\rho \frac{c_i \cdot a_i}{c^2} \Delta t. \quad (9)$$

a_i は各離散化速度分布 (f_i) に対する外力である。

建物などの物体境界の与え方として、粒子を並進運動の出発点に跳ね返す Bounce-Back (BB) 境界条件や、補間関数を用いる事で高次精度化した Bounce Back 境界条件 [13][14]、物体力により境界を表現する Immersed Boundary Method (IBM) [15] などが用いられている。本研究では、メモリ参照の局所性が良く高速に計算を行う事ができる BB 境界条件を採用した。

3. ラージエディ・シミュレーション

LES は格子で解像できる物理量 (GS:grid scale) に対して直接計算を行い、格子解像度以下の成分 (SGS:sub-grid scale) に対してはサブグリッド・スケール (SGS) モデルによりモデル化を行う。LES は定常解を求めるレイノルズ平均モデル計算 (RANS) [5] とは異なり、乱流の非定常性に起因する渦を高精度に捉える事が可能である。流体の運動エネルギーはエネルギーカスケードにより平均的には GS 成分から SGS 成分へ輸送されるため、運動エネルギーの散逸率は負 ($\epsilon < 0$) となる。渦粘性に基づいた SGS モデルでは、SGS 変動の効果が分子粘性と同様に作用すると仮定し、渦粘性 ν_{SGS} を導入する。

$$\nu_{SGS} = C\bar{\Delta}^2 |\bar{S}| \quad (10)$$

ここで C はモデル係数、 $\bar{\Delta}$ はフィルター幅であり、ひずみ速度テンソル \bar{S}_{ij} 、およびひずみ速度テンソルの強さ $|\bar{S}|$ は、

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right), \quad |\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}. \quad (11)$$

と定義される。

3.1 スマゴリンスキー・モデル

LES で広く用いられている SGS モデルとしてスマゴリンスキー・モデル [16] がある。スマゴリンスキー・モデルではモデル係数 C_s が一定であるため、物体壁面に対してモデルが正しい振る舞いをせず、van Driest 減衰関数 [17] など、壁面効果をモデル化した関数を用いる必要がある。

3.2 動的スマゴリンスキー・モデル

スマゴリンスキー・モデルはモデル係数を定数としたため、SGS 応力が常に散逸的に働き安定に計算を行うことができる利点があるが、モデル係数の決定に経験的な知識が必要なことや、複雑物体に対して正確な渦粘性を導入できないなどの欠点がある。動的スマゴリンスキー・モデル (DSM) [6][7] はモデル係数 C_{DSM} を流れ場から動的に求めることでそれらの問題点を解決した。DSM は、フィルター化された速度場に対して、更にテストフィルターをか

ける事でモデル係数を決定する。DSM の問題点として、モデル係数の非物理的な値を回避するためのクリッピング操作や、モデル係数を決定するために各格子点で広領域の平均操作が必要になり、大規模計算には極めて不向きである。

3.3 コヒーレント構造スマゴリンスキー・モデル

コヒーレント構造スマゴリンスキー・モデル (CSM) [18] は乱流のコヒーレント構造からモデル係数 C_{CSM} を決定する手法であり、複雑物体を含む流れ場に対しても高精度に計算を行う事ができる [8]。モデル係数は速度勾配テンソルの第 2 不変量 (Q) と速度勾配テンソルの大きさ (E) から求められるコヒーレント構造関数 F_{CS} により決定される。

$$C_{CSM} = C' |F_{CS}|^{3/2}, \quad (12)$$

$$F_{CS} = \frac{Q}{E} \quad (-1 \leq F_{CS} \leq 1), \quad (13)$$

$$Q = -\frac{1}{2} \frac{\partial \bar{u}_j}{\partial x_i} \frac{\partial \bar{u}_i}{\partial x_j}, \quad E = \frac{1}{2} \left(\frac{\partial \bar{u}_j}{\partial x_i} \right)^2. \quad (14)$$

C' は係数であり、本研究では $C' = \frac{1}{20}$ を用いる。モデル係数は隣接する物理量の値より求められ、高効率な並列演算処理を行う事ができる。

3.4 格子ボルツマン法のラージエディ・シミュレーション

コヒーレント構造スマゴリンスキー・モデルを LBM 計算に適用する事で、複雑物体を含む高レイノルズ数流れの大規模計算が初めて実施可能となる。乱流モデルを適用した、LBM の緩和時間 τ_* は以下の式で表される。

$$\tau_* = \frac{1}{2} + \frac{3\nu_*}{c^2 \Delta t}. \quad (15)$$

乱流モデルを考慮した粘性 (ν_*) は、分子粘性 (ν_0) とコヒーレント構造スマゴリンスキーモデルの渦粘性 (ν_t) の和で表される。

$$\nu_* = \nu_0 + \nu_t. \quad (16)$$

ここで、ひずみ速度 $|S|$ は LBM の速度分布関数 f_i から直接求めることで、局所的なメモリ参照が可能となり、正確な渦粘性係数の計算、および高速な演算が可能となる。

$$|S| = \sqrt{2S_{\alpha\beta}S_{\alpha\beta}}, \quad (17)$$

$$S_{\alpha\beta} = \frac{-3}{2\rho\tau} \sum_i (f_i - f_i^{eq}) c_{i\alpha} c_{i\beta}. \quad (18)$$

c_{ik} は、速度 c_i の k 軸成分となる。

4. 平行平板間乱流問題による検証計算

乱流計算の検証問題として、平行平板間乱流計算を行う。計算条件として、摩擦レイノルズ数 $Re_\tau = 590$ 、計算領域 $(L_1, L_2, L_3) = (6, 2, 3)$ 、格子数 $(N_1, N_2, N_3) = (768, 256, 384), (1536, 512, 768)$ を設定した。境界条件として、 x_2 方向にすべりなし境界条件、 x_1, x_3 方向に周期境界

条件を設定し、主流の x_1 方向に圧力勾配 $\frac{\partial p}{\partial x} = 1$ を与えた。

図 1 に主流方向 (x_1) の平均速度分布を示す。横軸は粘性座標 $x_2^+ = Re_\tau x_2$ である。壁面に近い粘性低層領域 $x_2^+ < 10$ では、格子点数が荒い条件 $(N_1, N_2, N_3) = (768, 256, 384)$ と細かい条件 $(N_1, N_2, N_3) = (1536, 512, 768)$ 、共に差分法の DNS[19] の結果と同様の速度分布が得られた。バッファー領域から対数領域 ($x_2^+ > 10$) においては、格子点数が少ない結果で平均速度をわずかに過大評価するものの、良い結果となった。

図 2-図 4 に各方向の乱流強度 (速度の rms 値) を示す。格子解像度が低い結果においては、主流方向の乱流強度 u_1' を過大評価、壁面・スパン方向 u_2', u_3' の乱流強度を過小評価したものの、非常に良好な結果が得られた。格子解像度が高い結果においては、DNS の結果と完全に一致している。SGS モデルを用いない LBM では、計算が不安定となり最終的に破綻した。

以上の結果より、LBM に CSM を適用することで DNS への良い収束性が期待されると共に、従来の LBM では計算が不可能であった実在流体に見られるような高いレイノルズ数においても安定に計算を行うことが可能となる。

5. 格子ボルツマン法に対する複数 GPU を用いた並列計算の最適化

5.1 MPI 通信とカーネル関数のオーバーラップ計算

並列計算で行われるプロセス間の通信の時間を隠蔽するための技法として、計算と通信のオーバーラップ手法が提案されている [20][21]。計算領域を通信に必要な境界部分と通信には無関係な中央部分に分割し、GPU の中央部分のカーネル関数と境界部分の MPI 通信を同時に行う。境界部分は必要最小限の計算領域を確保し、ホストメモリとデバイスメモリ間のデータ通信は `cudaMemcpyAsync` 等を、ノード間のデータ通信は MPI の `Isend・Irecv` 等の非同期通信を採用した。

MPI の領域分割法として、 yz 方向に分割を行う 2 次元分割法を採用した。LBM では各方向の速度分布関数のメモリ参照が非対称となるため、物理量が必要な領域のみ MPI 通信を行い、通信量の削減を行った。具体的には、 y 方向のみの MPI 通信は速度分布関数 $f_i = (0, \pm c, 0), (\pm c, \pm c, 0)$ の 6 成分に対して、 z 方向のみの MPI 通信は速度分布関数 $f_i = (0, 0, \pm c), (\pm c, 0, \pm c)$ の 6 成分に対して、 yz 方向の MPI 通信は速度分布関数 $f_i = (0, \pm c, \pm c)$ と速度 u_i の 7 成分に対して通信を行った。

計算の順序として、境界部分のカーネル関数を実行する。その後、中央部分のカーネル関数の命令を発効した後、境界部分の MPI 通信を行うことで、中央部分のカーネル関数と通信のオーバーラップ計算を行う。

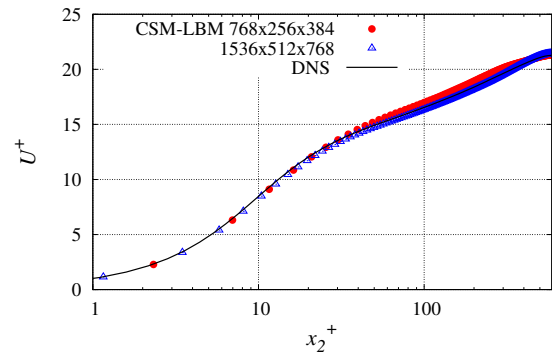


図 1 The profiles of the mean velocity.

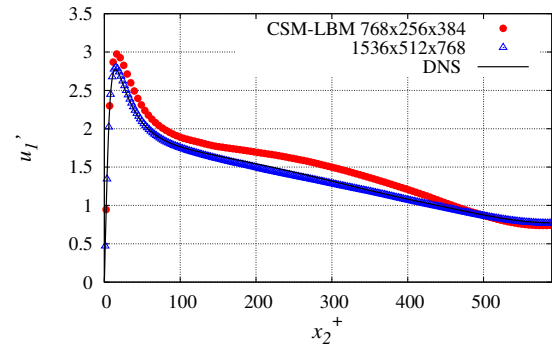


図 2 The profiles of the turbulent intensity in streamwise direction.

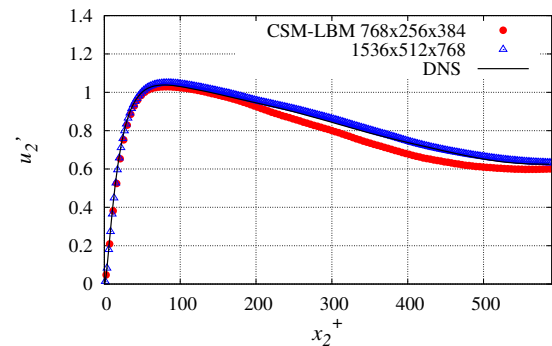


図 3 The profiles of the turbulent intensity in wall direction.

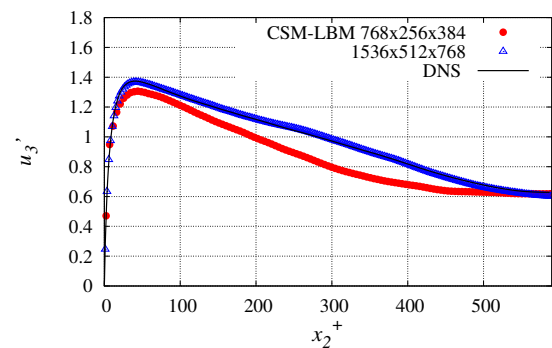


図 4 The profiles of the turbulent intensity in spanwise direction.

6. TSUBAME 2.0 での実行性能測定

東京工業大学のスパコン TSUBAME 2.0 は、4000 個以上の GPU (NVIDIA TESLA M2050) を搭載し、LINPACK ベンチマークにおいて 1.192 PFLOPS を達成している [21]。TSUBAME 2.0 の複数 GPU を用いた LBM ラージエディ・

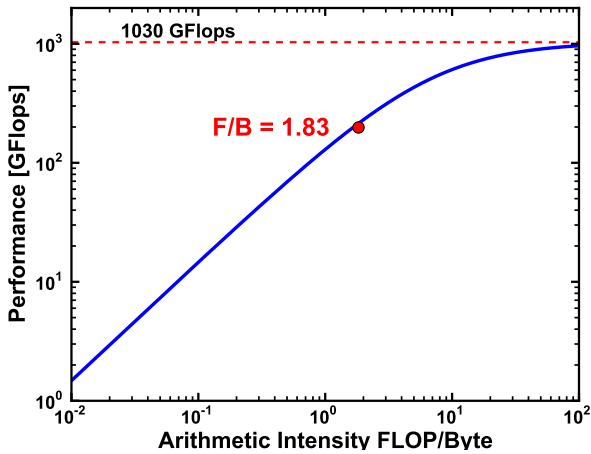


図 5 Performance of our single-GPU code with improved roofline model.

シミュレーションの実行性能測定を行う。

6.1 単体 GPU での実行性能

単体 GPU を用いた実行性能の測定を行った。高速化手法として、SFU (special function unit) を用いた演算の高速化、32bit 版のコンパイルによる配列の index 計算量の削減、LBM の並進・衝突過程のカーネル関数の結合 [2] を行った。

図 5 に最適化を行った計算の単精度演算性能、および理論性能の Improved Roofline Model [20] を示す。Improved Roofline Model は、1 格子点あたりの演算量 F とデータ参照量 B 、および演算器の Peak Performance (F_{peak}) とメモリバンド幅 B_{peak} より以下のように表される。

$$P = \frac{F/B}{F/B + F_{peak}/B_{peak}} F_{peak}. \quad (19)$$

本計算では以下のパラメータを用いた。

$$(F, B) = (476FLOPS, 260Byte), \quad (20)$$

$$(F_{peak}, B_{peak}) = (1030GFLOPS, 148Gbyte/sec), \quad (21)$$

$$(P, F/B) = (214GFLOPS, 1.83). \quad (22)$$

図 5 に単精度演算性能および、Improved Roofline Model によって求められる理論性能を示す。最適化を行った LBM の単精度実行性能は 198 GFLOPS となり、32bit 版のコンパイルを適用しない結果の 183 GFLOPS に対して 8% の性能向上を達成するとともに、Improved Roofline Model により予測されるピーク性能 (P) に対して 92% と、十分な最適化がなされている事が確認できる。

6.2 複数 GPU を用いた実行性能

TSUBAME 2.0 の複数 GPU (3 GPU/Node) を用いて性能測定を行う。図 6、図 7 に実行性能の弱スケーリング

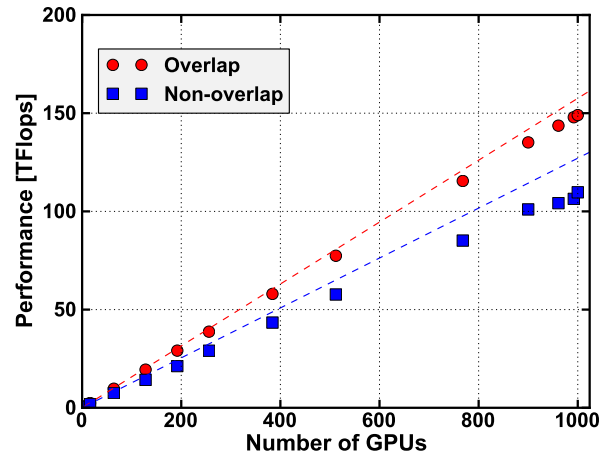


図 6 Weak scalability on TSUBAME2.0.

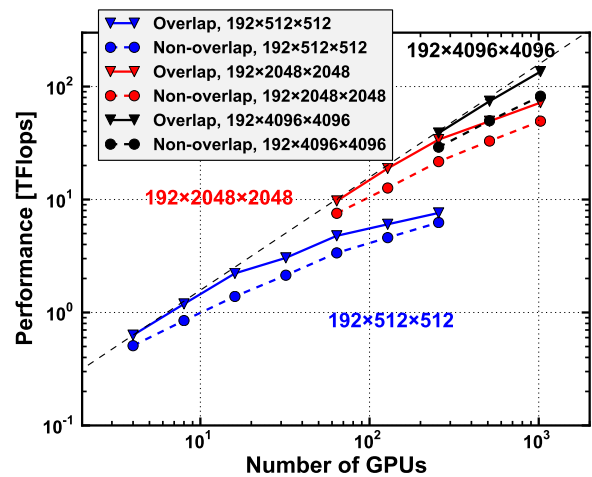


図 7 Strong scalability on TSUBAME2.0.

グ、および強スケーリングを示す。横軸は GPU 数、縦軸は単精度実行性能 (TFLOPS) となる。

弱スケーリング計算では、1 GPU あたり (N_1, N_2, N_3) = (192, 256, 256) の格子点数を確保した。オーバーラップ計算を行う事で通信の隠蔽が可能となり、オーバーラップ無しに比べて 30% 程度の性能向上、および 768 GPU で 115 TFLOP、1000 GPU で 149 TFLOPS の実行性能が得られる。強スケーリングの結果においても同様に、オーバーラップ計算を行う事により良いスケーリングが得られ、格子点数 (N_1, N_2, N_3) = (192, 2048, 2048) の 32 GPU を用いた結果に対して、2 倍の 64 GPU では 97%、4 倍の 128 GPU では 87%、8 倍の 256 GPU では 63% の効率得られ、計算時間を効果的に減らす事が可能となる。

以上の結果より、局所的なメモリアクセスが可能な乱流モデルである CSM を LBM に適用した事で、動的なマゴリンスキーモデルの広域での平均操作が不要となり、弱スケーリング、強スケーリング共に良いスケーリングが得られた。

7. 都市気流計算

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 (株) パスコ) を元に 1 m 間隔の格子解像度で気流をシミュレーションを行う。東京工業大学のスパコン TSUBAME 2.0 の 4032 個の GPU を用いて、格子解像度 $(N_1, N_2, N_3) = (10080, 10240, 512)$ を設定した。境界条件として、主流方向 (x_2 :南北方向) に流入・流出境界条件、スパン方向 (x_1 :東西方向) に周期境界条件、高さ方向 (x_3) の上面にスリップ境界条件を与えた。流入速度は対数則による速度分布関数 $U_{in} = A \log_{10} z/z_0$ から与え、相対粗度を $z_0 = 2$, 高さ 100 m において風速を 10 m/s と設定した。流入速度, 空気の物性値, および代表長さ 1m で規格化したレイノルズ数はおよそ 10^6 程度となり, 乱流モデル無しでの解析を行ったところ計算が破綻した。

図 8 に今回計算した 10 km 四方の地図データ (2012Google ZENRIN) を示す。図 9 に高さ 25 m での風速分布の瞬時値を示す。建物群の影響により、全体的に風速が遅い領域が分布している事が分かる。一方で、計算領域右下側などの海面部や左側の明治神宮、中央部の皇居など、高い建物が無い領域においては平均的な風速が高くなっており、上空の高速な風が流れ込んでいる事が確認できる。これらの地域では、木などの建築物以外の障害物を考慮した物理モデルを適用する事が課題となる。

図 10 に高さ 100m での風速分布の瞬時値を示す。乱流の特徴的なストリーク構造 (速度の速い・遅いが交互に存在する構造) が見られ、乱流拡散が活発に行われている事が再現されている。計算領域左上部の新宿副都心や計算領域中央部の六本木において、高層ビル群の影響で風速が下がっており、非常に広い領域に対して影響を与えている事が確認される。

図 11 に港区に可視化用の粒子分布を示す。ビル群の間を粒子が流れていく様子が確認できる。

図 12-図 14 に新宿都庁前を抽出した図を示す。参考までに、可視化されている領域は格子点数 $960 \times 640 \times 256$ であり、青色の速度 0 の領域は建物に相当する。図 13 に高さ 25 m の速度分布の瞬時値を示す。主要道路に沿って風速が強い領域が存在しており、特に計算領域中央部から南方に向かって強い流れ (ビル風) が再現されている。図 14 に高さ 100 m の速度分布の瞬時値を示す。高さ 100 m において、高層ビル間に風速の非常に強い領域が存在しており、高さ 25 m の風速分布に大きな影響を与えている事が確認できる。高層ビル後方には風速の弱い領域が存在しており、非常に複雑な速度分布が形成されている。

図 15 に地図データ 12 の赤点線上の垂直断面内 (高さ 256 m まで) の速度分布を示す。高層ビルにより、速度の速い風が上空へと押し出されている様子が確認できる。さ

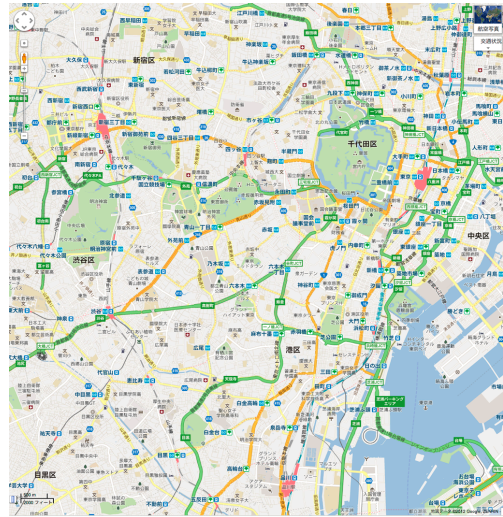


図 8 Central Tokyo area corresponding to the computation. (2012Google ZENRIN)

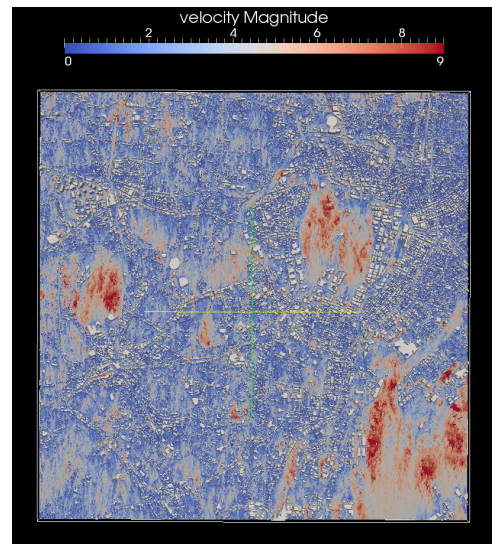


図 9 Wind velocity profile at height of 25 meters.

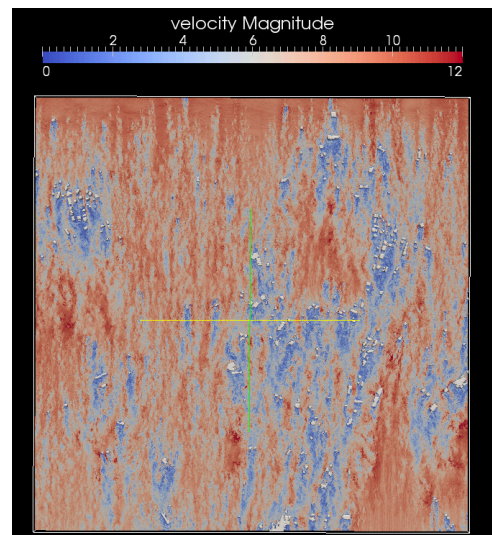


図 10 Wind velocity profile at height of 100 meters.

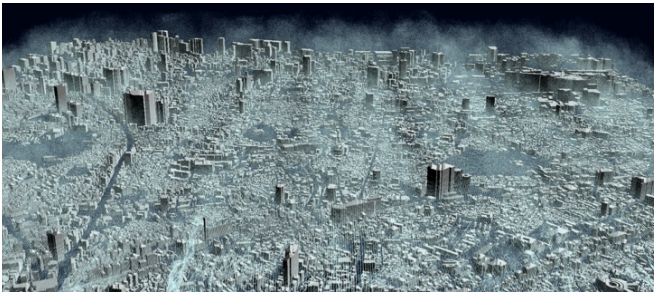


図 11 Stream lines describing airflow using particles in Minato-ku.

らに、中央部の高層ビル後方部では風の巻き込みにより、上空と地上付近の空気が混合されている事が確認できる。

図 16 に都庁周辺の流れ場を粒子により可視化した図を示す。空間上の固定された点から一定時間の間に放出された粒子は流れ場によって移流される。計算結果より、ビル群の後方では粒子の流れが乱された複雑な流れが、中央部の建物が無い領域では流線が繋がった比較的乱れの少ない流れが確認できる。

以上の結果より、実際の建物データを含む大規模都市計算の実行だけでなく、非定常乱流の細かな構造までとらえる事のできる LES 解析を初めて可能にした。

8. 結言

本研究では、格子ボルツマン法にラージエディ・シミュレーションの乱流モデルを適用する事で、東京都心部の 10 km 四方のエリアに対し、1m 間隔の格子解像度の超大規模気流シミュレーションを行った。

乱流のラージエディ・シミュレーションとして、LBM に CSM を適用する事で以下の事が明らかになった。

- 平行平板間乱流計算において、CSM を適用する事で $Re_\tau = 590$ と高いレイノルズ数の条件での長時間積分計算においても安定に計算を行うこと、および平均速度分布や乱流強度など各統計量が DNS へと収束することを確認した。
- CSM は局所的にモデル係数を決定可能であることより、1 m 格子で 10 km 四方を計算する大規模計算を初めて行う事ができた。

TSUBAME 2.0 を用いて実行性能を測定する事で以下の事が明らかになった。

- 単体 GPU を用いた計算において、高速化手法として SFU (special function unit) を用いた演算の高速化、および 32bit 環境でのコンパイルによる配列の index 計算の削減を行った結果、Improved Roofline Model の 92% の単精度実行性能が得られた。
- 複数 GPU を用いた計算において、計算と通信のオーバーラップ計算を行う事で、弱スケージングの結果において 768 GPU で 115 TFLOPS、1000 GPU で 149



図 12 Shinjyuku area corresponding to the computation. (2012Google ZENRIN)

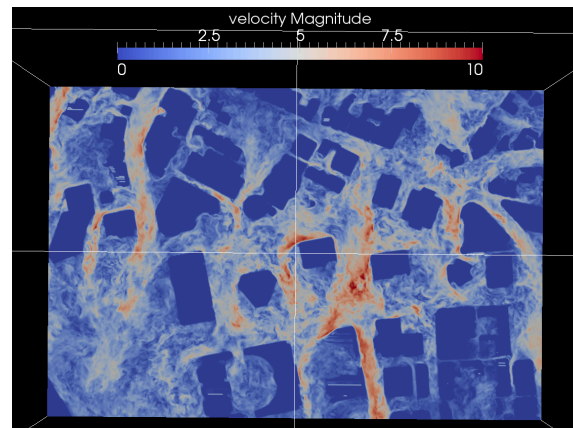


図 13 Wind velocity profile at height of 25 meters near by Tokyo-tochou.

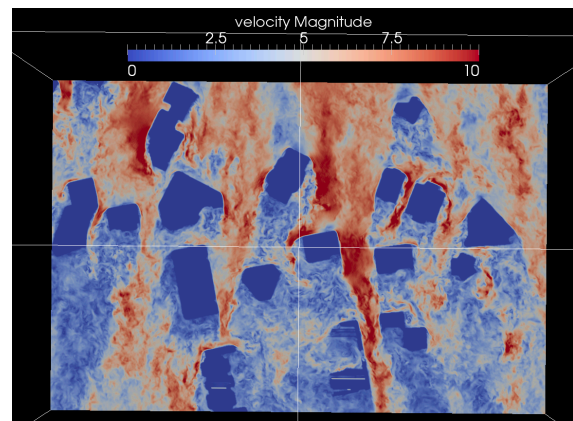


図 14 Wind velocity profile at height of 100 meters near by Tokyo-tochou.

TFLOPS の実行性能が得られた。

- 局所的にメモリ参照を行う事ができる CSM のアルゴリズムを適用した事により、弱スケージング、強スケージング共に良いスケージングが得られた。

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 間隔の格子解像度で詳細な気流シミュレーションを行い、以下の事が明らかになった。

- 東京都心全域での高さ 25 m の水平断面において、建物群の影響により、全体的に風速が遅い領域が分布し

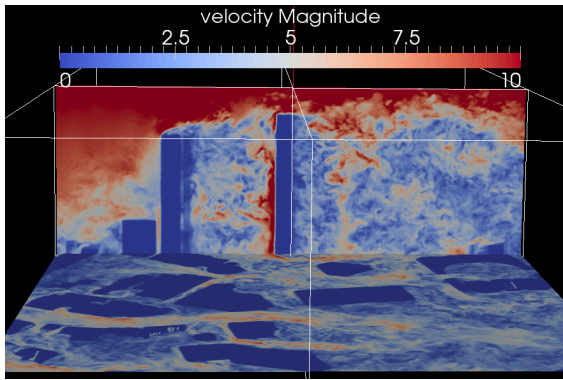


図 15 Side view of wind velocity profile near by Tokyo-tochou.

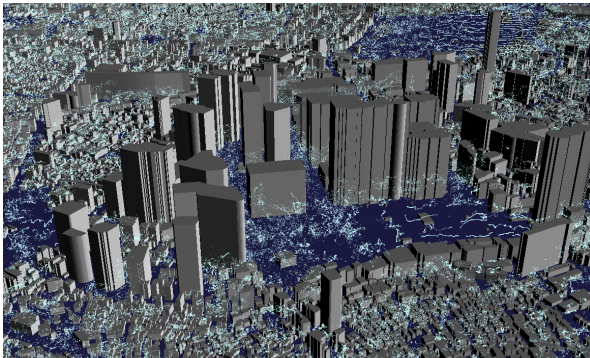


図 16 Stream lines describing airflow using particles near by Tokyo-tochou.

ている事が分かる．一方で，計算領域右下側などの海面部や左側の明治神宮，中央部の皇居など，高い建物が無い領域においては平均的な風速が大きくなっており，上空の高速な風が流れ込んでいる事が確認できる．

- 東京都心全域での高さ 100 m の水平断面の速度分布において，乱流の特徴的なストリーク構造（速度の速い・遅いが交互に存在する構造）が見られ，乱流拡散が活発に行われている事が再現されている．計算領域左上部の新宿副都心や計算領域中央部の六本木において，高層ビル群の影響で風速が下がっており，非常に広い領域に対して影響を与えている事が確認される．
- 東京都庁付近の高さ 25 m の速度分布において，主要道路に沿って風速が強い領域が存在しており，特に計算領域中央部から南方に向かって強い流れ（ビル風）が再現されている．
- 高さ 100 m の速度分布において，高層ビルの中に風速の非常に強い領域が存在しており，高さ 25 m の風速分布に大きな影響を与えている事が確認できる．高層ビル後方には風速の弱い領域が存在しており，非常に複雑な速度分布が形成されている．
- 東京都庁前の垂直断面内の速度分布において，高層ビルにより，速度の速い風が上空へと押し出されている様子が確認できる．
- 東京都心の実際の建物データを使い，高層ビルにより，

速度の速い風が上空へと押し出されている様子が確認できる．さらに，中央部の高層ビル後方部では風の巻き込みにより，上空と地上付近の空気が混合されている事が確認できる．

以上の結果より，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・モデルを格子ボルツマン法に導入することに成功し，1 m 解像度の格子を用いて広範囲の気流の大規模かつ非定常乱流の細かな構造までとらえる事ができる LES 解析を初めて可能にした．

謝辞 本研究の一部は科学研究費補助金・基盤研究（B）課題番号 23360046 「GPU スパコンによる気液二相流と物体の相互作用の超大規模シミュレーション」，科学技術振興機構 CREST 「次世代テクノロジーのモデル化・最適化による低消費電力ハイパフォーマンス」および「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」から支援を頂いた．記して謝意を表す．

参考文献

- [1] A. Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, et al. Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE Computer Society, 2010.
- [2] X. Wang and T. Aoki. Multi-gpu performance of incompressible flow computation by lattice boltzmann method on gpu cluster. *Parallel Computing*, 2011.
- [3] M. Bernaschi, M. Fatica, S. Melchionna, S. Succi, and E. Kaxiras. A flexible high-performance lattice boltzmann gpu code for the simulations of fluid flows in complex geometries. *Concurrency and Computation: Practice and Experience*, 22(1):1–14, 2009.
- [4] H. Yu, S.S. Girimaji, and L.S. Luo. Dns and les of decaying isotropic turbulence with and without frame rotation using lattice boltzmann method. *Journal of Computational Physics*, 209(2):599–616, 2005.
- [5] G. Kalitzin, G. Medic, G. Iaccarino, and P. Durbin. Near-wall behavior of rans turbulence models and implications for wall functions. *Journal of Computational Physics*, 204(1):265–291, 2005.
- [6] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3:1760, 1991.
- [7] DK Lilly. A proposed modification of the germano subgrid-scale closure method. *Physics of Fluids A: Fluid Dynamics*, 4:633, 1992.
- [8] H.Kobayashi, F. Ham, and X. Wu. Application of a local sgs model based on coherent structures to complex geometries. *International Journal of Heat and Fluid Flow*, 29(3), 2008.
- [9] Q. Zou and X. He. On pressure and velocity flow boundary conditions and bounceback for the lattice boltzmann bgk model. *Arxiv preprint comp-gas/9611001*, 1996.
- [10] C. Obrecht, F. Kuznik, B. Tourancheau, and J.J. Roux. A new approach to the lattice boltzmann method for

- graphics processing units. *Computers Mathematics with Applications*, 61(12):3628–3638, 2011.
- [11] J. Habich, C. Feichtinger, H. Kostler, G. Hager, and G. Wellein. Performance engineering for the lattice boltzmann method on gpgpus: Architectural requirements and performance results. *Computers Fluids*, 2012.
- [12] C. Feichtinger, J. Habich, H. Koestler, G. Hager, U. Ruede, and G. Wellein. A flexible patch-based lattice boltzmann parallelization approach for heterogeneous gpu-cpu clusters. *Parallel Computing*, 37(9):536–549, 2011.
- [13] X. Yin and J. Zhang. An improved bounce-back scheme for complex boundary conditions in lattice boltzmann method. *Journal of Computational Physics*, 2012.
- [14] B. Chun and AJC Ladd. Interpolated boundary condition for lattice boltzmann simulations of flows in narrow gaps. *Physical review E*, 75(6):066705, 2007.
- [15] C. Shu, N. Liu, and YT Chew. A novel immersed boundary velocity correction–lattice boltzmann method and its application to simulate flow past a circular cylinder. *Journal of Computational Physics*, 226(2):1607–1622, 2007.
- [16] JS Smagorinsky. General circulation model of the atmosphere. *Monthly Weather Review*, 164:91–9, 1963.
- [17] ER Driest. van.(1956). On turbulent flow near a wall. *J. Aeronaut. Sci*, 23:1007.
- [18] H. Kobayashi. Large eddy simulation of magnetohydrodynamic turbulent channel flows with local subgrid-scale model based on coherent structures. *Physics of Fluids*, 18:045107, 2006.
- [19] R.D. Moser, J. Kim, and N.N. Mansour. Direct numerical simulation of turbulent channel flow up to $re= 590$. *Physics of Fluids*, 11:943, 1999.
- [20] T. Shimokawabe, T. Aoki, C. Muroi, J. Ishida, K. Kawano, T. Endo, A. Nukada, N. Maruyama, and S. Matsuoka. An 80-fold speedup, 15.0 tflops full gpu acceleration of non-hydrostatic weather model asuca production code. In *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*, pages 1–11. IEEE, 2010.
- [21] T. Shimokawabe, T. Aoki, T. Takaki, T. Endo, A. Yamanaka, N. Maruyama, A. Nukada, and S. Matsuoka. Peta-scale phase-field simulation for dendritic solidification on the tsubame 2.0 supercomputer. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 3. ACM, 2011.