

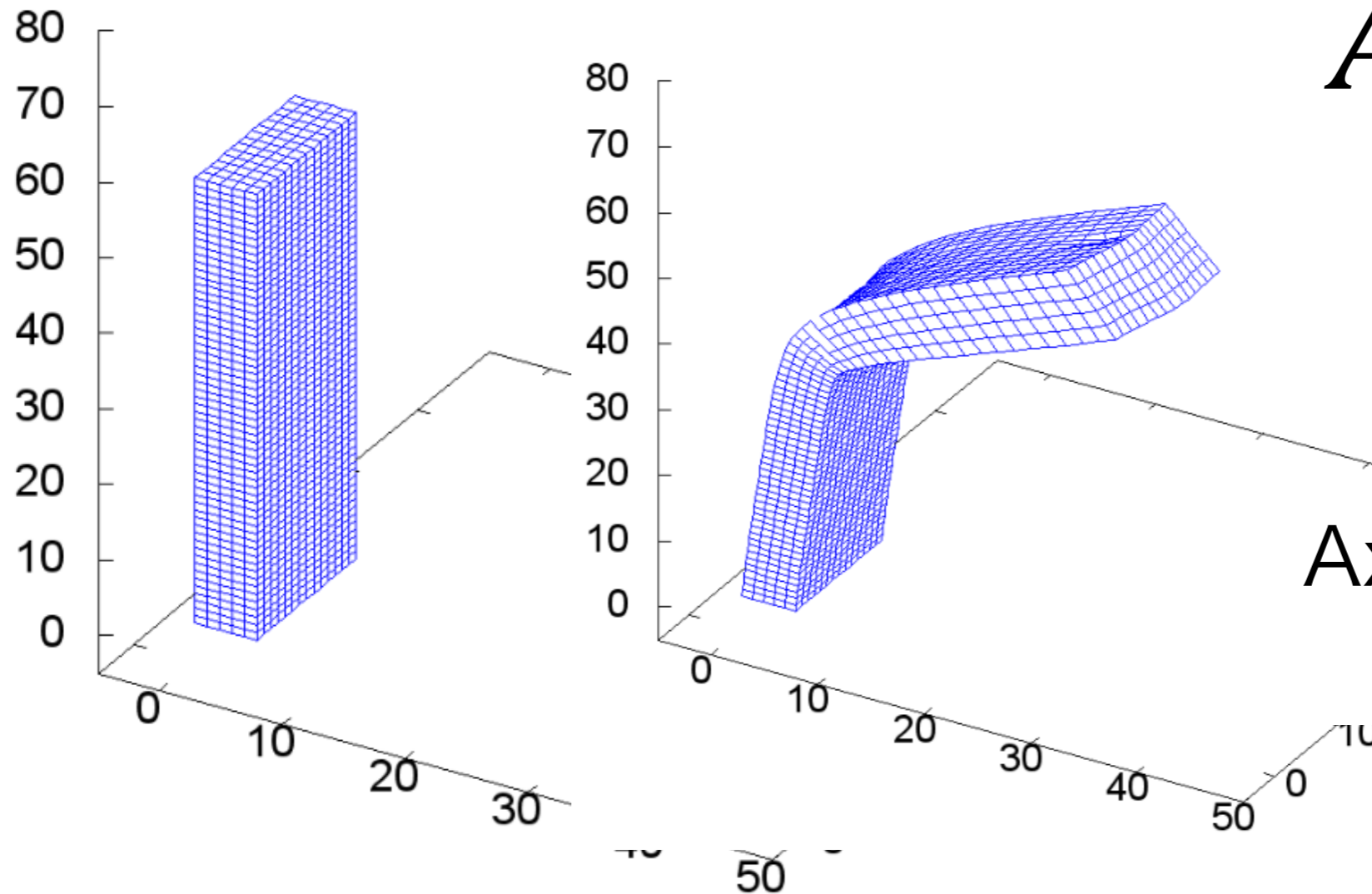
代数的多重格子法の パラメタ探索空間について

藤井昭宏、野村直也、田中輝雄（工学院大学）

SA-AMG法の有用性

- ・ 近似解に含まれる”収束しにくい成分”を粗いレベルで補正することで、収束性を改善できる
 - ・ $Ax \sim 0$ となる非ゼロベクトル x が収束しにくい代表的な成分
 - ・ PETSc, ML などはこの成分をユーザが設定できるインタフェースがある
- ・ 収束しにくい成分をうまく設定することで、解法の収束性を向上させることが可能

SA-AMG法の有用性



$$Ax = b$$

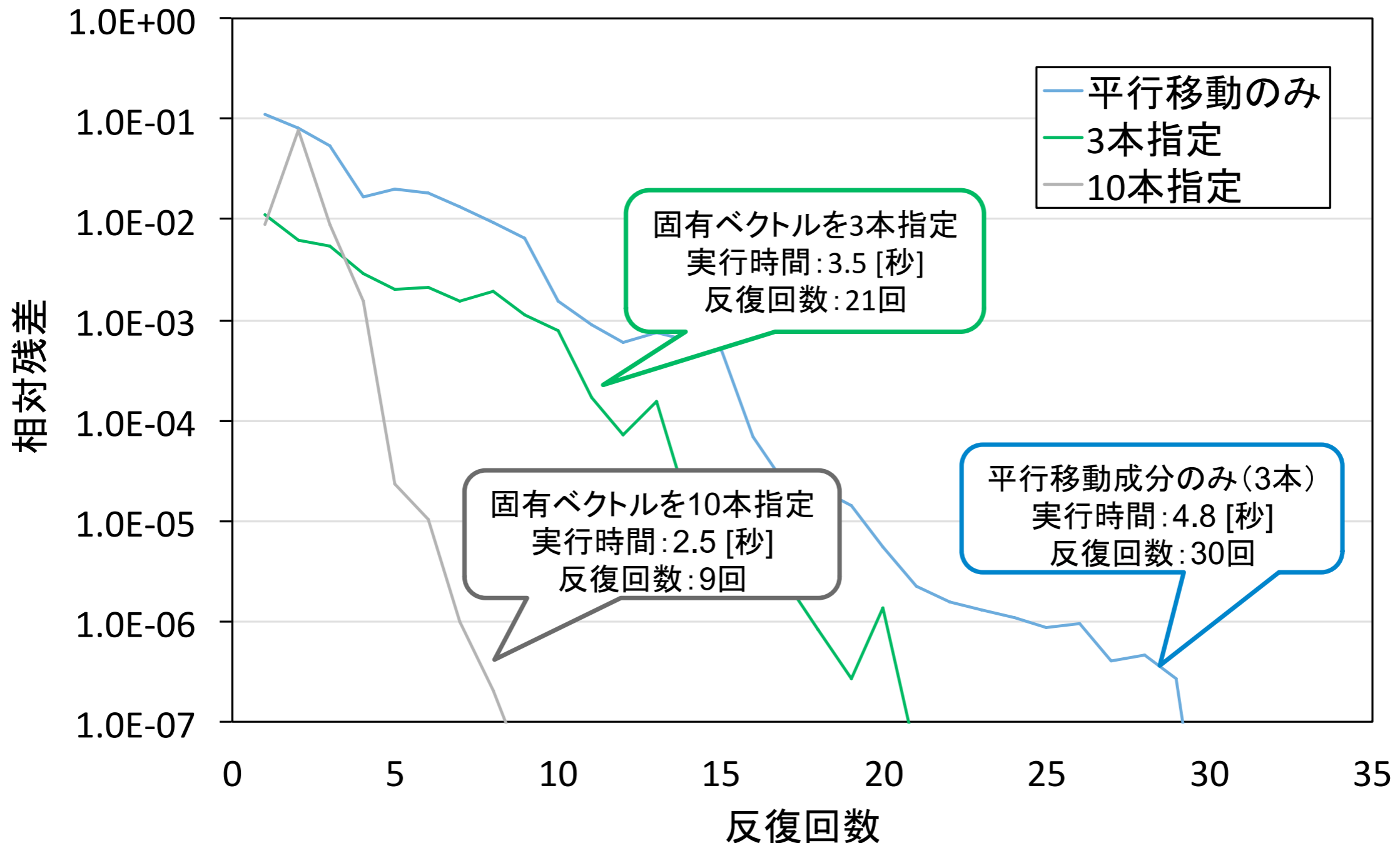
x :各点 i の変位(x_i, y_i, z_i)

b :各点 i に作用する力

(f_{xi}, f_{yi}, f_{zi})

Ax : x の変形があるときの
の応力

SA-AMG法の有用性



工学院大 M1 の野村直也が作成@東大インターン

SA-AMG法のパラメータ設定

PETScの場合

はじめのブロック幅:	MatSetBlockSize()
収束しにくい成分の設定:	MatSetNearNullSpace()
強連結成分の閾値:	PCGAMGSetThreshold()
アグリゲートの減速ヤコビの係数:	

対称正定値行列の場合 最大固有値から設定

そうでない場合 -pc_gamg_agg_nsmooths 0 と指定
減速ヤコビ係数を0に設定できる

SA-AMG法のパラメータ設定

我々の実装の場合

はじめのブロック幅: `-vector_problem=..`

収束しにくい成分の設定: `amgs_store_kernel_vecs()`

強連結成分の閾値: `-strong_con_threshold=..`

アグリゲートの減速ヤコビの係数: `-dump_jacobi_coef=..`

SA-AMG法のパラメータ設定と その課題

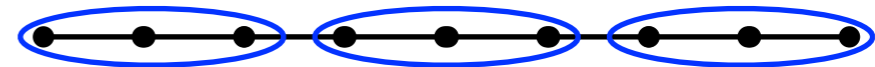
- ・ 他にも設定できるパラメータは数多い
 - ・ 各レベルでのスムーザの種類やその反復回数、加速係数など
 - ・ PETScでは（マニュアルからは）ユーザから変更できるかどうかは不明
- ・ **問題**：収束に影響を与える重要なパラメータが多すぎて、収束しないのがパラメータ設定のためなのか、解法が対応できないのか、判別が難しい。
 - ・ 収束しにくい成分とその本数 このプレゼンで挙動を調べるパラメータ
- ・ レベル生成に関わる重要なパラメータのレベルごとの設定
 - ・ 強連結成分の閾値とアグリゲートの緩和係数

行列生成部のパラメタ

1. 行列Aを対角要素を基準にフィルタリング
グラフの簡素化 強連結成分の閾値

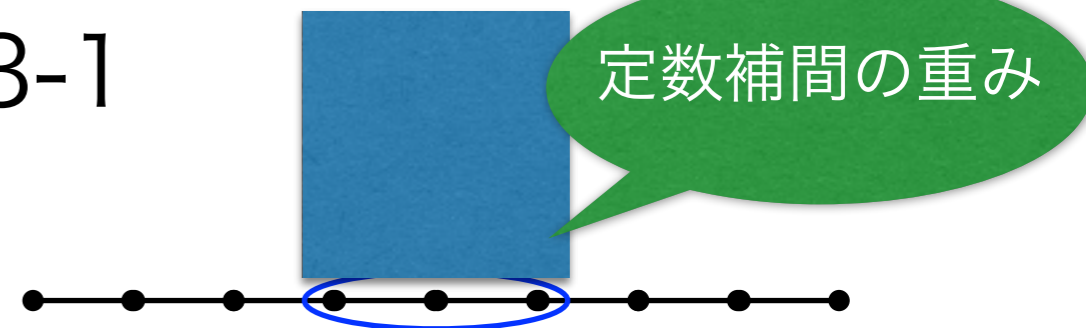


2. アグリゲート生成：独立な未知数集合



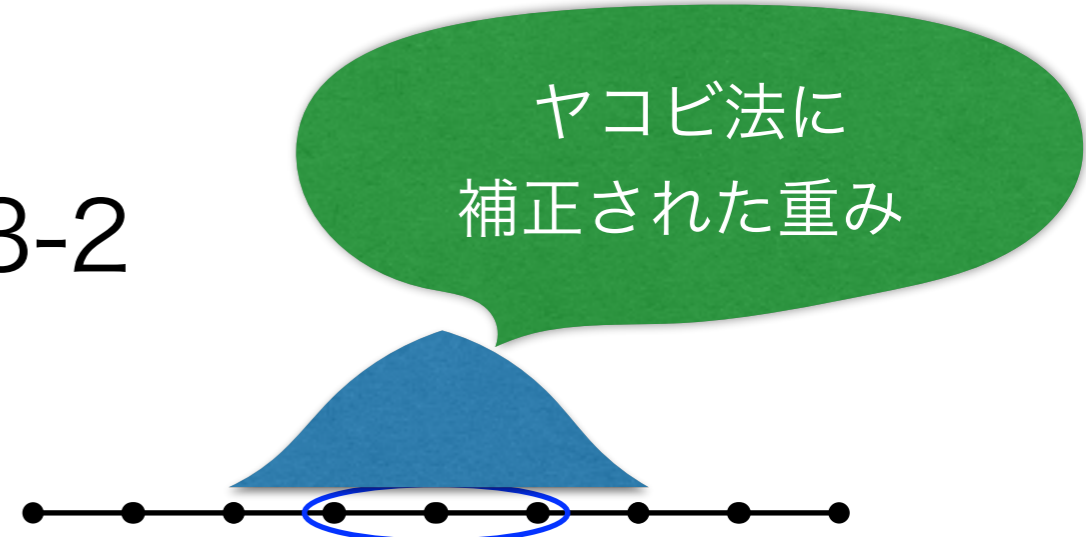
3. フィルタリングされた行列Aによる
アグリゲートごとの減速ヤコビ緩和法
減速係数

3-1



4. この重みを列ベクトルとして行列P
を生成し、 $P^t AP$ を計算することで粗い
レベルを生成

3-2



主な反復解法部のパラメタ

- ・ 各レベルの対称ガウスザイデルスムーザの加速係数と反復回数

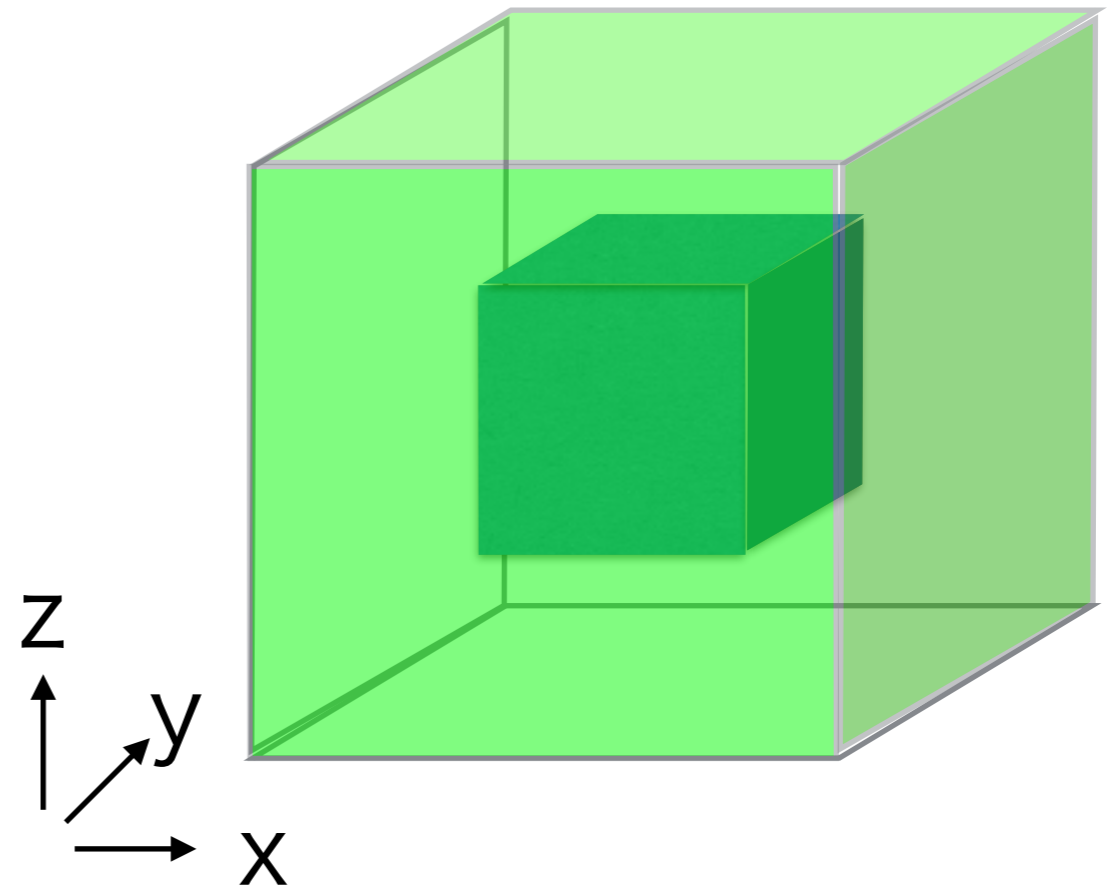
smoother iteration

smoother acceleration

収束性の変化を調べる

テスト問題

- ・ 柔らかい大きい立方体の中に固い立方体(各辺1/3倍)が入っており、上から力を加えたときの変位を求める
 - ・ ヤング率の比が2倍
 - ・ 上から一定の力で押す
 - ・ 境界条件:
 - 下の面 (xy平面)
 - 手前の面 (xz平面)
 - 左側面 (yz平面) の節点は
平面内に固定
- ・ 収束しにくい成分としては各軸に平行移動成分3本を指定し、AMG-CGソルバを適用

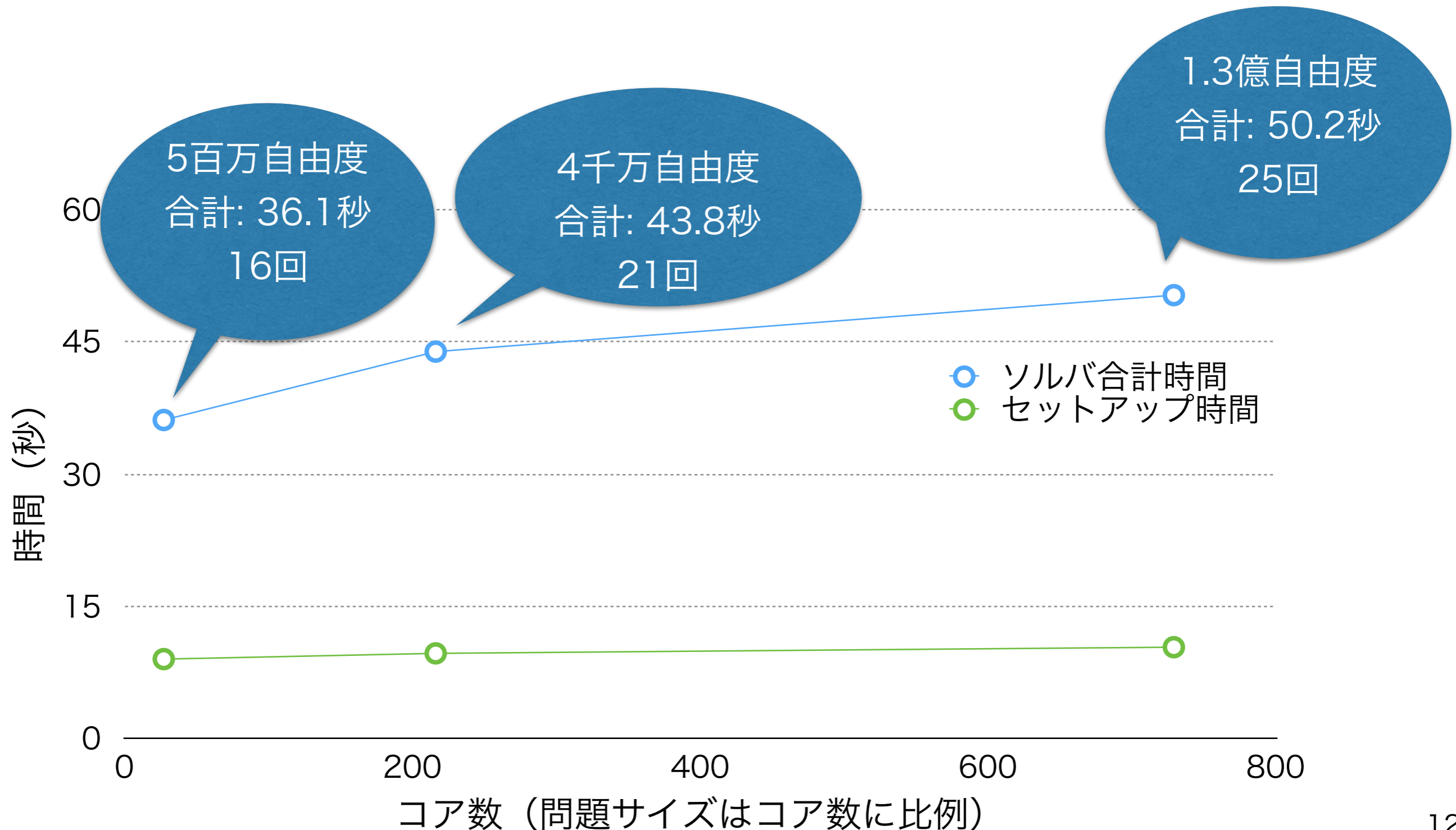


パラメタのデフォルト値と探索範囲

	強連結の閾値 theta	スムーズにする減速係数 dump	smoother iteration	smoother acceleration
Fine lev.	0.001 0.001-0.45	0.8 0.4-0.95	1	1.0
2nd fine	0.001 0.001-0.45	0.8 0.4-0.95	1	1.0
Coarser levs	0.001	0.8	2	1.0

各パラメタごとにより良い値を選び、デフォルト値を設定

Weak scaling (最適化後)



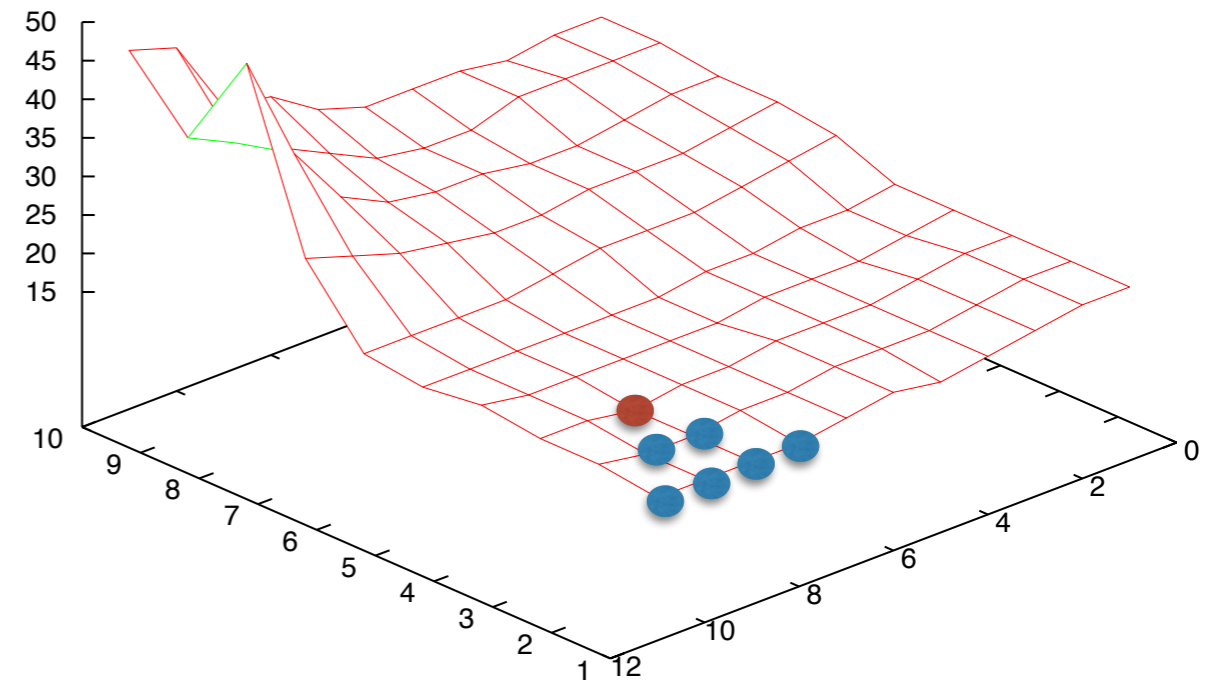
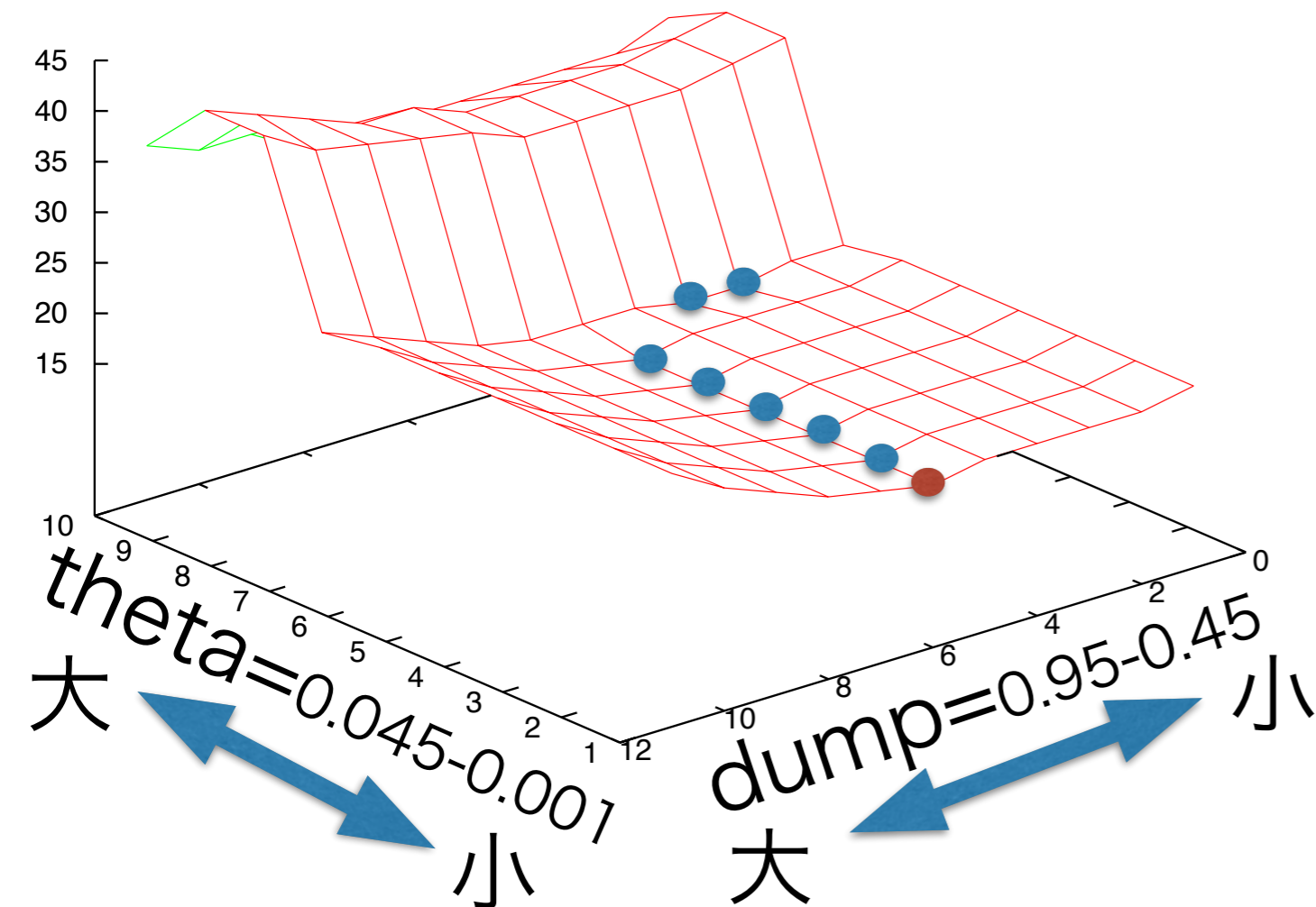
120x120x120 on 27cores

Lev. 1 最適化

'tjopt2_iter2' —

Lev. 2 最適化

'ftjopt2_iter2' —



18回
30.6秒

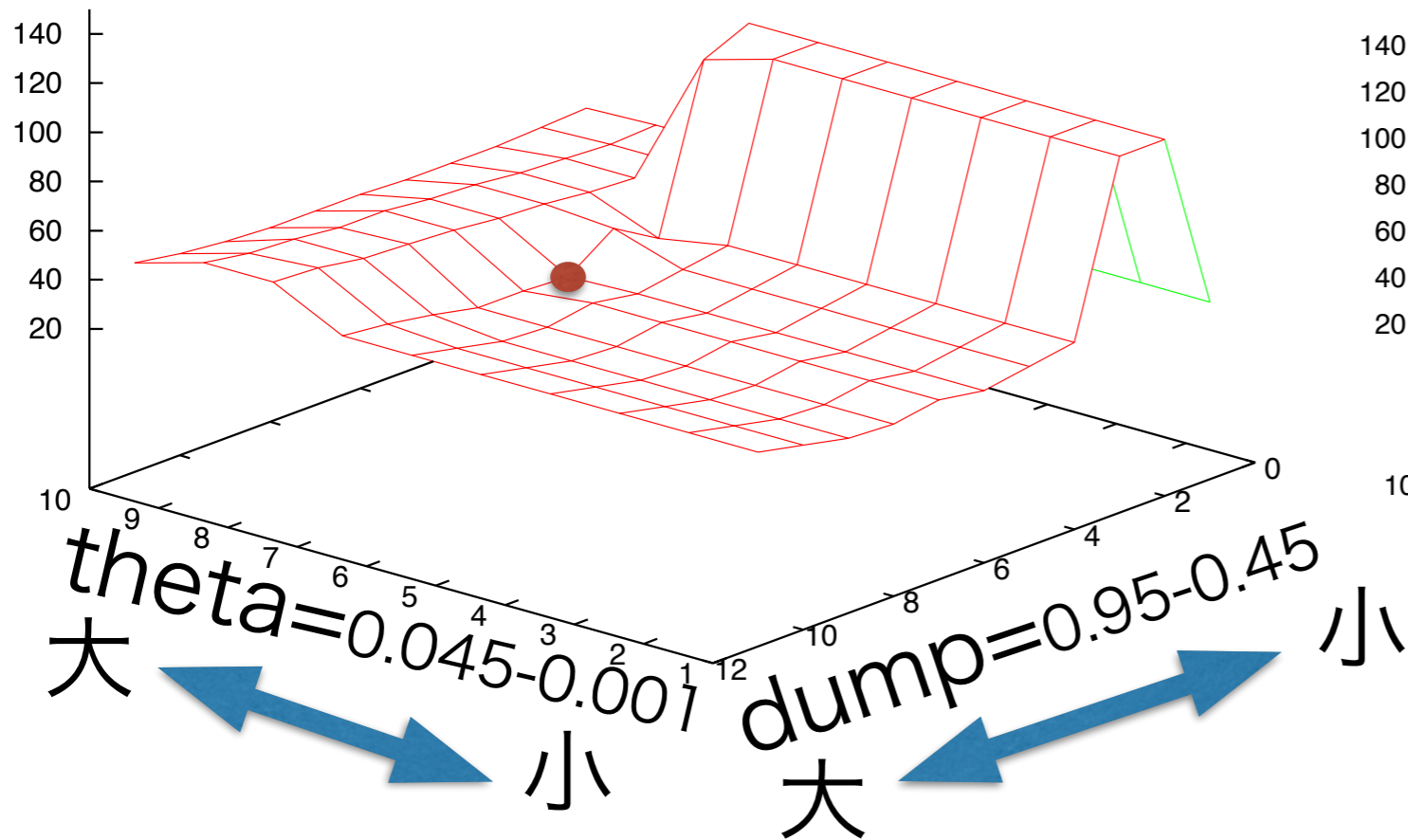
最小反復回数 16回
茶色 8.97+27.0秒

最小反復回数 16回
茶色 9.03+27.1秒

240x240x240 on 216 cores

'larg_tjopt2_iter2'

Lev. 1 最適化

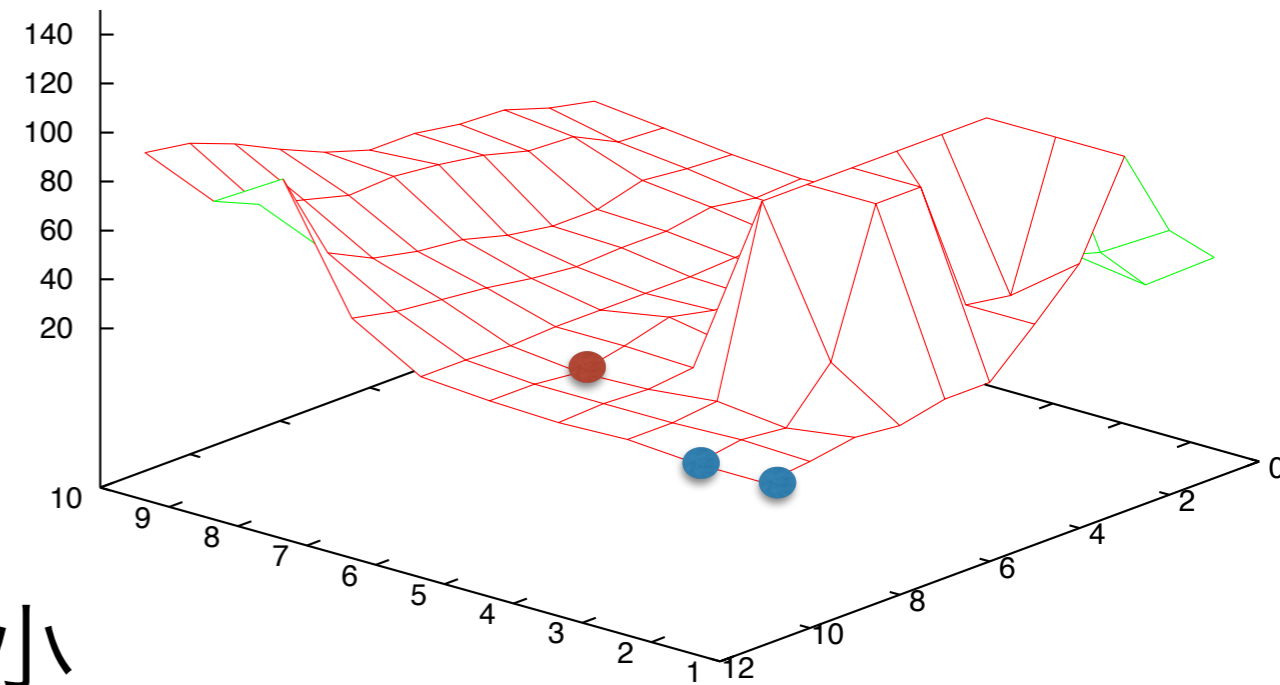


25回
40.2秒

最小反復回数 24回
茶色 : 9.74+37.8秒

'larg_ftjopt2_iter2'

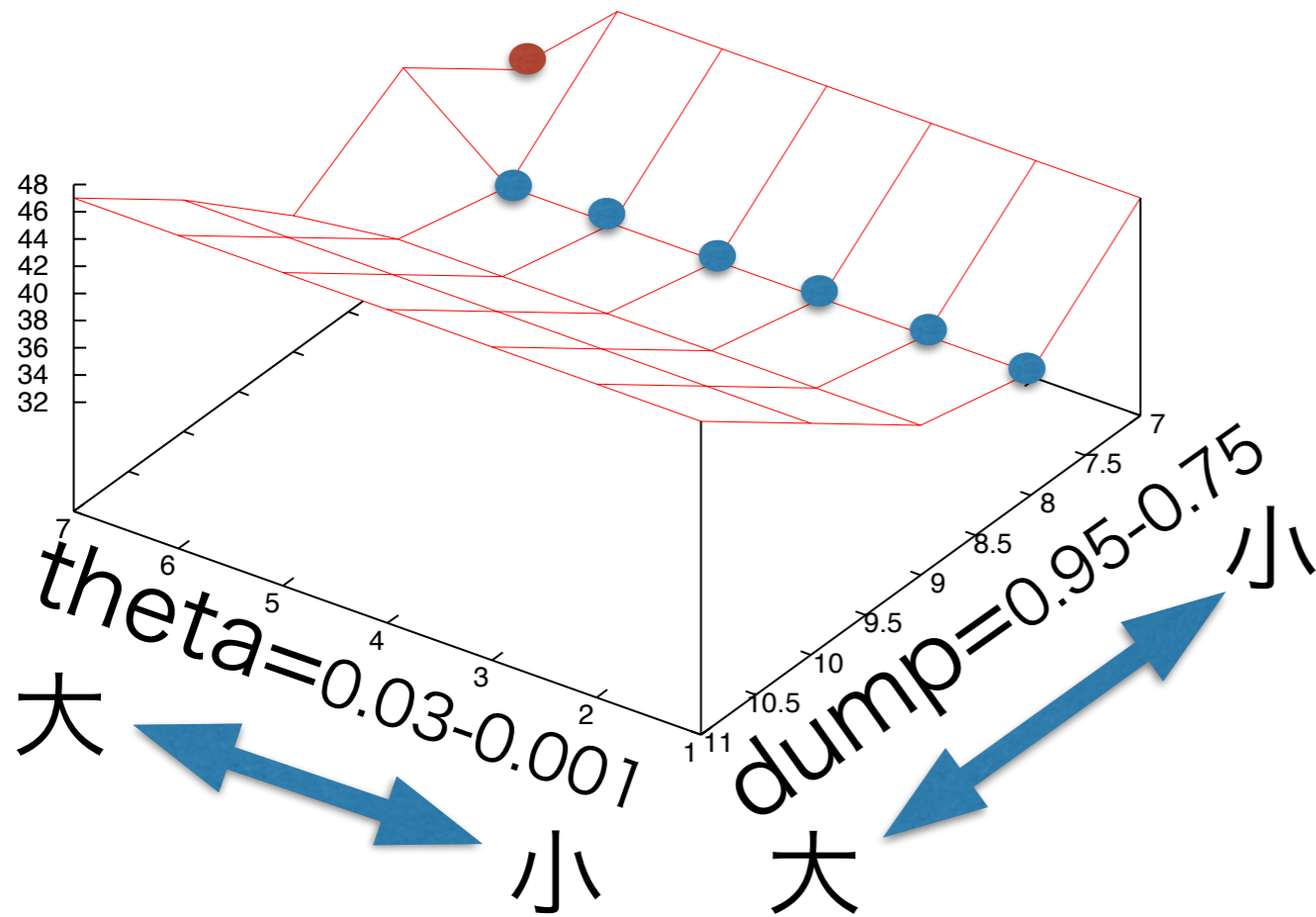
Lev. 2 最適化



最小反復回数 21回
茶色 : 9.68+34.2秒

360x360x360 on 729 cores

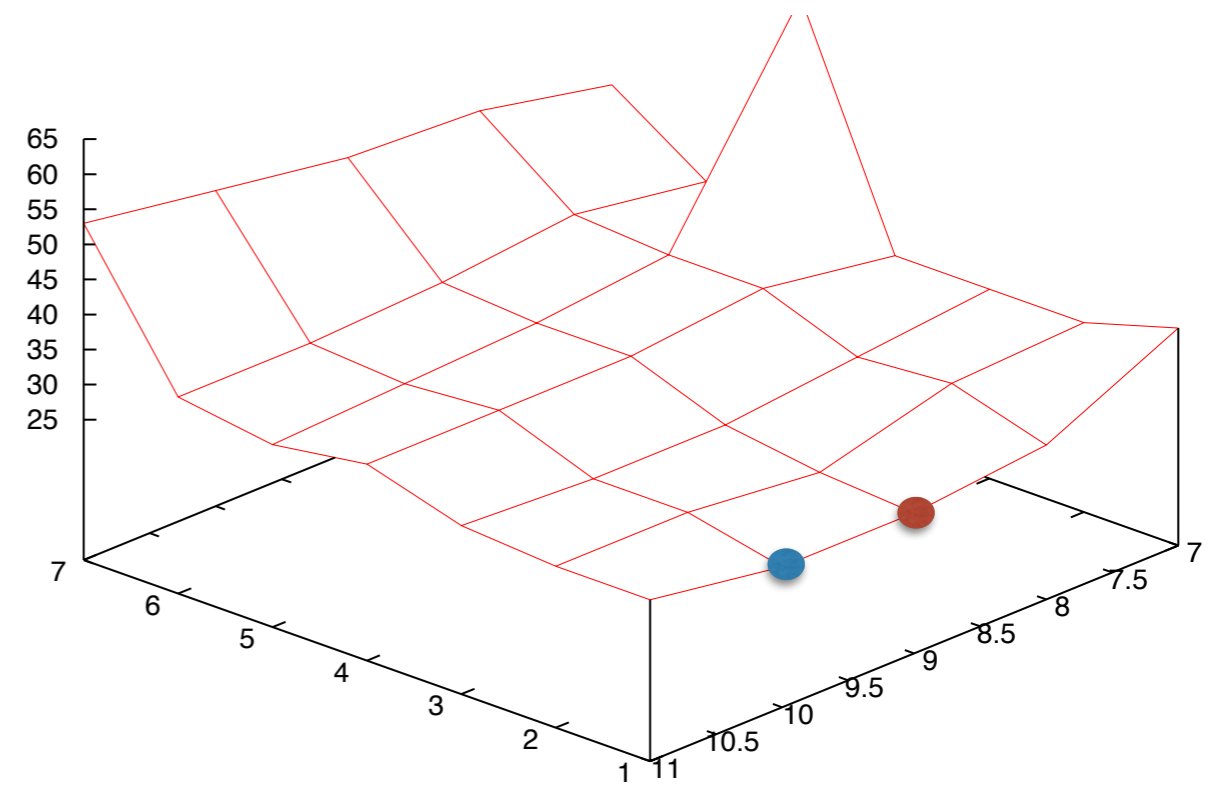
Lev. 1 最適化 !_tj_iter2'



33回
49.2秒

最小反復回数 33回
茶色 : 10.3+49.1秒

Lev. 2 最適化 !_ftj_iter2'



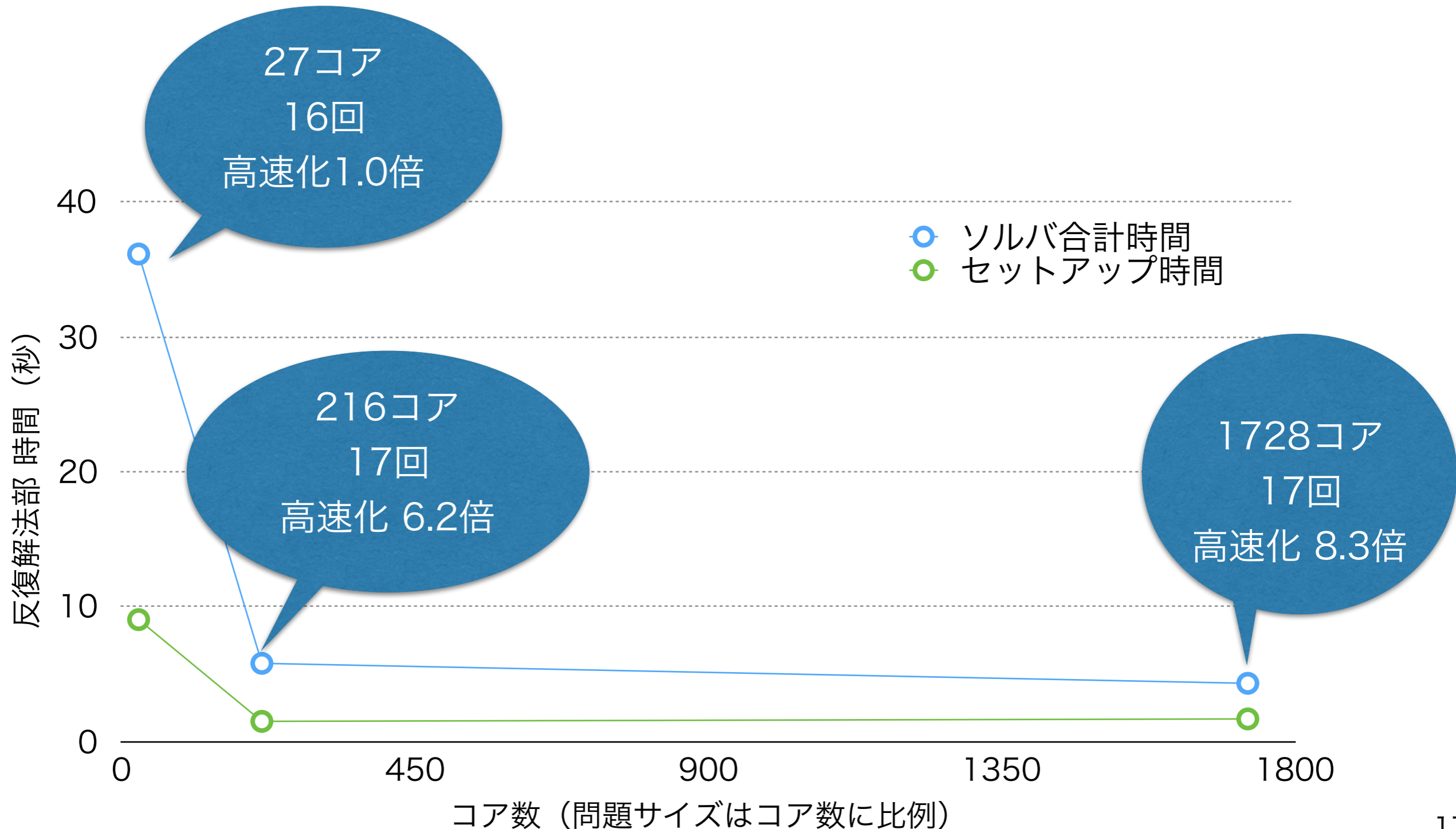
最小反復回数 25回
茶色 : 10.3+39.9秒

Weak scaling まとめ

- ・ スケーリング性能はほどほど
収束しにくい成分を3本しか設定していないため、
問題サイズの増加に伴って反復回数が増加している
- ・ 強連結成分の閾値と減速係数の特定の組み合わせにより、不安定な挙動になる場合もあった
- ・ 問題サイズが大きくなるとレベル1、レベル2で個別に最適化することで大きく改善
 - ・ レベル1で適正の値に設定をしていても、レベル2で適切な値に設定されていないと発散することもある

Strong scaling(最適化後)

518万自由度(120x120x120)



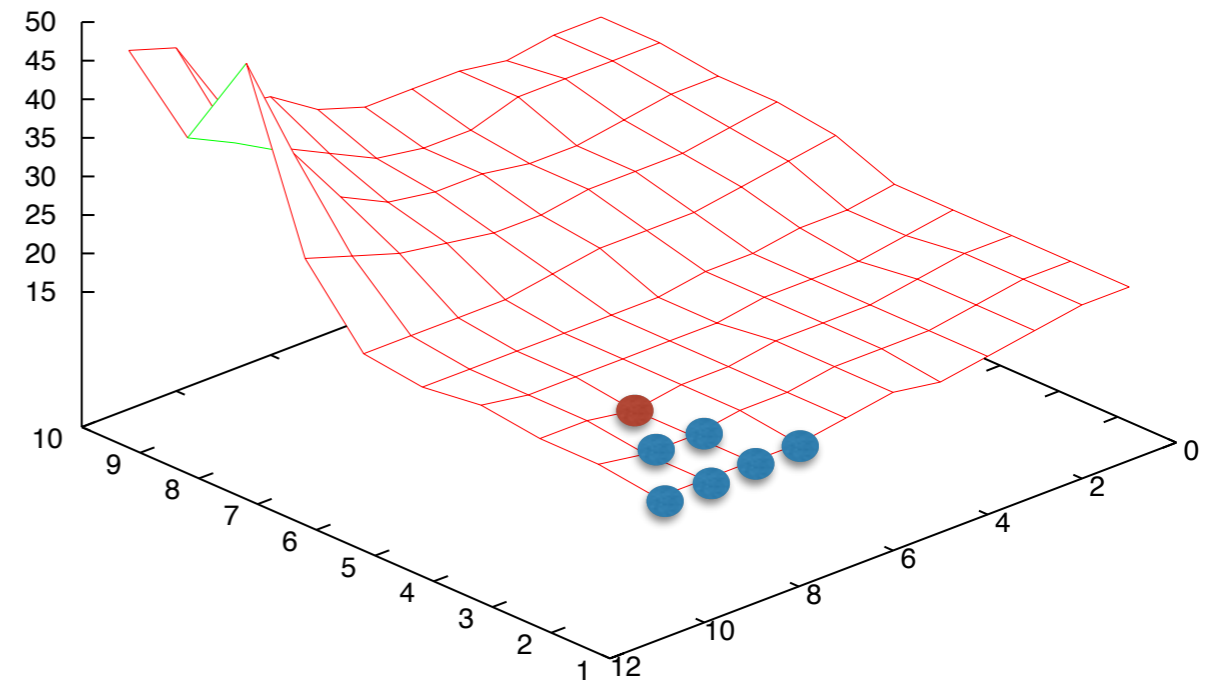
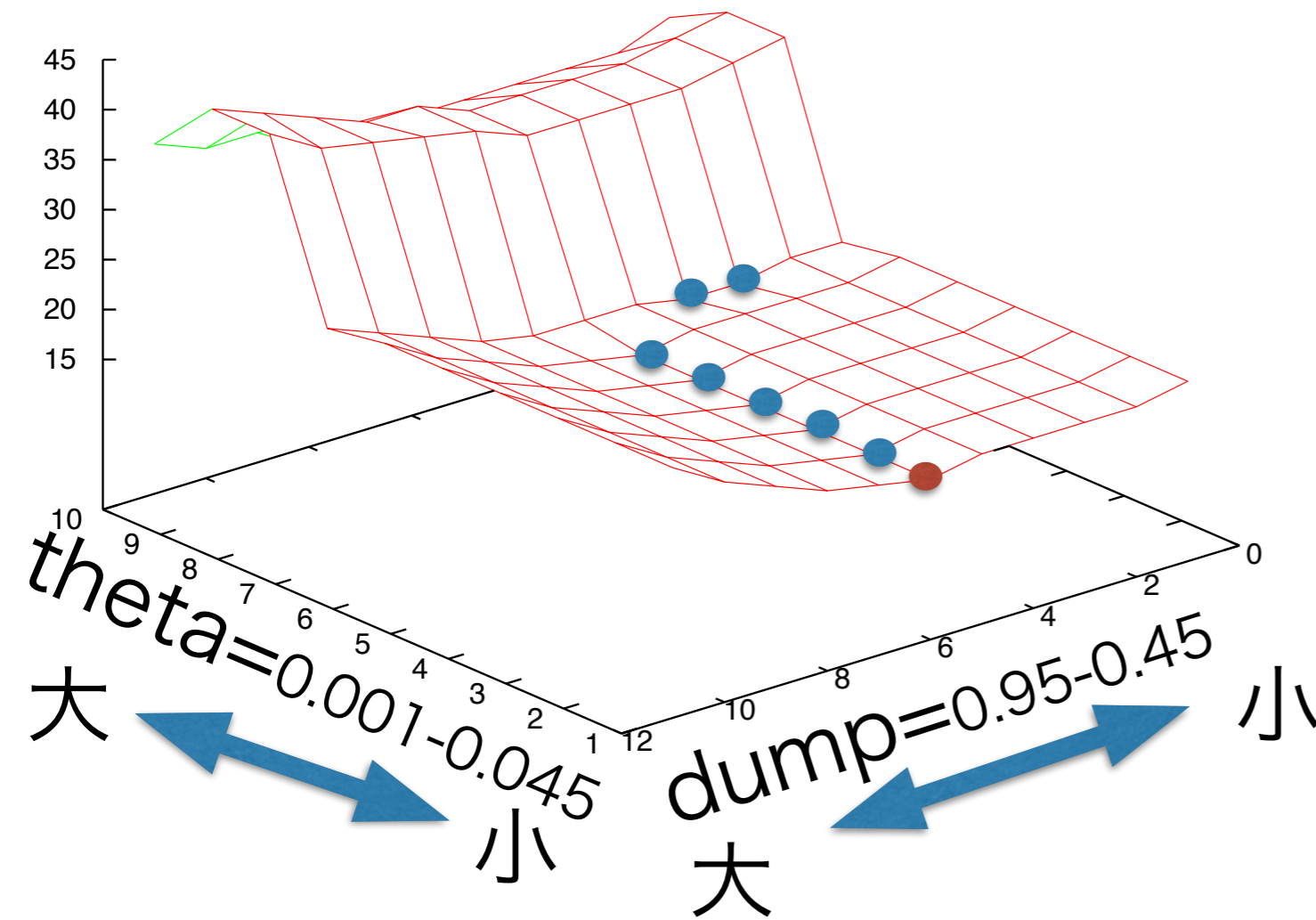
120x120x120 on 27cores

Lev. 1 最適化

'tjopt2_iter2' ———

Lev. 2 最適化

'ftjopt2_iter2' ———



最小反復回数 16回

最小反復回数 16回

18回
30.6秒

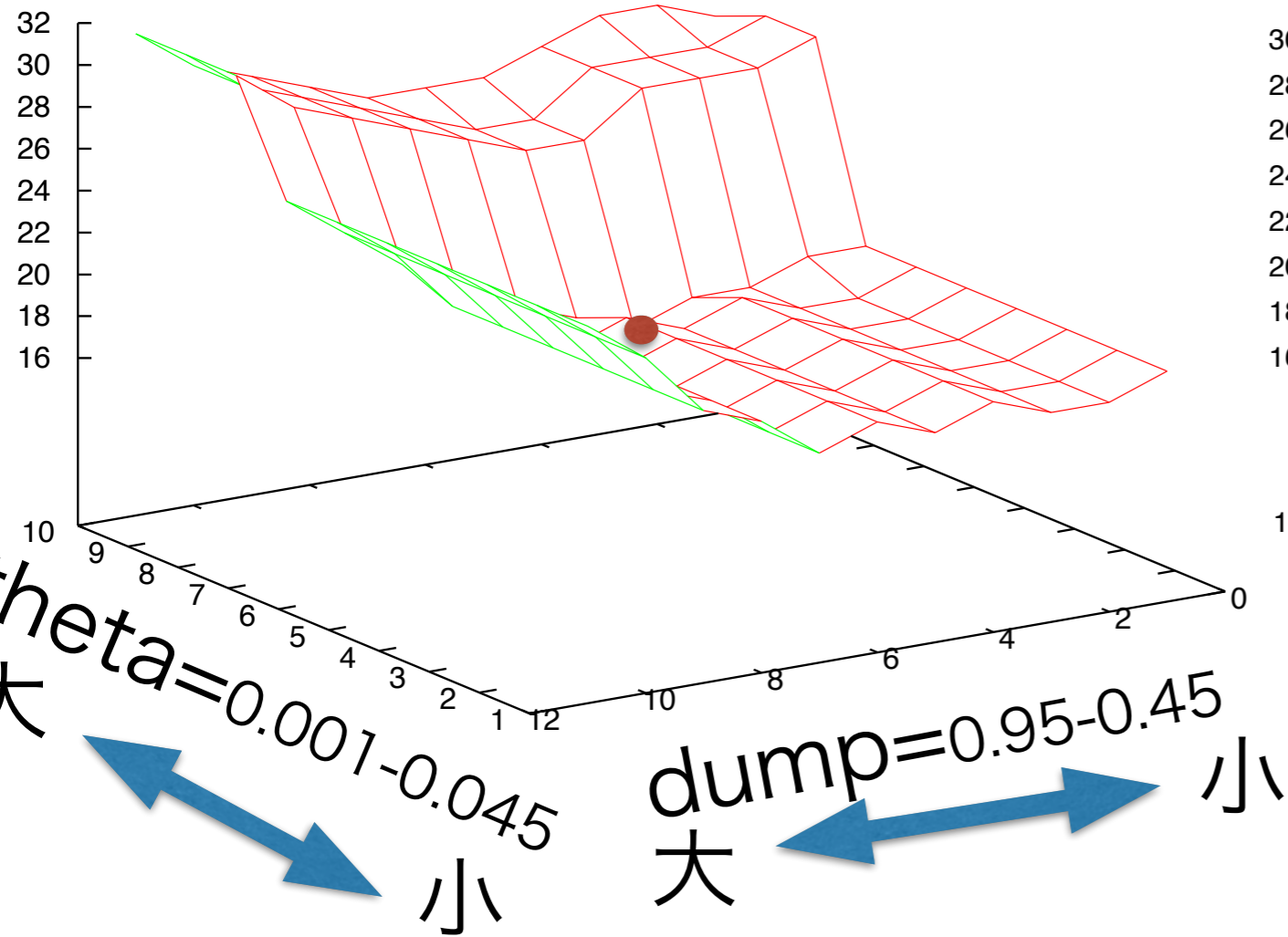
茶色 8.97+27.0秒

茶色 9.03+27.1秒

120x120x120 on 216 cores

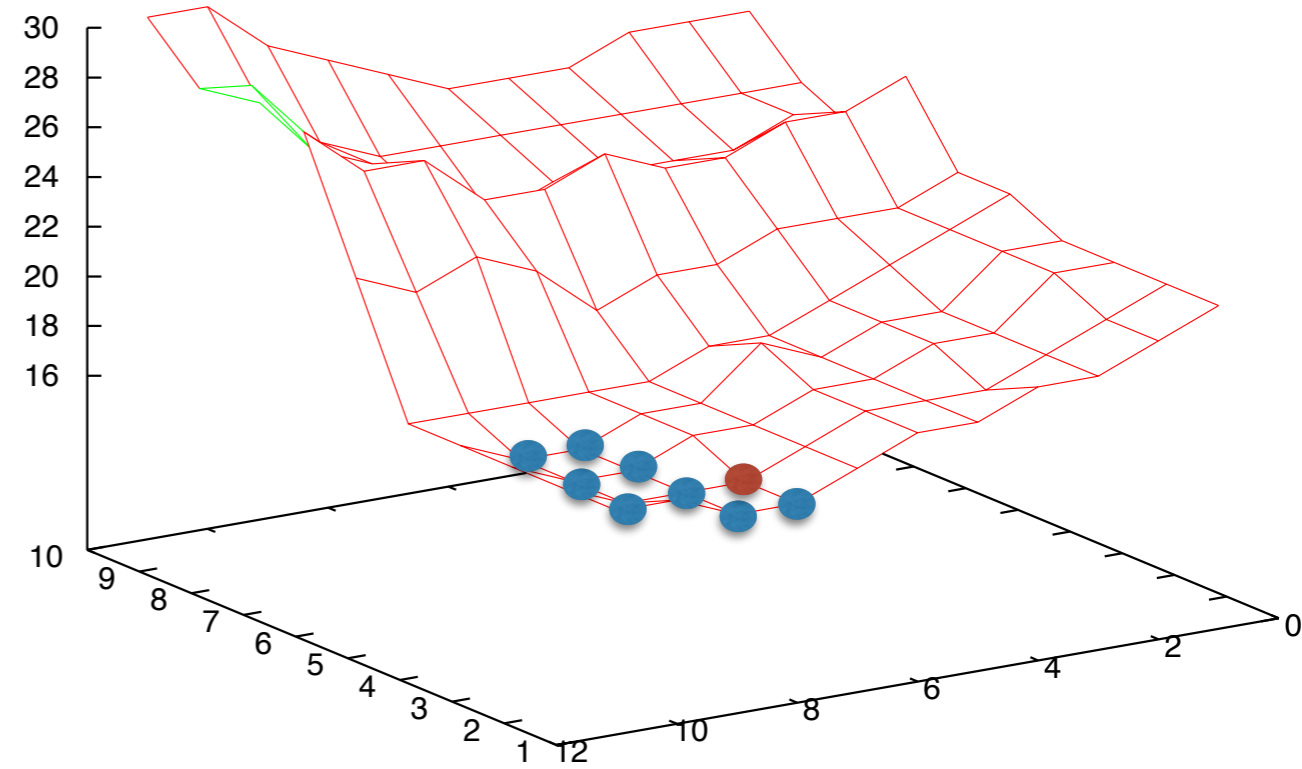
'larg_tj_strng_iter2'

Lev. 1 最適化



'larg_ftj_strng_iter2'

Lev. 2 最適化



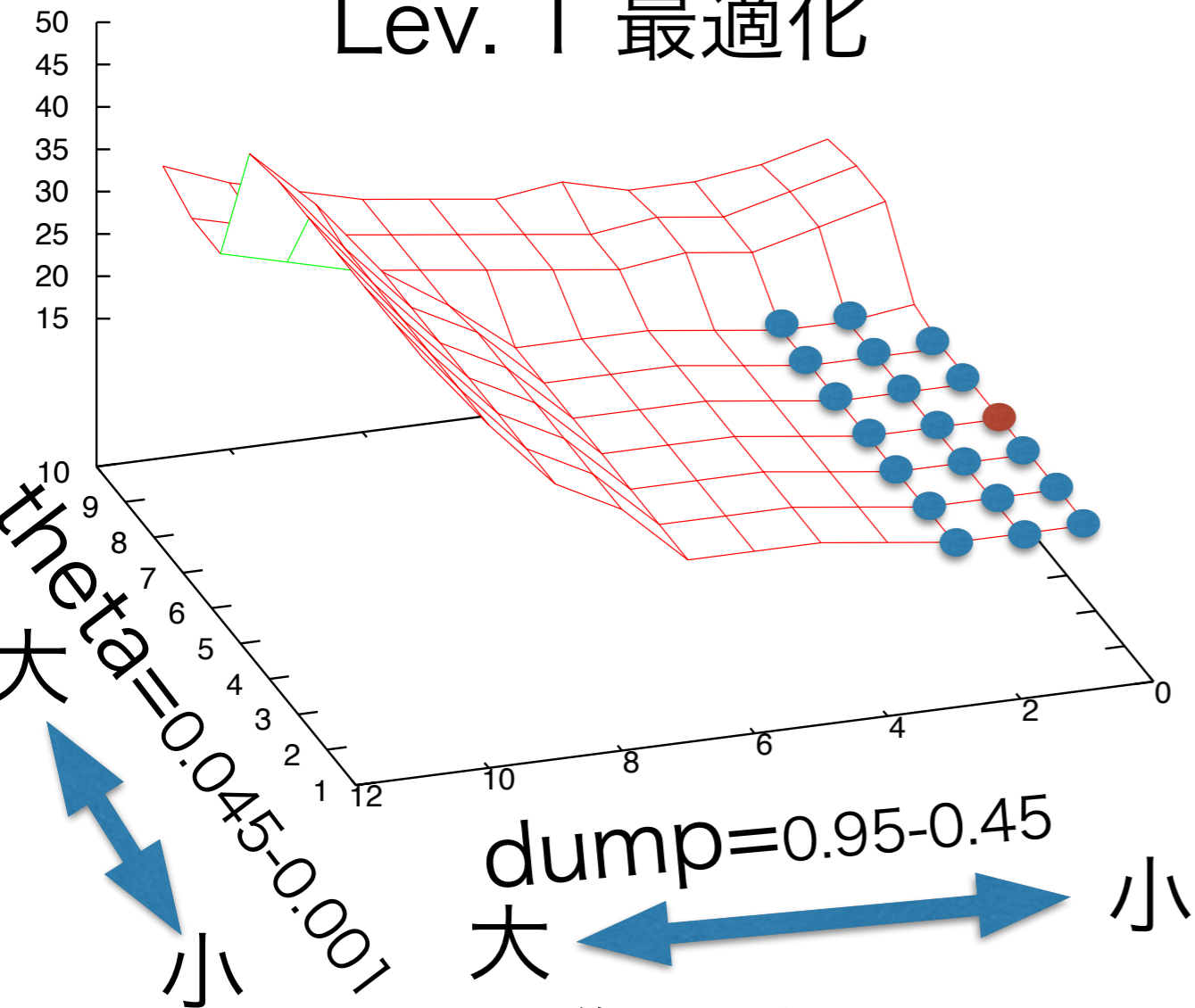
20回
4.97秒

反復回数 17回
茶色 : 1.5+4.3秒

反復回数 17回
茶色 : 1.5+4.3秒

120x120x120 on 1728 cores

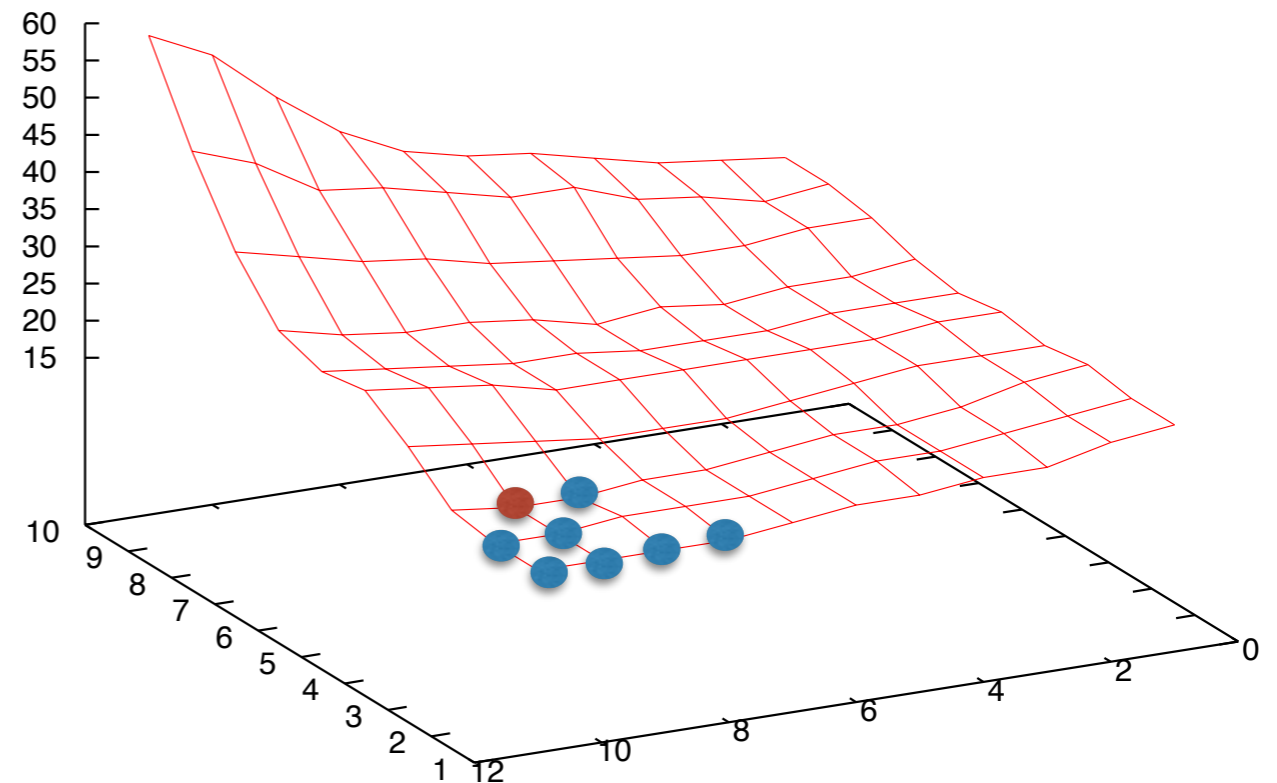
Lev. 1 最適化



26回
3.3秒

反復回数 17回
茶色：1.69+2.68秒

Lev. 2 最適化



反復回数 17回
茶色：1.67+2.64秒

Strong scaling まとめ

- ・ スケーリング性能は良い。
並列度が大きすぎても、領域集約をするため実行時間が急増しなかった。
- ・ 問題設定、サイズが同じであれば、並列度を変えても適切なパラメタの範囲は大きくは変わらなかった
- ・ レベル1, 2での最適化によってデフォルトの設定より30%程度反復回数を減らしたものもあった

まとめ

- ・ マルチレベル生成部の強連結成分の閾値とアグリゲートの減速緩和係数の調整を2レベルで順次行った
 - ・ 実験では問題サイズが大きくなるほど、この係数の調整の有効性が増した
 - ・ 問題が同一であれば、適切なパラメタ設定は並列度の変化により悪くなることは少なかった
- ・ スケーリング性能はパラメタ調整後のデータを使うと、反復回数の伸びを抑えられ、大きく改善した。
- ・ パラメタ間の自動チューニングはAMGの有用性を向上させる重要な機能となり得る。