

原子力防災に向けた風況デジタルツイン開発

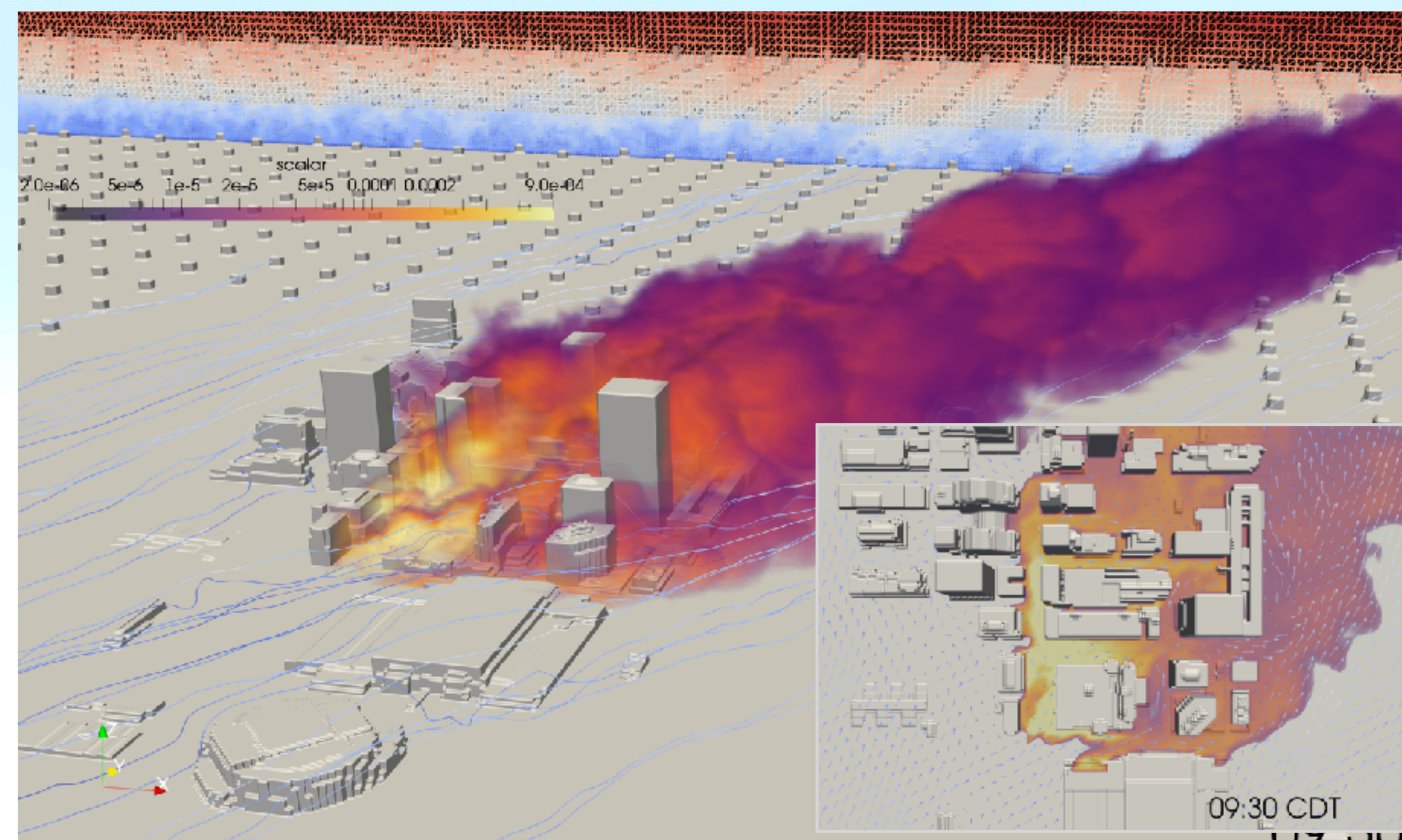
謝辞

- 日本原子力研究開発機構 大型計算機利用
- 基盤B：GPUスーパーコンピュータを用いた1m解像度リアルタイム風況デジタルツイン
- JHPCN2023：都市街区の風況デジタルツインの実現に向けたデータ同化手法および観測システムの開発

研究背景/研究目的

原子力サイト/都市街区内の風況デジタルツイン

- 原子力施設周辺の気象・観測データに基づく、汚染物質の拡散監視システム
 - 都市街区内の熱中症評価や突風予測などスマートシティ設計・運用
- ➡高解像度の風況予測用の物理モデル
- ➡リアルタイム予測のための高性能計算
- ➡広域/局所の観測データの同化技術



原子力施設近傍の汚染物質監視システムのイメージ

マルチスケールの風況解析の実現@東工大2012 - 2015

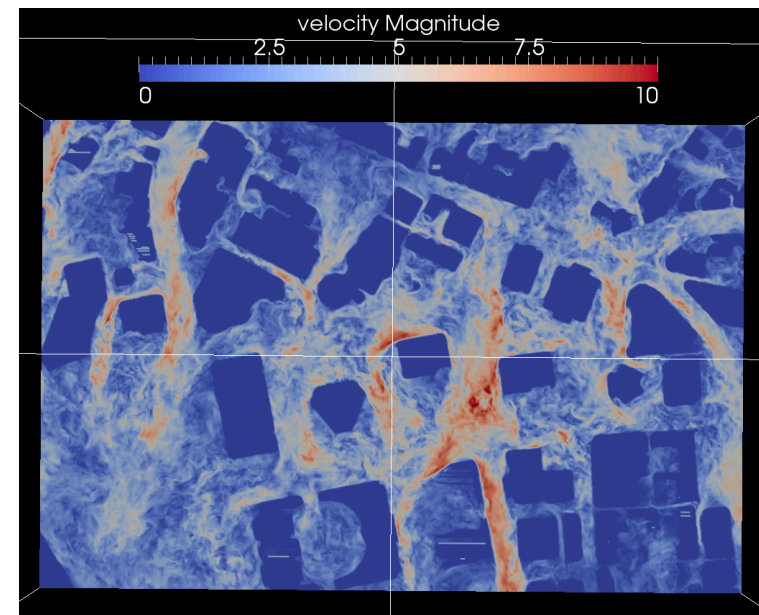
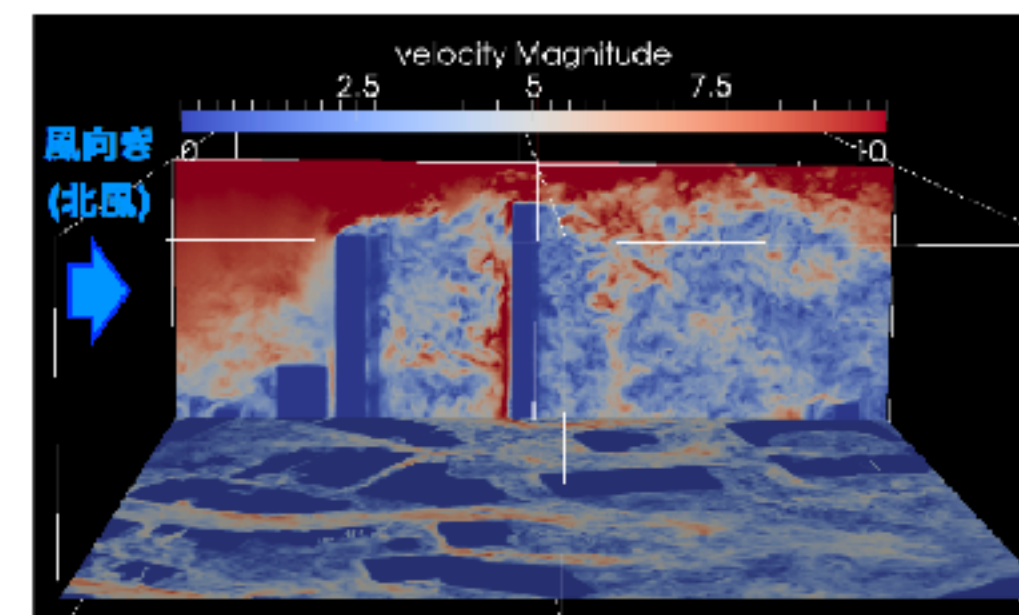
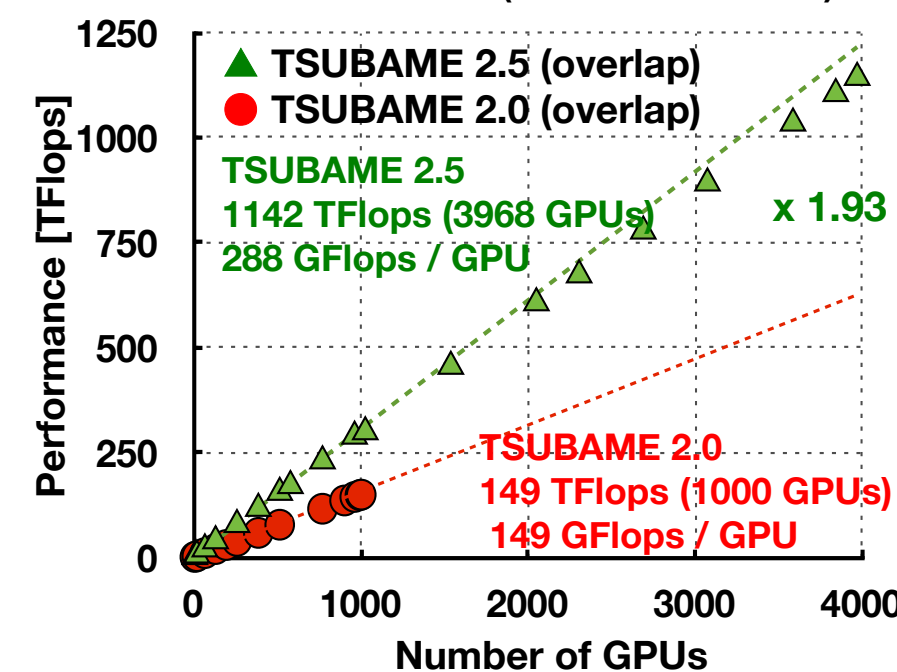
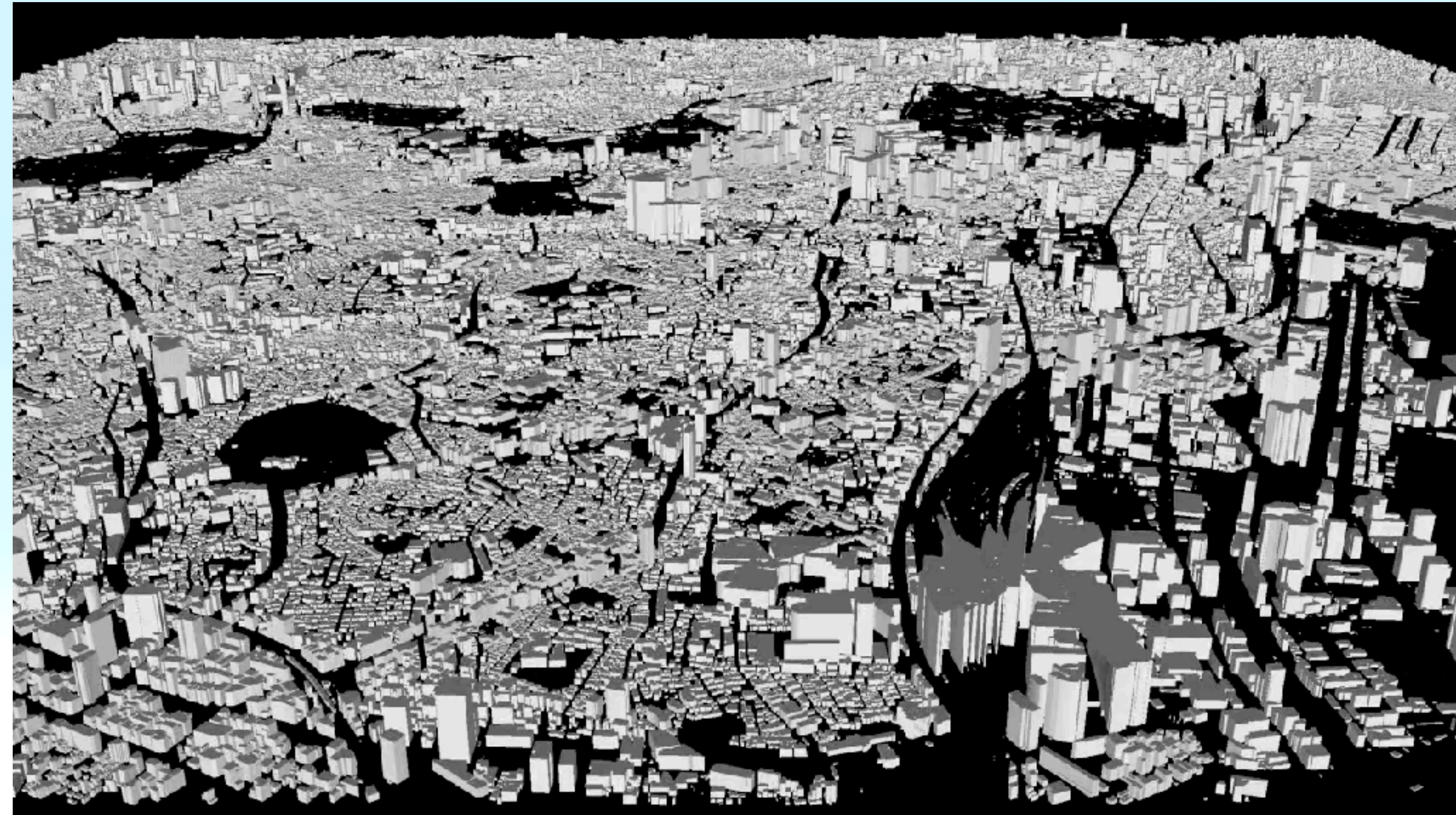
都市街区の風況解析実現のための技術開発

- 格子ボルツマン法のGPU最適化 (CPU比で約10倍の高速化)
 - 都市の風況解析に適した乱流モデル
 - GPU計算/MPI通信の同時実行による高速化
- ➔東工大のスパコンの全てのGPUを利用した解析により、世界で初めて東京都心部1m解像度10km四方の解析を実現* (実行性能=1.14 PFlops, 3968GPUs)

課題

- 計算速度：リアルタイムの30倍遅い
- 計算資源：大規模すぎる = 4000台のGPU

*小野寺直幸, 青木尊之, 下川辺隆史, 小林宏充, "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", 情報処理学会ハイパフォーマンスコンピューティング研究会主催 HPCS シンポジウム 2013、最優秀論文賞



東京都心部1m解像度10km四方の風況解析
500億格子(10,080x10,240x512, 4,032GPUs@TSUBAME2.0)

リアルタイムの風況解析の実現@原子力機構 2018 - 現在

リアルタイムの風況解析実現のための技術開発

- 適合細分化格子 (AMR) 法に基づく計算資源の削減 (従来手法の1/10以下)

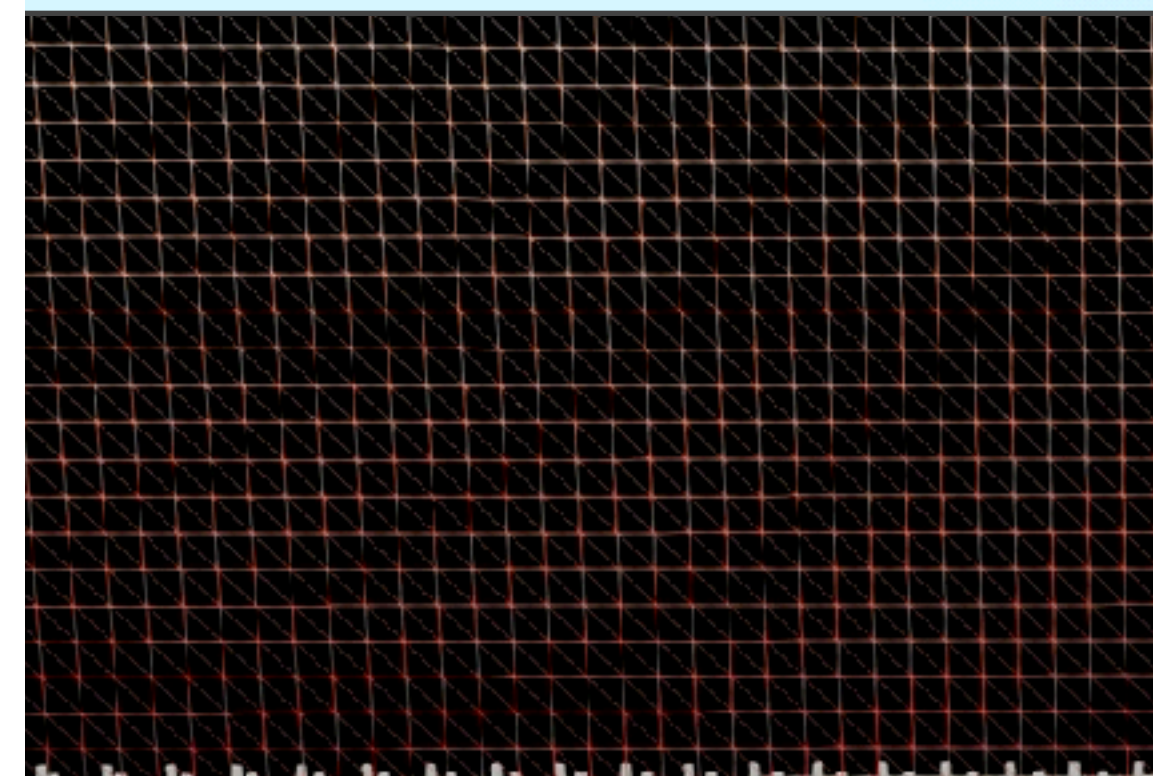
➡2m解像度4km四方のリアルタイムのマルチスケール風況解析を実現@16 GPUs(V100)

➡オクラホマシティの野外拡散実験の解析にて、Factor2*の正答率70%以上を達成 (*計算値が実験値の0.5~2倍以内)

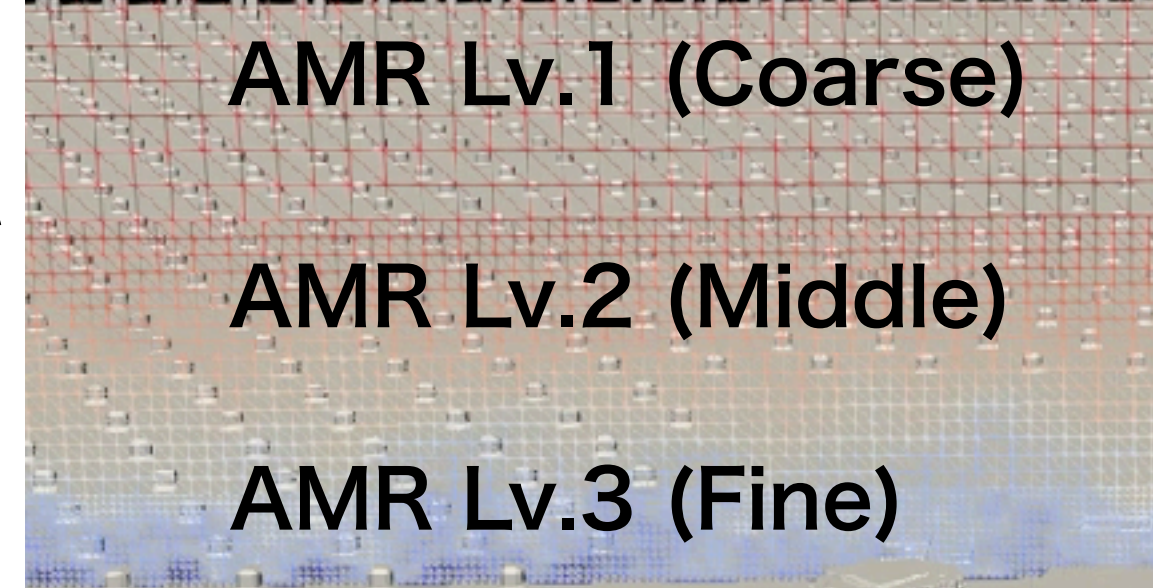
x 課題：複雑な気象条件および都市条件への対応

デジタルツインに向けた新たな研究開発

- 都市街区モデルの高精度化 (建物・地形・植生)
- 複雑な建物間の風況予測に向けたデータ同化技術



AMR Lv.1 (Coarse)

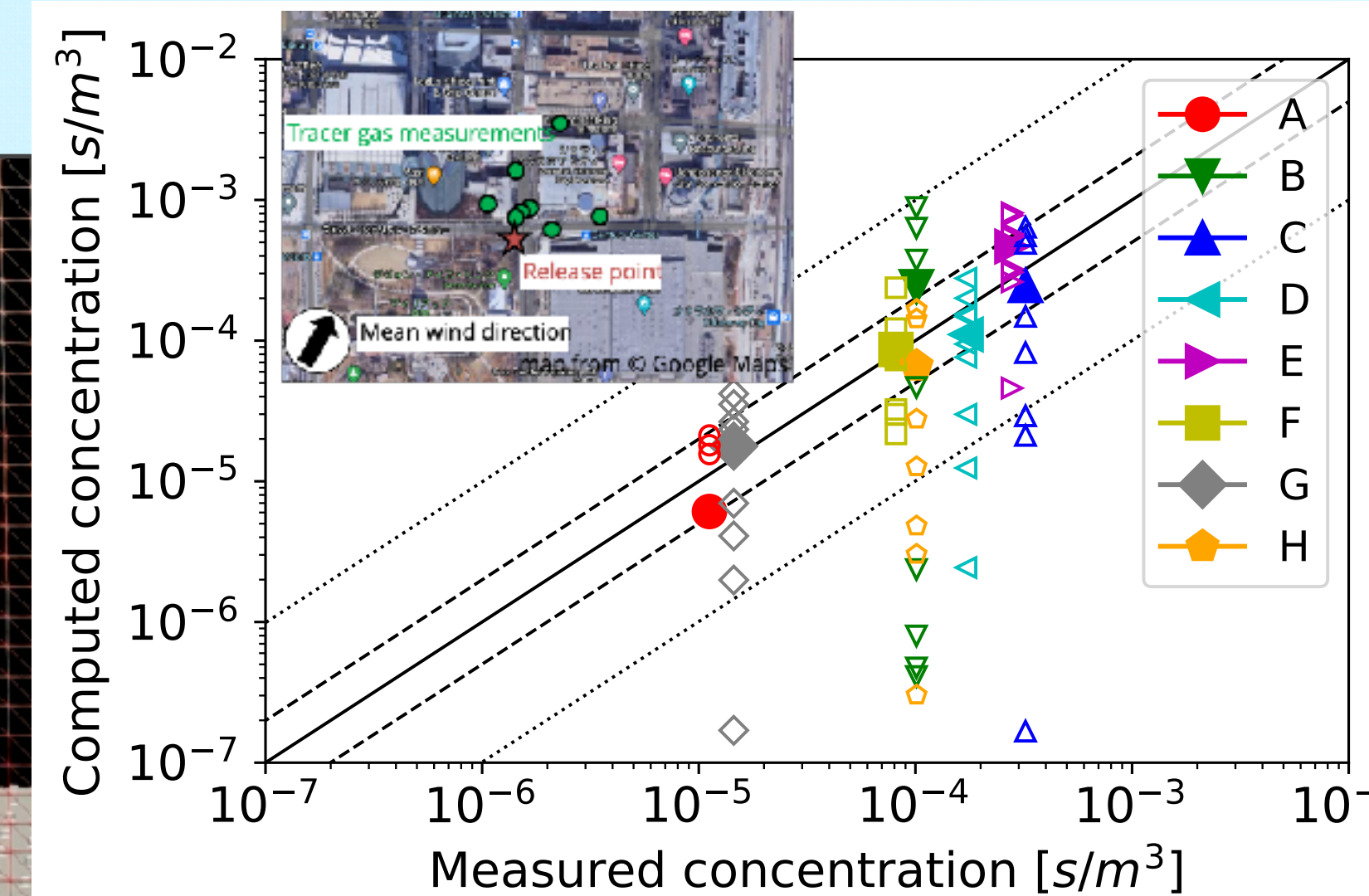


AMR Lv.2 (Middle)



AMR Lv.3 (Fine)

09:03 CDT



速度場・汚染物質濃度の可視化、および、汚染物質濃度の実験と計算の比較 (色付き記号がアンサンブル平均値、白抜き記号が各計算値)

* N. Onodera, Y. Idomura, Y. Hasegawa, H. Nakayama, T. Shimokawabe, T. Aoki, "Real-time tracer dispersion simulation in Oklahoma City using locally mesh-refined lattice Boltzmann method", Boundary-Layer Meteorology, 2021. doi: 10.1007/s10546-020-00594-x

CityLBMコード

共同研究先に公開中 <https://github.com/CityLBM/CityLBM.git>

- 格子ボルツマン法 (LBM)
- 都市風況向けの物理モデル
- メソスケール境界条件
- プログラミングモデル
- 実行性能測定

CityLBMコードの定式化

格子ボルツマン法と差分法のハイブリッド計算 / AMRによる計算量削減

速度 → 格子ボルツマン法 (Cumulant model)

$$f_{ijk}(x + c_{ijk}\Delta t, t + \Delta t) = f_{ijk}(x, t) + \Omega_{ijk}(x, t) + F_{ijk}$$

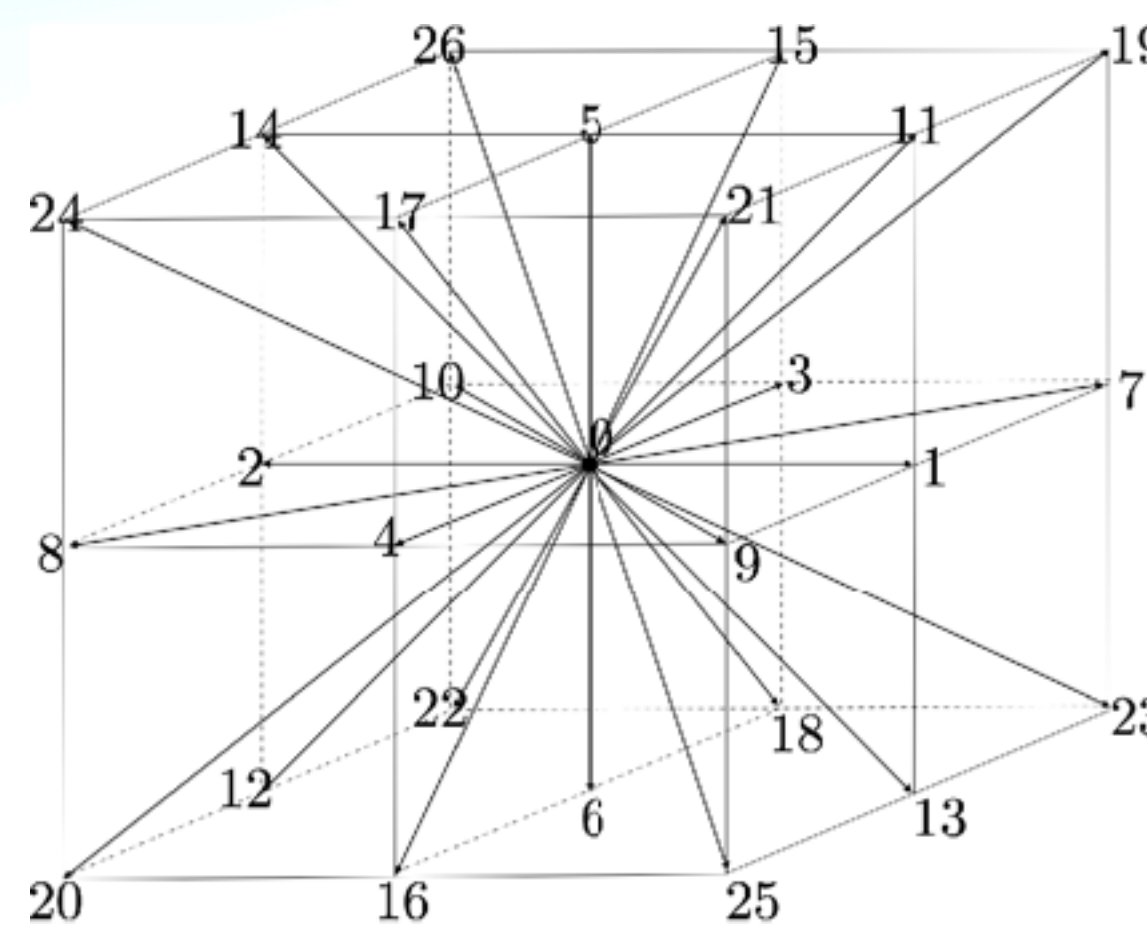
$$F_{buoyancy} = -g\alpha\Delta T \quad F_{plant\ model} = -\rho c_d a_f u_i \sqrt{u_j^2}$$

温位 → 有限差分法

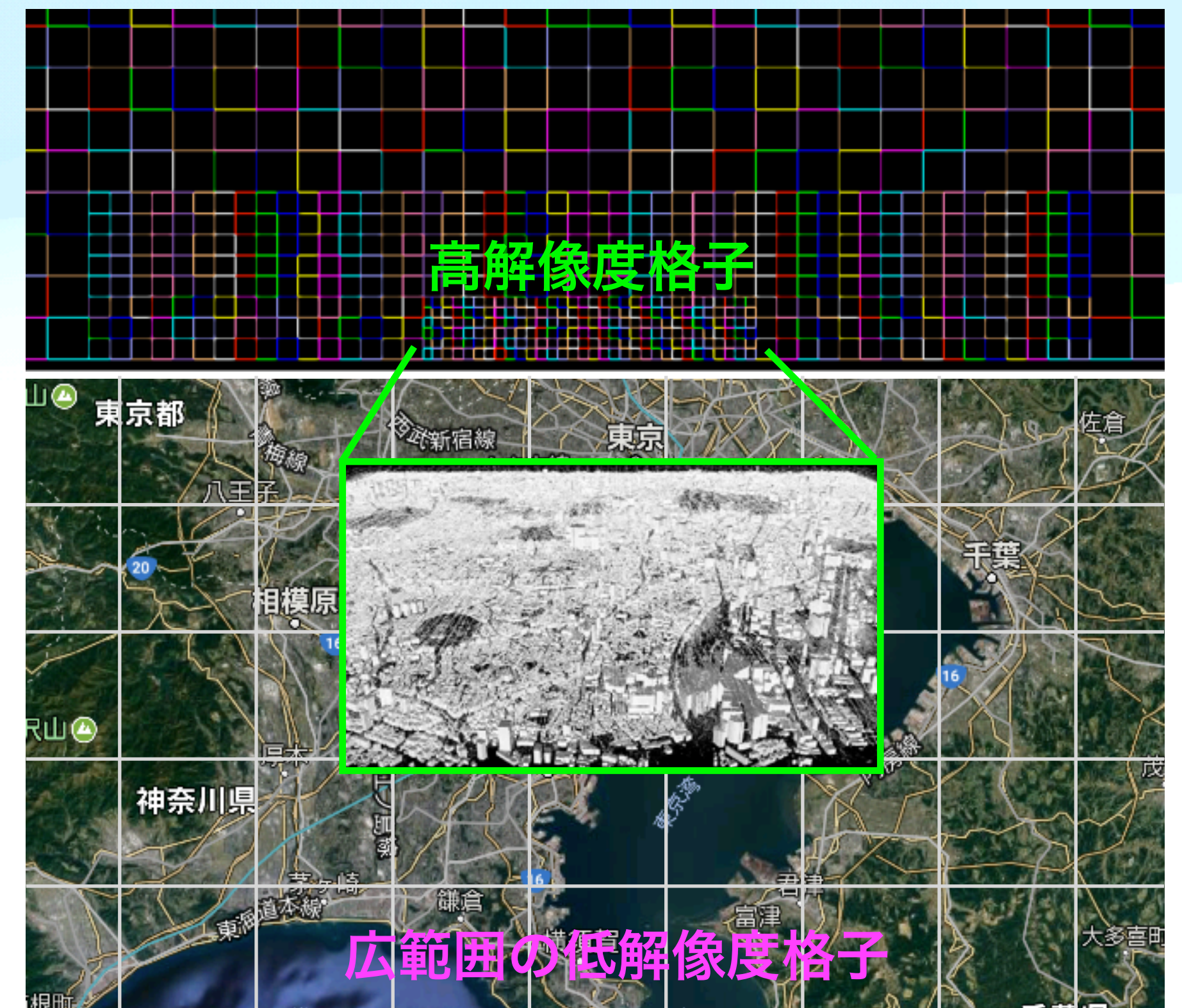
$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} = C_T \frac{\partial^2 T}{\partial x_j^2}$$

汚染物質濃度 → 有限体積法

$$\frac{\partial q}{\partial t} + \frac{\partial u_j q}{\partial x_j} = s$$



D3Q27 model



適合細分化格子によるマルチスケール風況解析

都市風況向け物理モデル

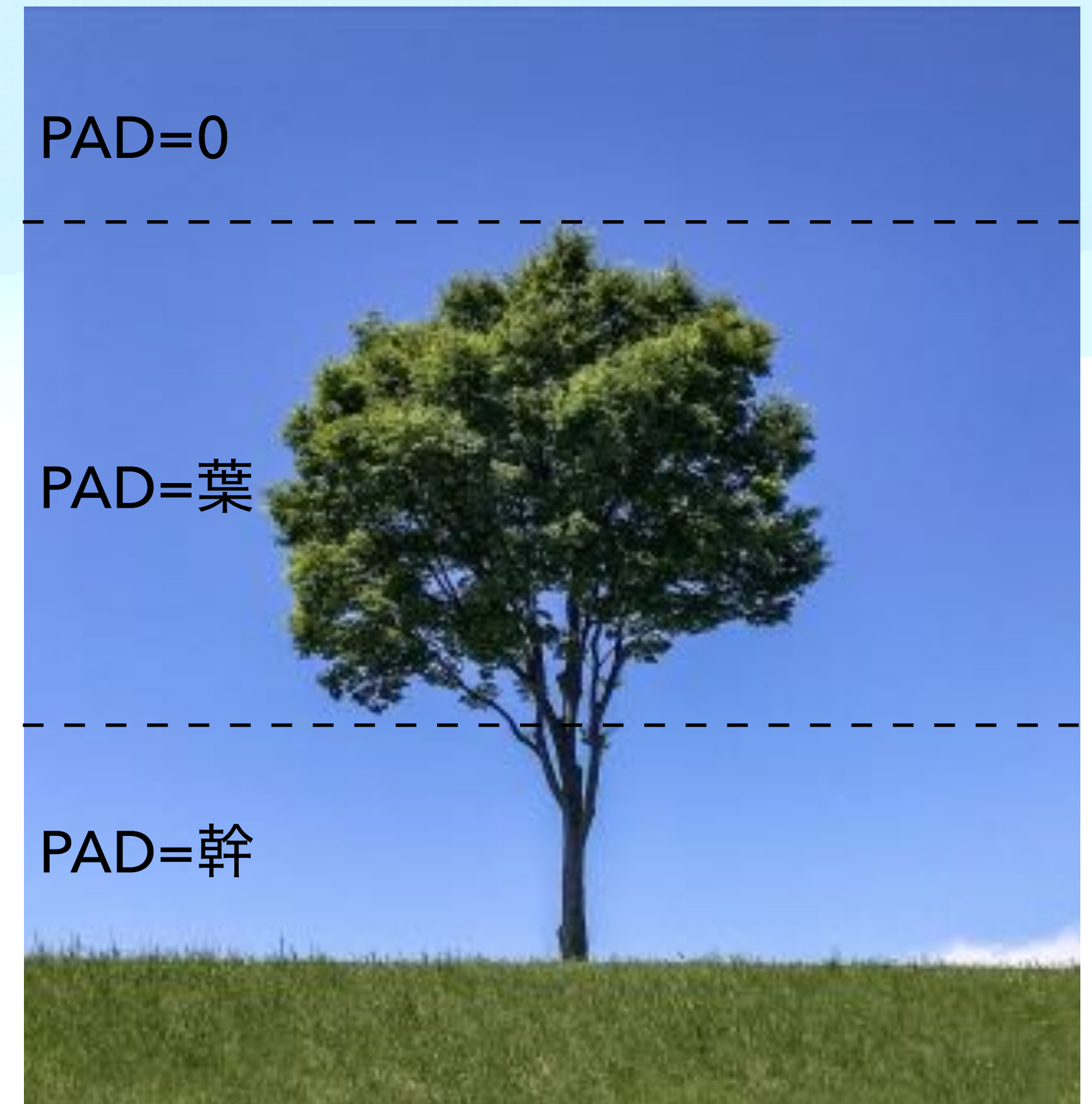
植生キャノピーモデルによるメートル・スケールの樹木の評価

- ・ 数m解像度の解析では、公園等の樹木の評価が重要
- ・ 植生の影響を外力項として評価
- ・ モデル係数が局所的に求められるため、GPUに適している

$$F_{d,i} = -\rho c_d a_f u_i \sqrt{u_j^2}$$

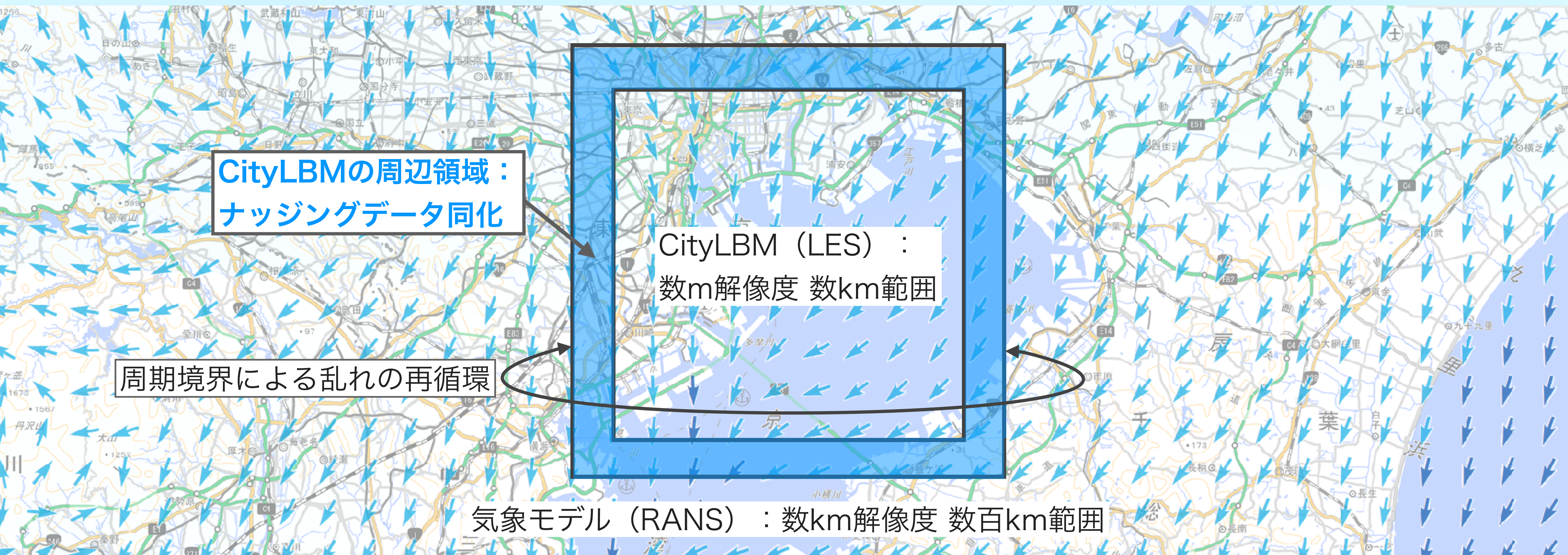
c_d : Drag coefficient (=0.1)

a_f : One-sided plant-area density (PAD=0.1), $\text{m}^2 \text{m}^{-3}$



参考文献 : Large-Eddy Simulation Comparison of Neutral Flow Over a Canopy: Sensitivities to Physical and Numerical Conditions, and Similarity to Other Representations, Boundary-Layer Meteorol, (2017)

メソスケール気象予測モデルを用いた境界条件 ナッジングデータ同化手法＋周期境界条件による乱れの生成



気象モデル (RANS) とCityLBM (LES) の周期境界で発達した流れをナッジングにより融合

$$F_{LBM,new} = \alpha_{nud} F_{LBM} + (1 - \alpha_{nud}) F_{WRF} \quad \in \text{Nudging region}$$

CityLBMのデータ構造およびプログラミングモデル

AMR格子およびNVIDIA/AMD GPU、Intel/AMD CPU対応

Leafのオフセット



- Forest-of-octrees (非構造データ) による複雑格子配置
 - Leaf (outer loop) : OpenMPスレッド/CUDAブロック
 - ブロック型 (=4³) 格子による高速な連続メモリアクセス
 - 格子 (inner loop) : SIMD演算/CUDAスレッド
- マルチプラットフォームでの開発効率・計算速度を両立

```

5 __HD__ void RhsPoisson(
6     const int* id_tasks, const int num_tasks,
15     ) noexcept
16 {
17     FOR_EACH1D_BLOCKIDX(l, num_tasks) {
18         const int idl = id_tasks[l];
19
20         int offset7[7];
21         for (int idv=0; idv<7; idv++) { offset7[idv] = mesh_offsets7[idl*7 + idv]; }
22
23         FOR_EACH3D(i, j, k, NX_LEAF, NX_LEAF, NX_LEAF) {
24             auto ID = [&](int _i, int _j, int _k) {
25                 return IndexMG::id_offset7<NX_LEAF>(i+_i, j+_j, k+_k, offset7);
26             };
27
28             const type du = u[ ID( 1, 0, 0 ) ] - u[ ID( 0, 0, 0 ) ];
29             const type dv = v[ ID( 0, 1, 0 ) ] - v[ ID( 0, 0, 0 ) ];
30             const type dw = w[ ID( 0, 0, 1 ) ] - w[ ID( 0, 0, 0 ) ];
31
32             Rhs[ ID( 0, 0, 0 ) ] = -( du + dv + dw ) / (dx*dt);
33         } // FOR_EACH3D
34     } // FOR_EACH1D_BLOCKIDX
35 }

```

```

29 #ifdef __CUDA_ARCH__
30     #define FOR_EACH1D_BLOCKIDX(L, NL) \
31         const auto L = blockIdx.x;
32
33     #define FOR_EACH3D(I, J, K, NX, NY, NZ) \
34         const auto I = threadIdx.x; \
35         const auto J = threadIdx.y; \
36         const auto K = threadIdx.z;
37
38     #define FOR_EACH1D(IJK, NN) \
39         const auto IJK = threadIdx.x + NX_LEAF*threadIdx.y + NX_LEAF*threadIdx.z;
40
41     #define SKIP_FOR() return
42 #else
43     #define FOR_EACH1D_BLOCKIDX(L, NL) \
44         PRAGMA_FOR_SIMD \
45         _Pragma("omp parallel for") \
46         for(int L=0; L<NL; L++)
47
48     #define FOR_EACH3D(I,J,K, NX,NY,NZ) \
49         PRAGMA_FOR_SIMD \
50         for(int K=0; K<NZ; K++) \
51         PRAGMA_FOR_SIMD \
52         for(int J=0; J<NY; J++) \
53         PRAGMA_FOR_SIMD \
54         for(int I=0; I<NX; I++)
55
56     #define FOR_EACH1D(IJK, NN) \
57         PRAGMA_FOR_SIMD \
58         for(int IJK=0; IJK<NN; IJK++)
59
60     #define SKIP_FOR() continue
61 #endif

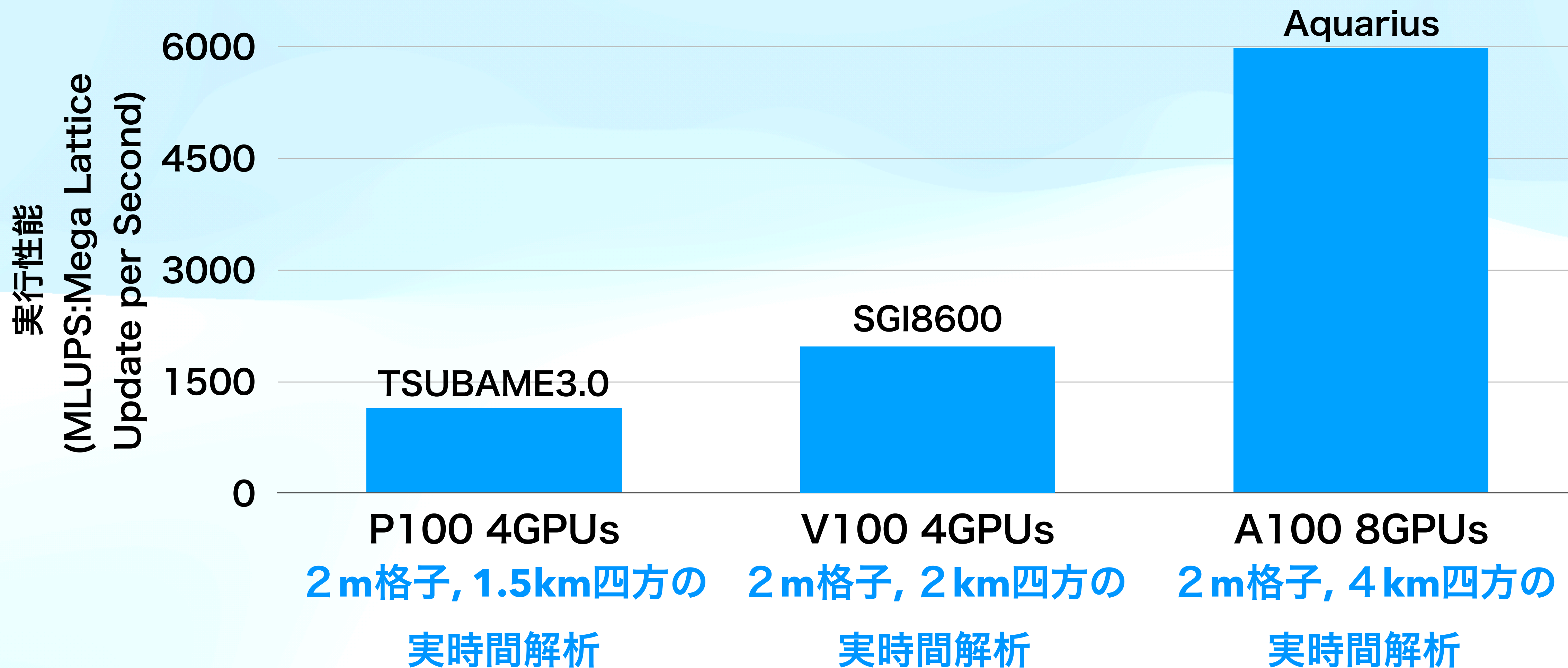
```

GPU

CPU

都市街区を捉えた実時間風況解析の実現範囲

1ノードの計算範囲 (2m解像度、1アンサンブル)



東京工業大学周辺の風況解析

- 都市モデルの作成
- ドップラーライダーとの比較

計算条件：東工大周辺の風況シミュレーション

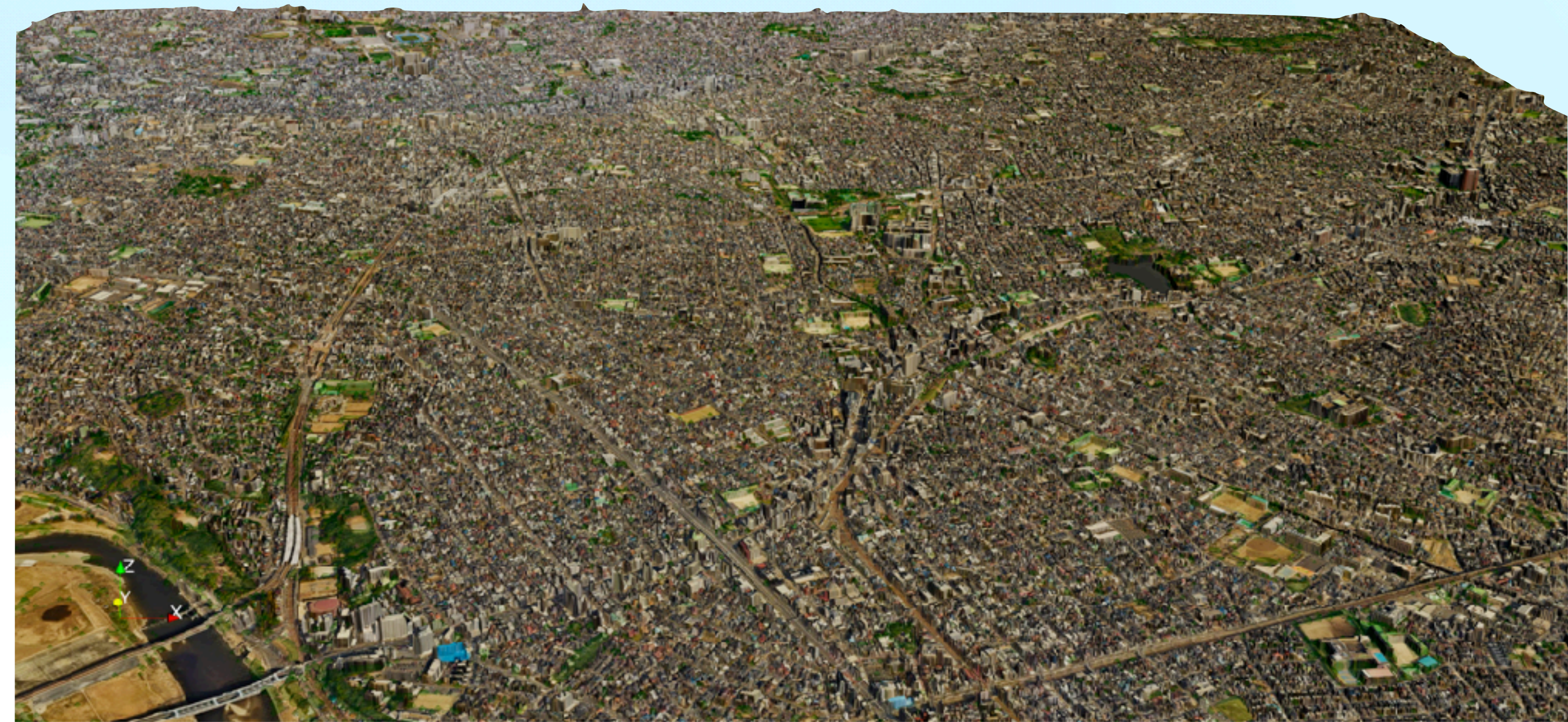
地理情報システム（GIS）に基づく建物・地形・植生データの利用

建物・地形データの作成

- ・ 建物・地形のGeoTiffデータを基に、CityLBM用のLevelsetデータを作成
- ➡ Interpolated bounce-back法にて、滑り無し境界条件を計算

植生データの作成

- ・ 航空写真を基に樹木の領域を抽出
 - ・ 樹木高さは一定値（8m）に設定
- ➡ Plant canopyモデルにて抵抗力を計算



都市街区風況シミュレーションの計算領域の可視化例

計算領域：6,144m x 6,144m x 1,536m

格子解像度：4m³（0～256m）、8m³（256～512m）、16m³（512～1536m）

検証：2022年12月7日の気象条件に基づく風況解析

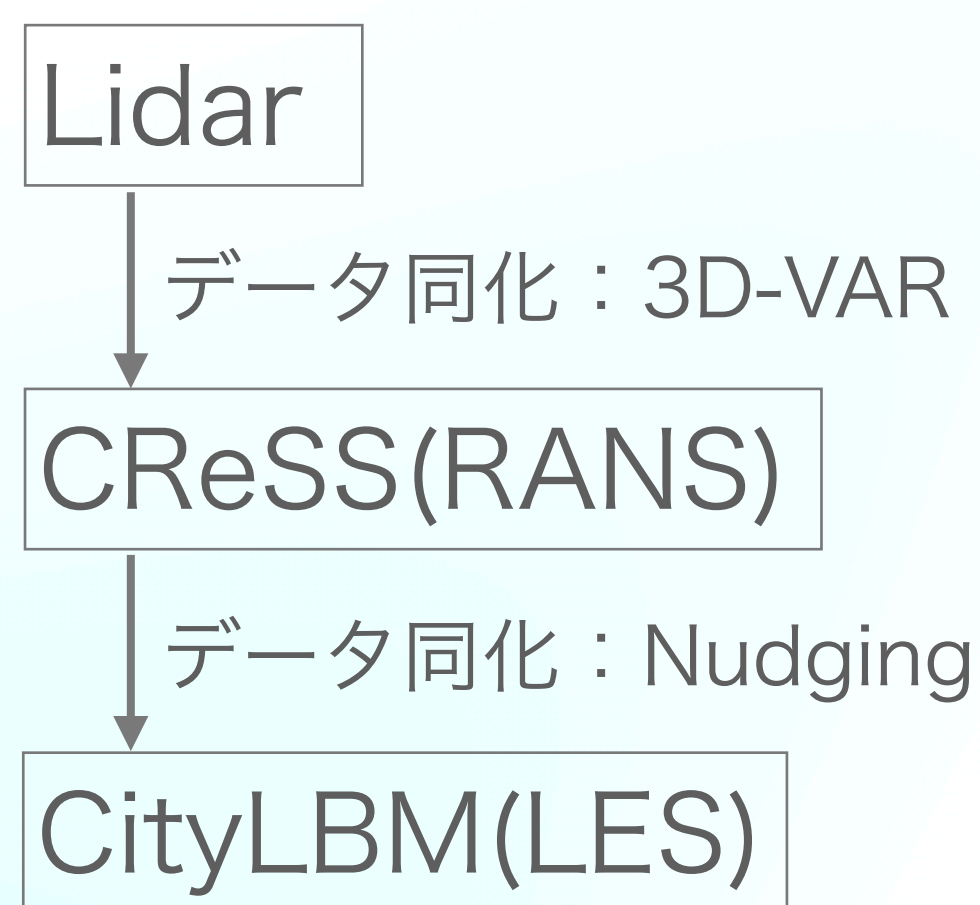
CityLBMによる気象モデルのダウンスケーリング計算

ドップラーライダー（高さ約81mに設置）

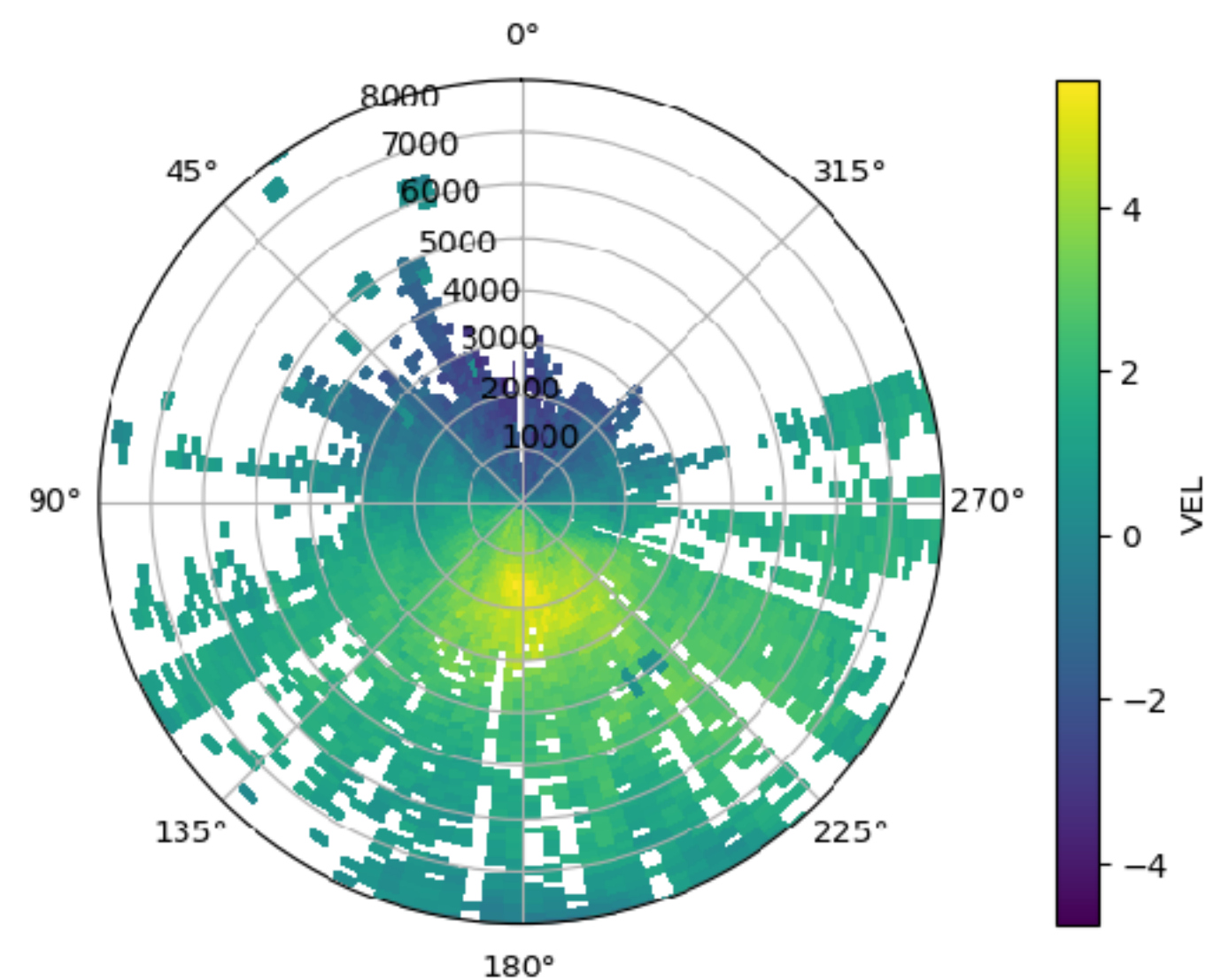
- ・視線方向速度を観測（離れる方向が正）、信号ノイズ比（SNR）が15以上を可視化

気象モデル CReSS（水平方向1km解像度、防災科研）

- ・ドップラーライダーおよび気象庁から配信されるデータを同化
- ・CityLBMの境界条件として、CReSSから与えられる速度・温位を使用



CityLBMでのライダーを模擬した観測点配置



ドップラーライダーが観測した速度分布

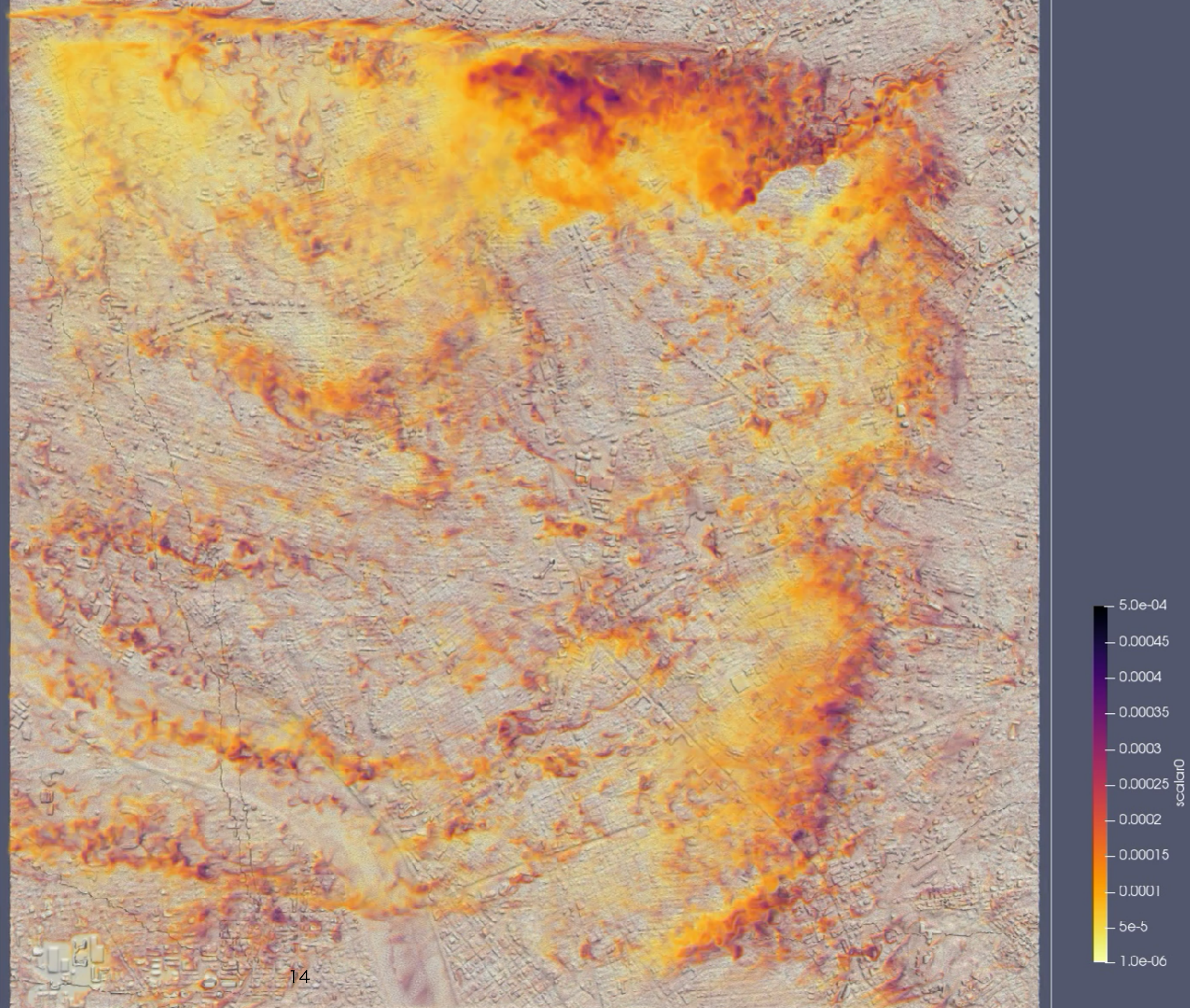
東工大周りの 流れ場の可視化

期間：7:00-17:00

発生源：4m高さ

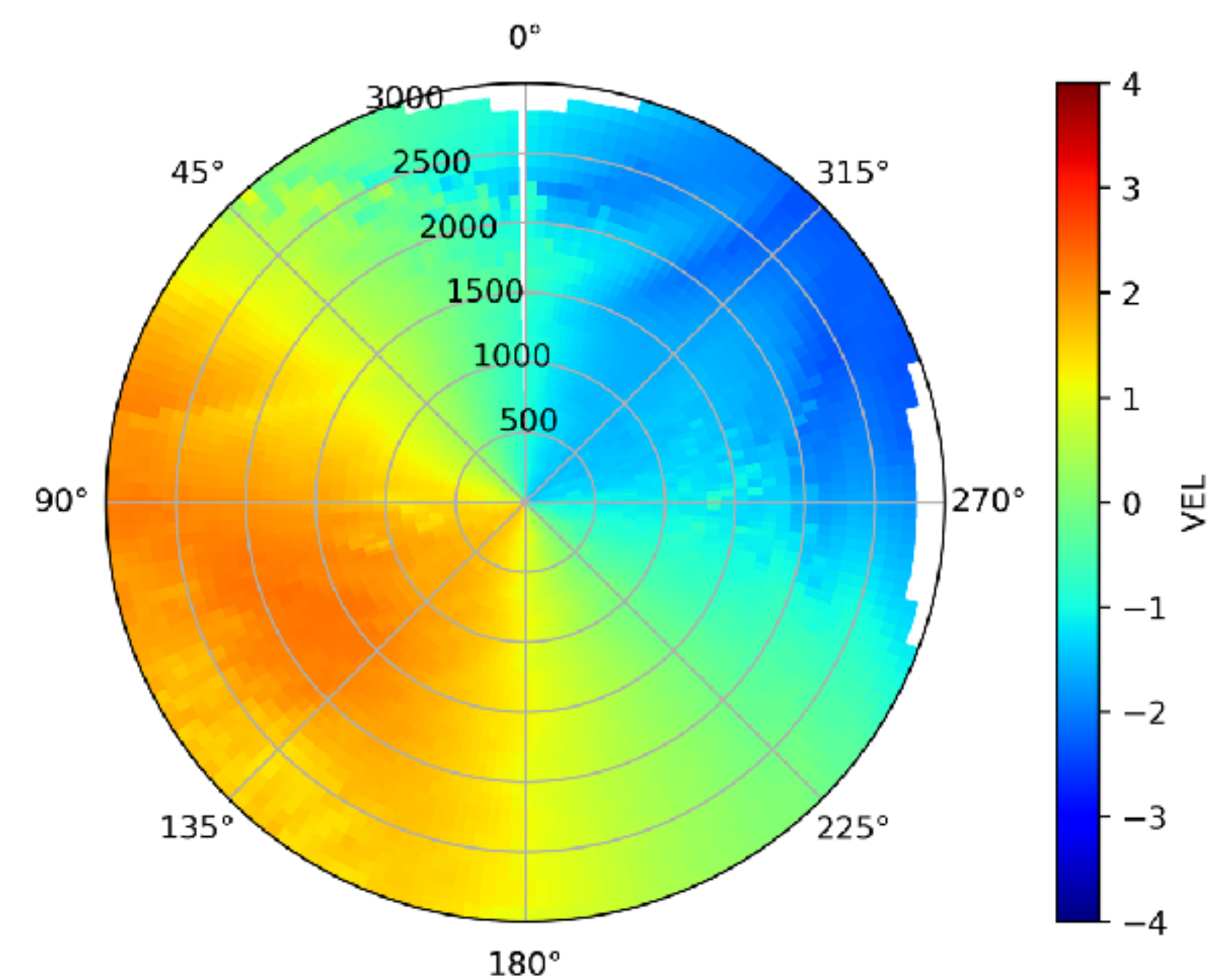
解像度：4m解像度

範囲：6km四方

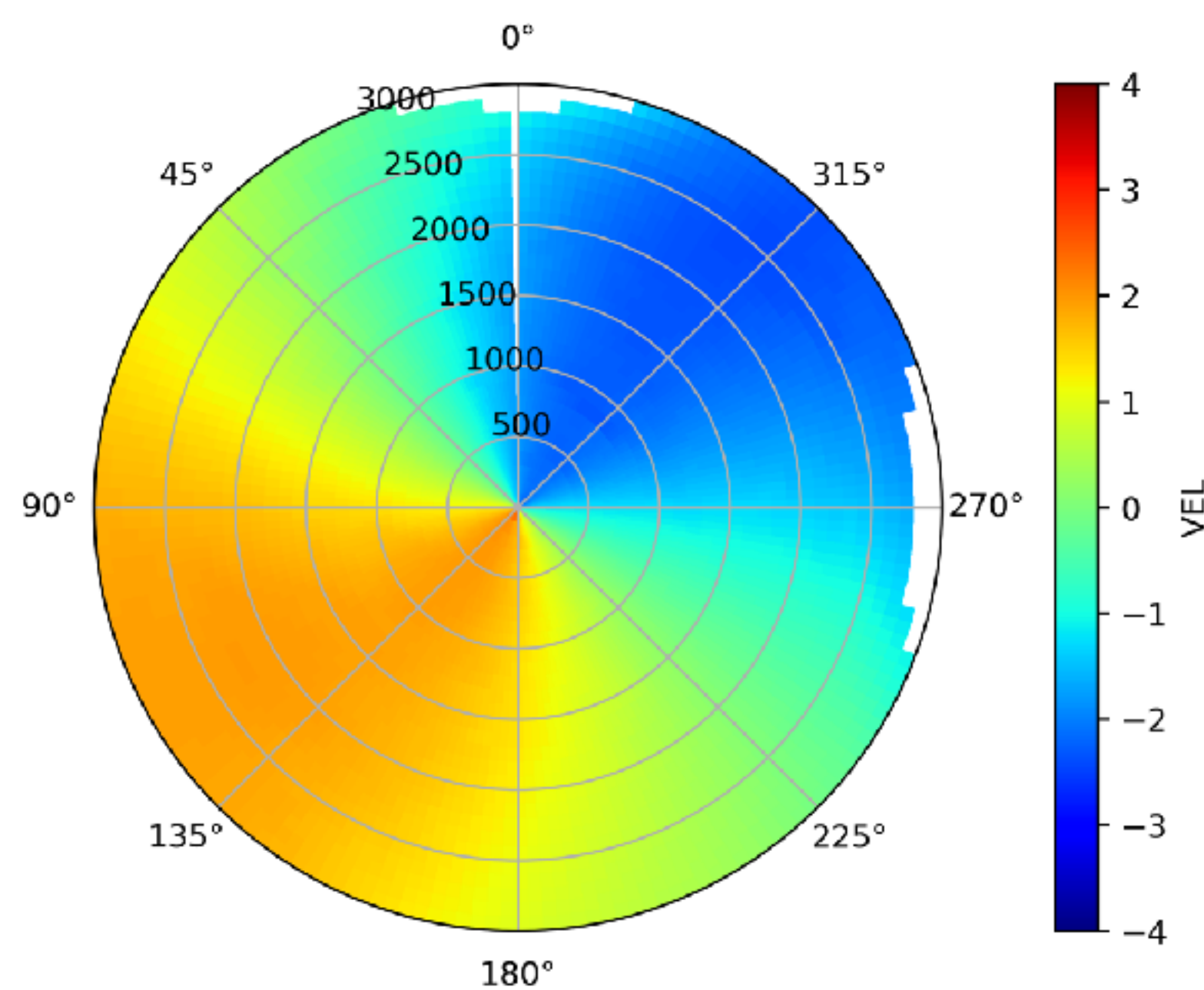


CityLBMによる風況シミュレーション

視線方向速度の比較@10:00JST

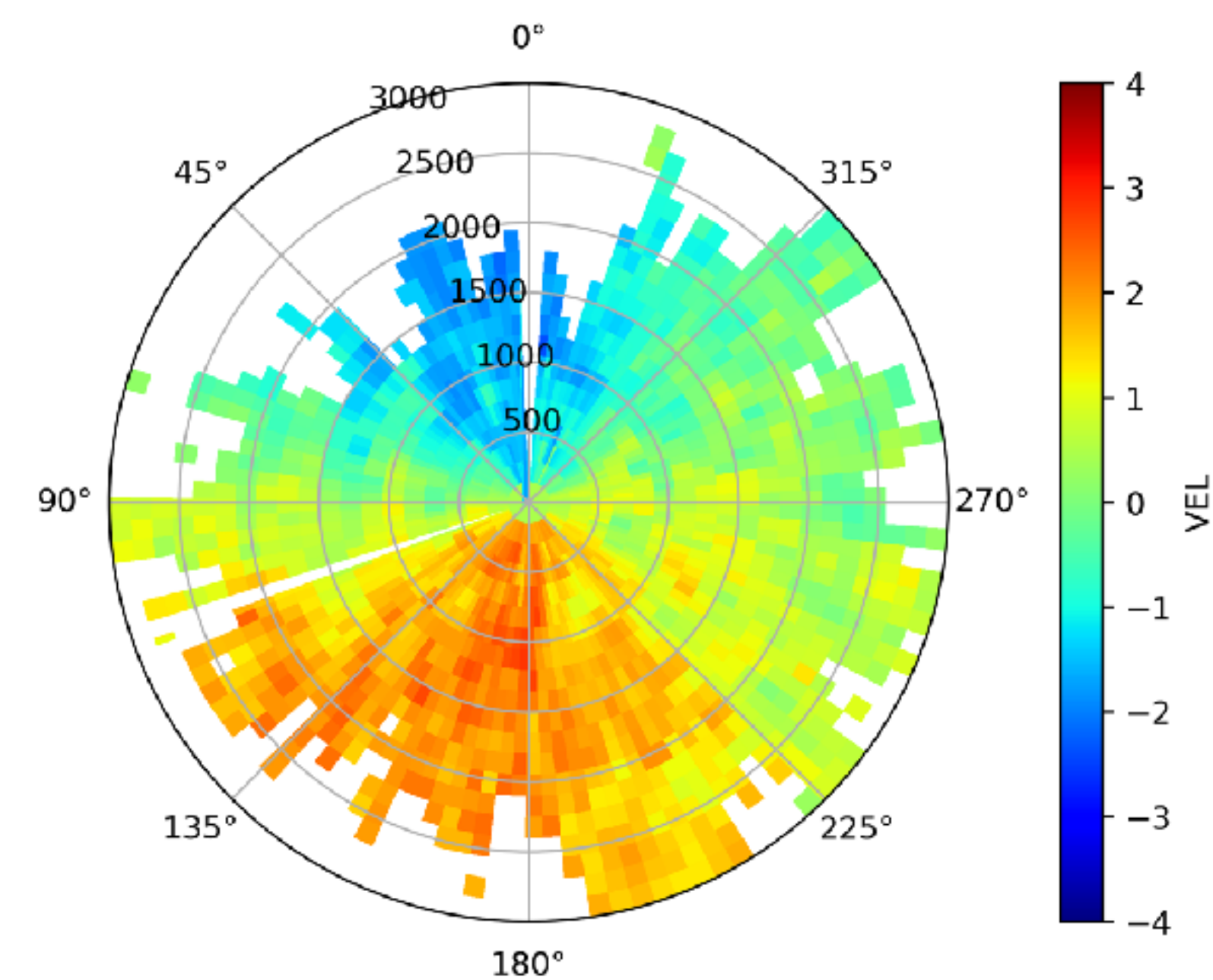


CityLBM(LES)



CReSS (RANS)

CityLBMの境界条件として使用

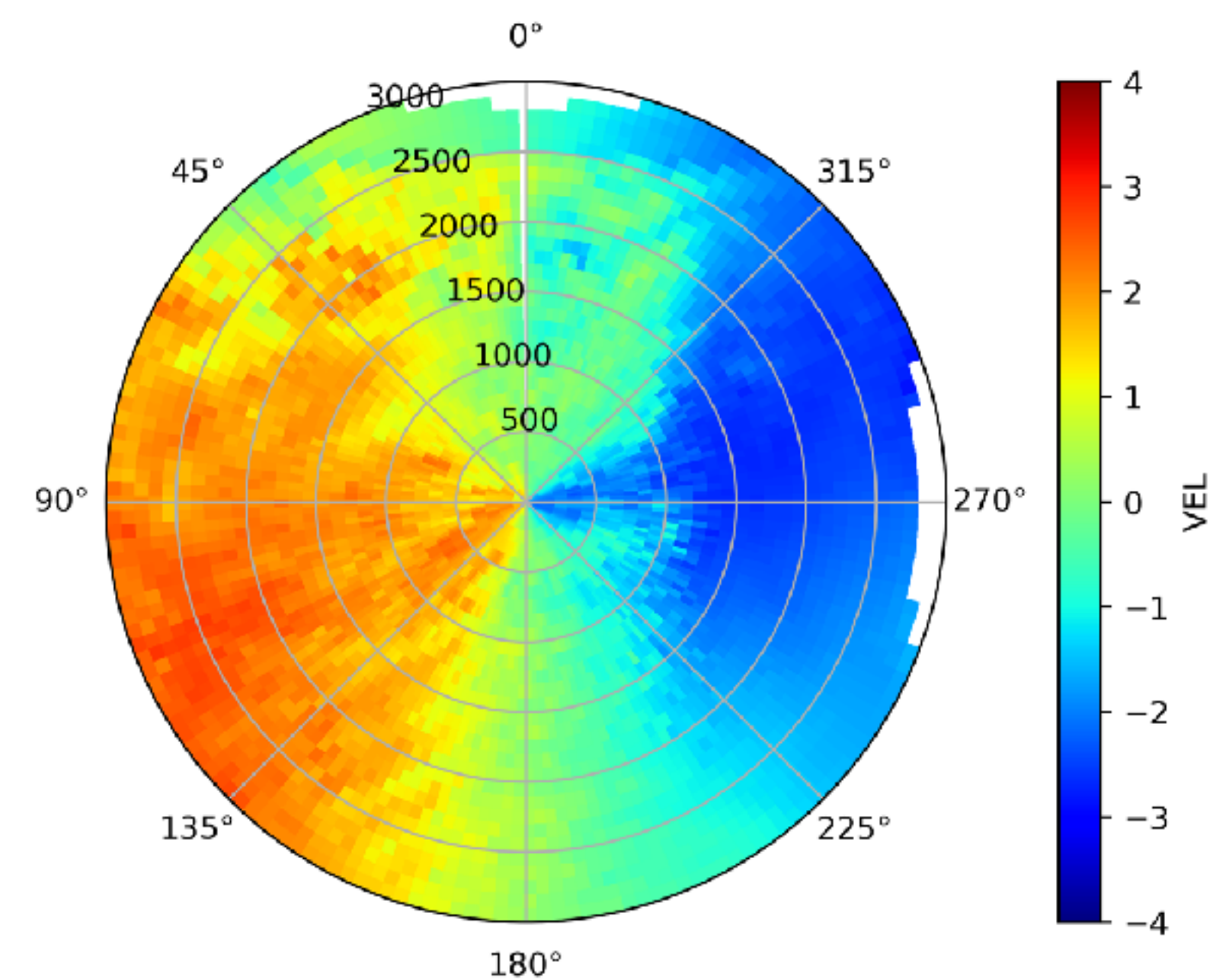


Lidar (SNR>15)

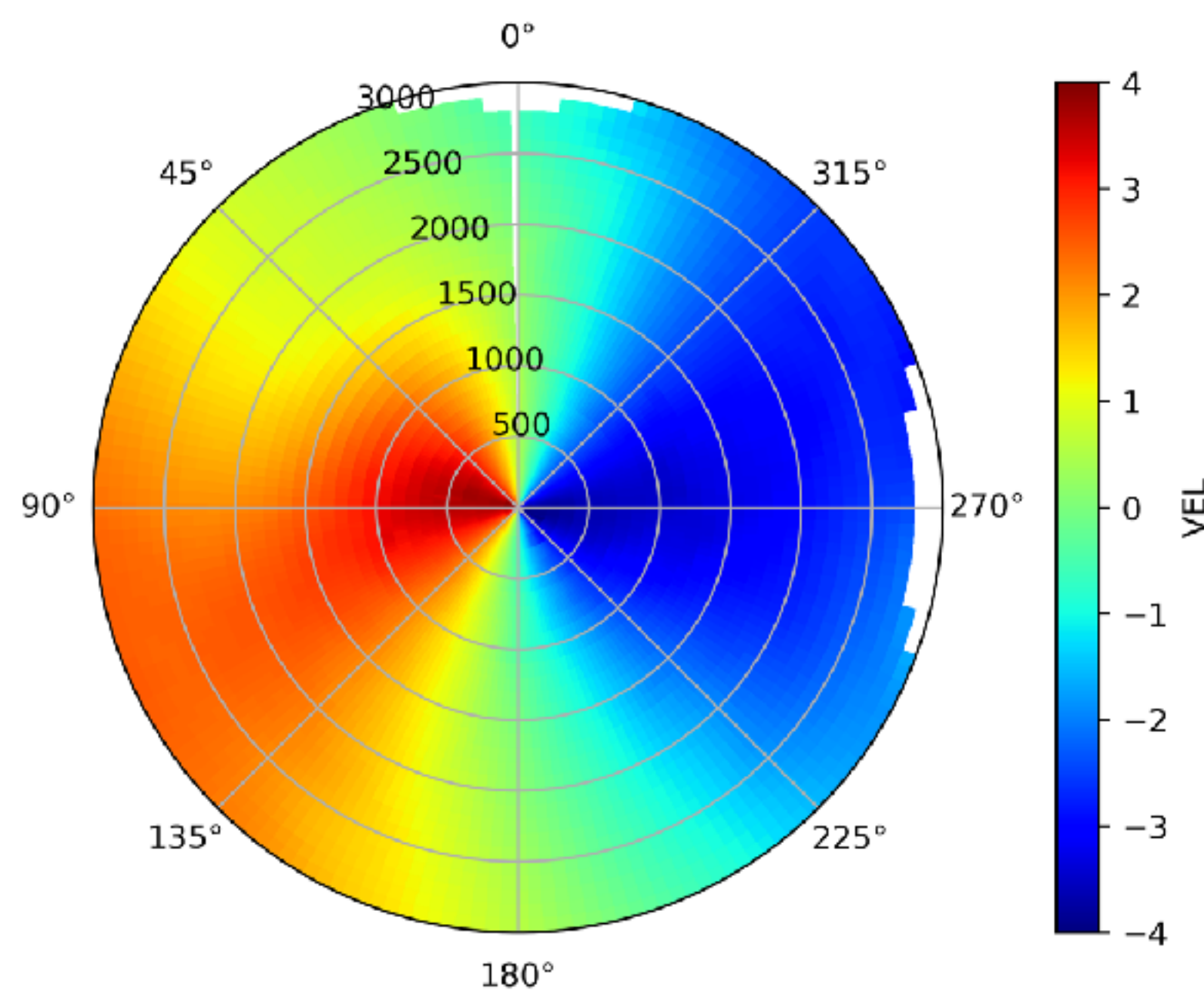
観測データ

CityLBMによる風況シミュレーション

視線方向速度の比較@11:00JST

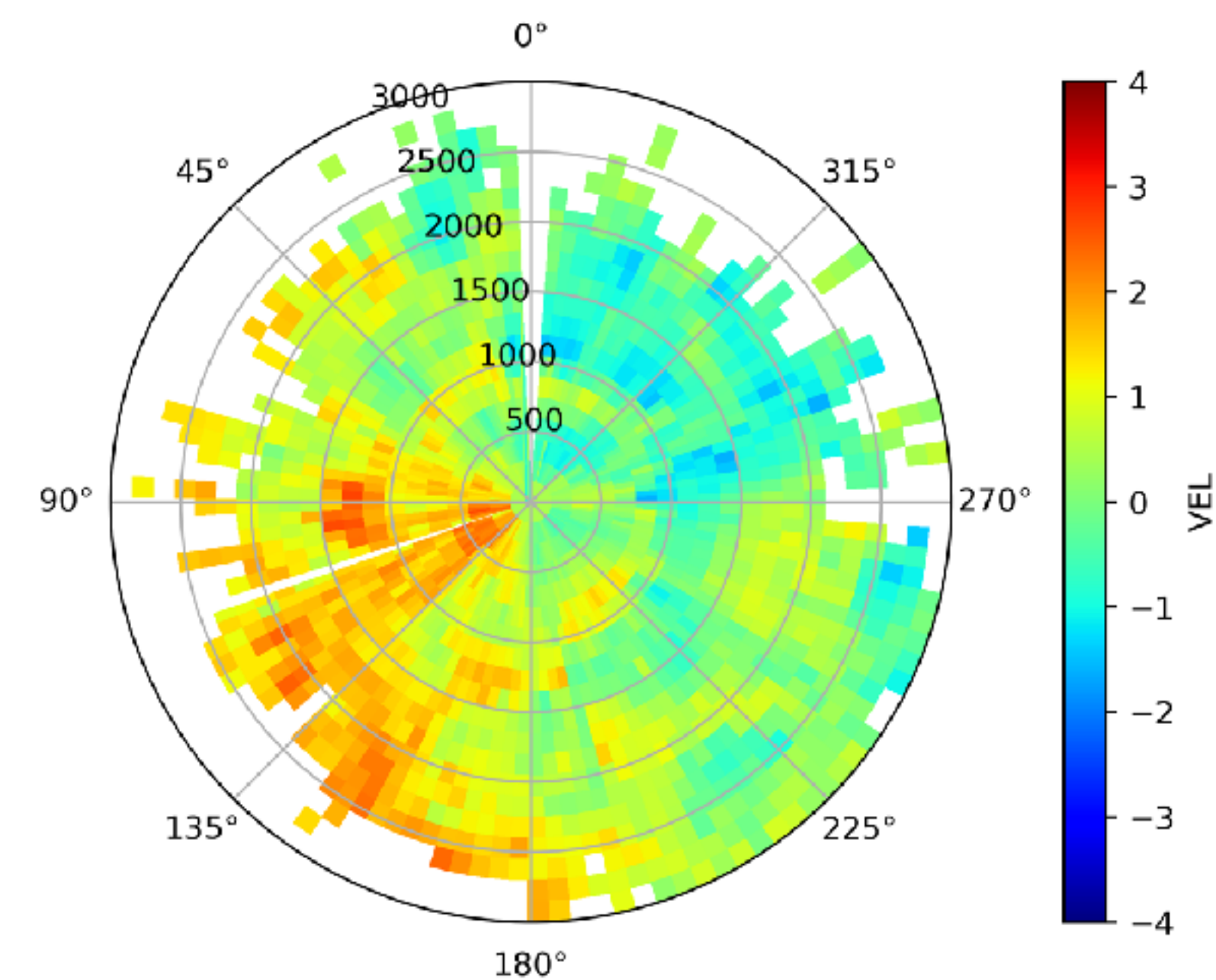


CityLBM(LES)



CReSS (RANS)

CityLBMの境界条件として使用

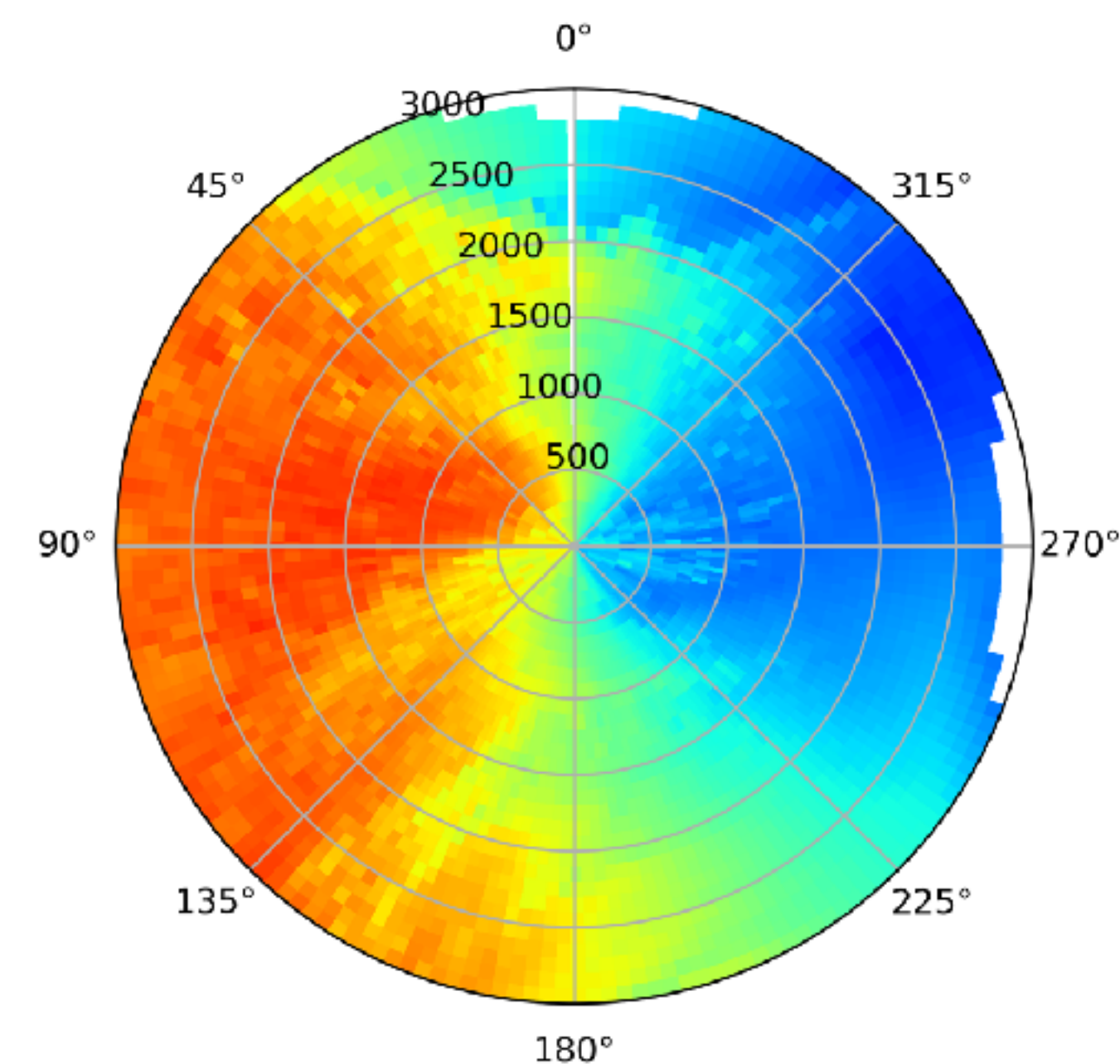


Lidar (SNR>15)

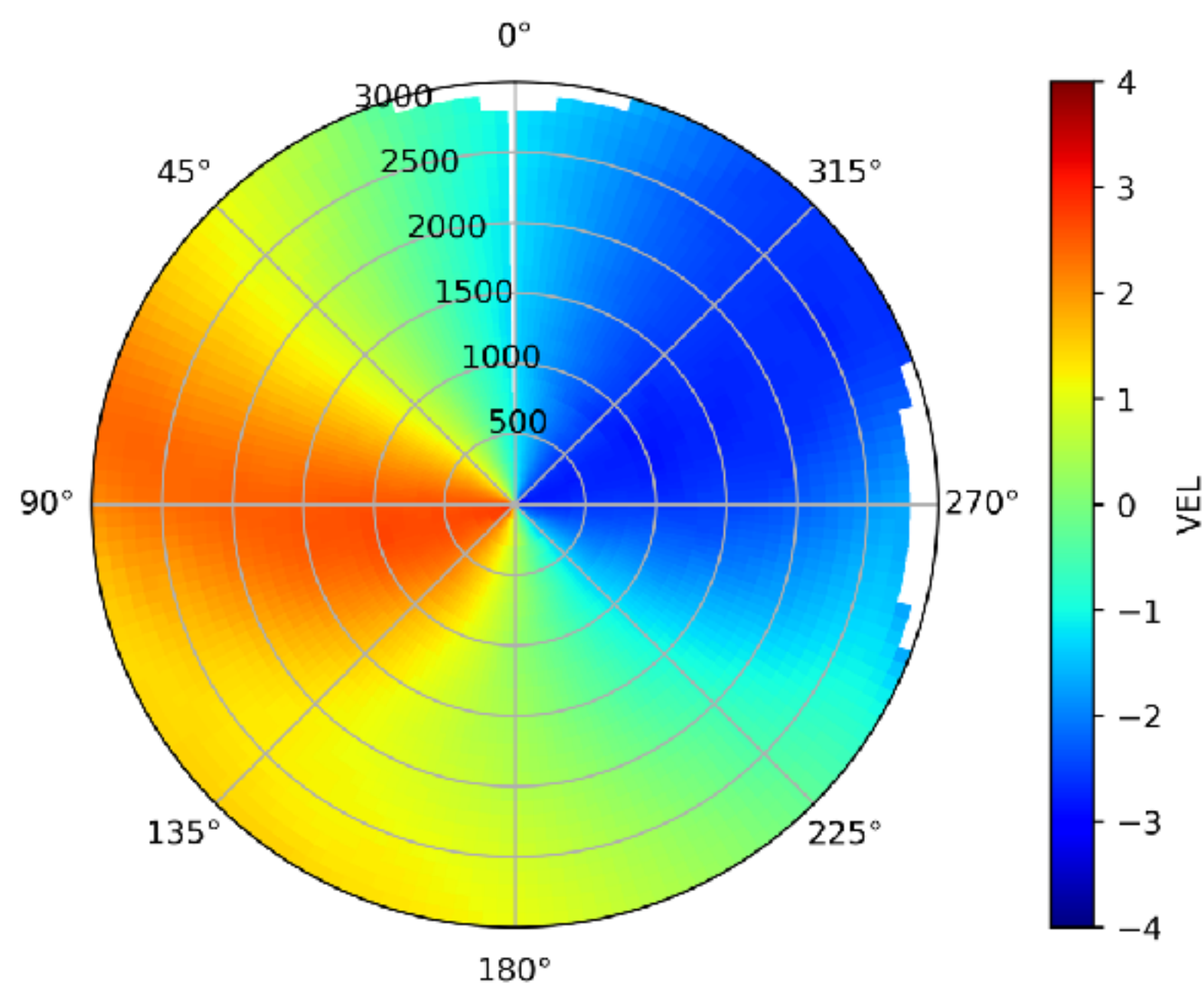
観測データ

CityLBMによる風況シミュレーション

視線方向速度の比較@13:00JST

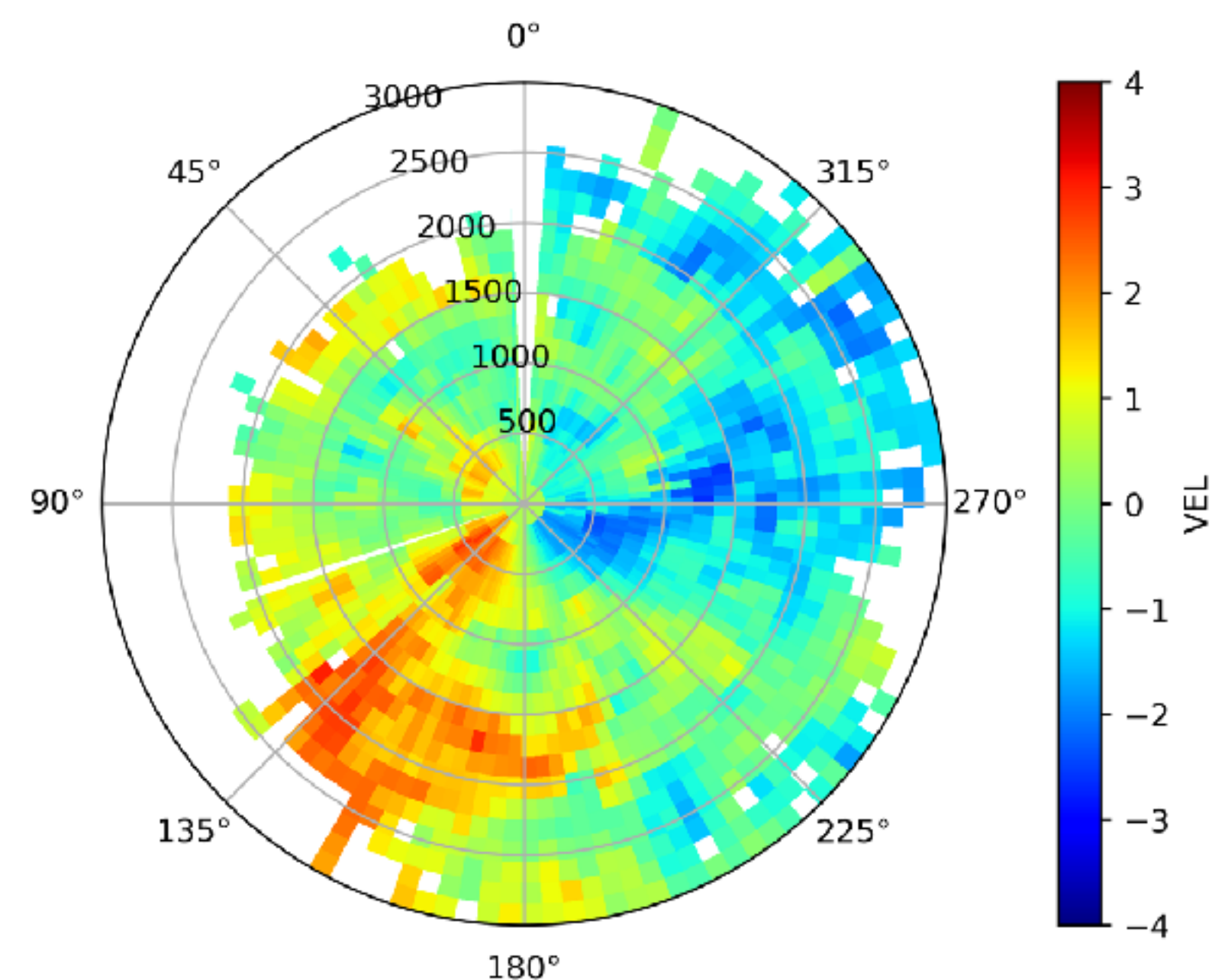


CityLBM(LES)



CReSS(RANS)

CityLBMの境界条件として使用

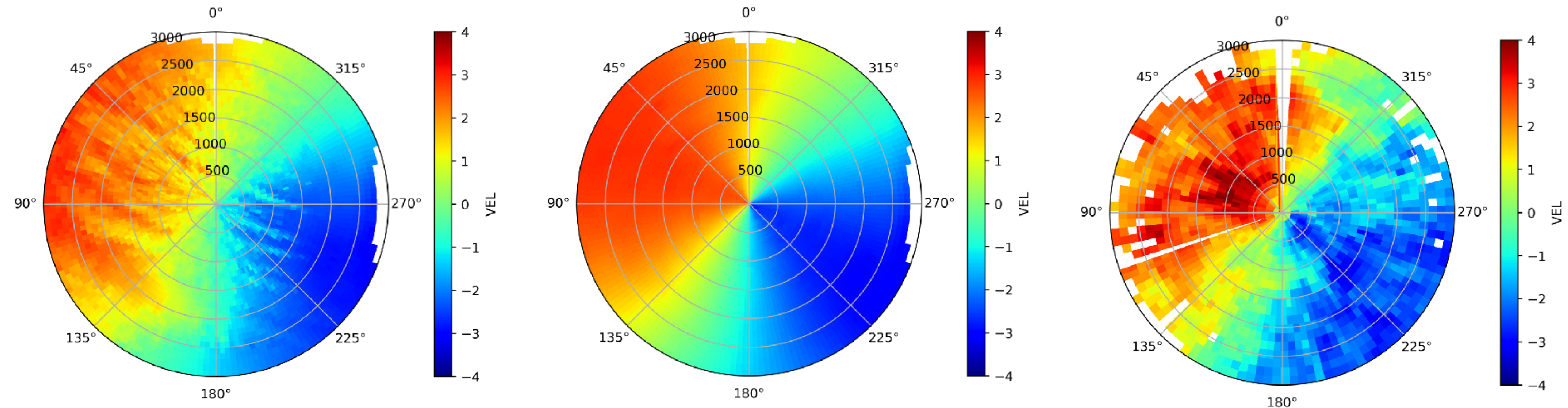


Lidar (SNR>15)

観測データ

CityLBMによる風況シミュレーション

視線方向速度の比較@15:00JST



CityLBM (LES)

CReSS (RANS)

Lidar (SNR>15)

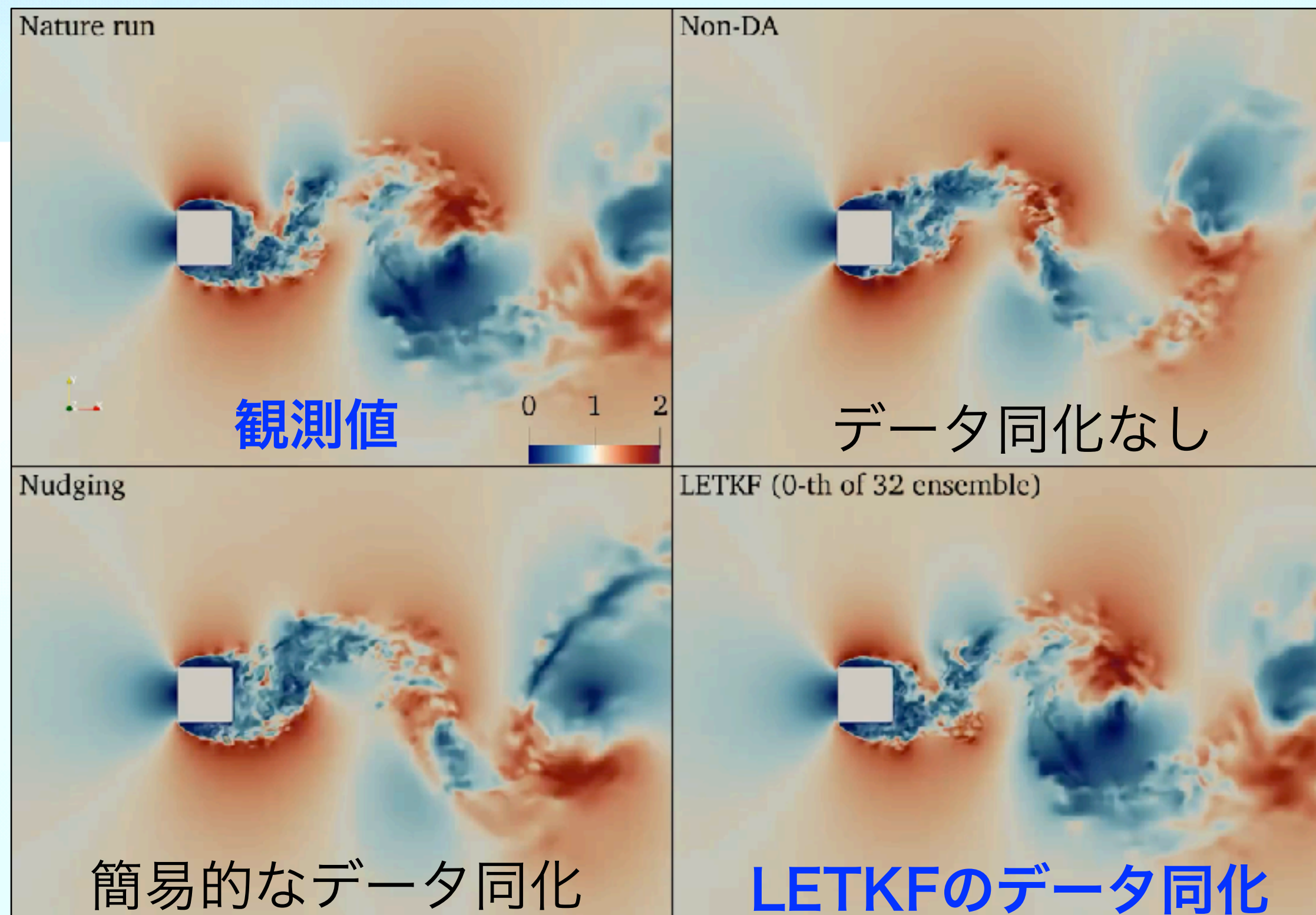
全体的な視線方向速度は、CReSSに従うが、細かな乱流構造の表現が可能
➡精度が悪化する風況条件も存在するため、街区内の観測値の同化が必須

複雑な建物間の風況予測に向けたデータ同化技術

局所アンサンブル変換カルマンフィルタ LETKF

- 観測量 (例、速度) Y_0 =観測、 Y =解析
- データ同化される状態量 X (f_{LBM})

➡観測できないシミュレーション値を同化



• LETKFのデータ同化式

$$X_{t|t} = X_{t|t-1} + \delta X (w_{col:1 \rightarrow N_{ens}} + W' - I)$$

• 共分散行列など

$$Q = (n_e - 1)I + (\delta Y^T R^{-1} \delta Y) = V D V^T$$

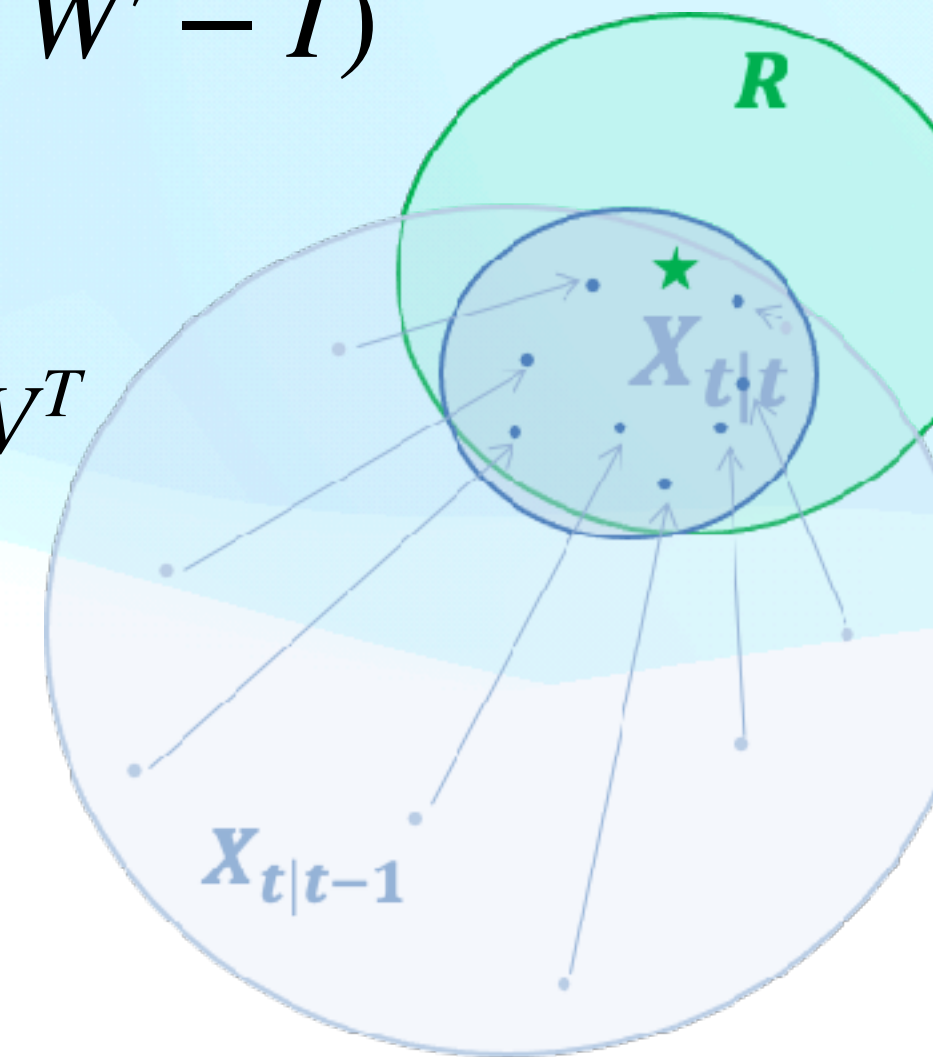
$$w = (V D^{-1} V^T) (\delta Y^T R^{-1} \delta y_o)$$

$$W' = \sqrt{n_e - 1} (V D^{-1/2} V^T)$$

• アンサンブル平均・変動

$$\delta y_o(k) = y_o - \bar{y}_{t|t-1} \quad \bar{x}_{t|t-1} = op_{mean,ens}(X_{t|t-1})$$

$$\delta X_{(k)} = X_{t|t-1}(k) - \bar{x}_{t|t-1} \quad \delta X_{(k)} = X_{t|t-1}(k) - \bar{x}_{t|t-1}$$



LETKFの適用により、非定常な乱流現象の振幅と位相を同期した解析が可能

➡ビル間や路地等の高精度な風況予測に期待

まとめと今後の研究計画

- CityLBMと東工大・防災科研の知見と組み合わせることで、現実の都市街区を模擬した高解像度の風況解析を実現
- 局所アンサンブル変換カルマンフィルタ LETKFにより、物体近傍の非定常流体現象を同化
- 今後は、東工大内に風況観測装置を設置し、観測データ同化解析を実施

