第3楽章：グラフ演算

# 行列冪カーネル



## CBCG (チェビシェフ基底 CG) 法

for $i = 1, 2, \ldots$

$S_i = (T_{j-1}(A) r_{(i-1)k})$

疎行列ベクトル積

if $(i = 1)$ then $Q_i = S_i$

else

$B_{i-1} = \left(Q_{i-1}^T A Q_{i-1}\right)^{-1} Q_{i-1}^T A S_i$

内積

$Q_i = S_i - Q_{i-1} B_{i-1}$

$a_i = \left(Q_i^T A Q_i\right)^{-1} Q_i^T r_{(i-1)k}$

内積

$x_{ik} = x_{(i-1)k} + Q_i a_i$

$r_{ik} = r_{(i-1)k} - A Q_i a_i$



## CBCG 法

- 内積の回数は削減できた
- 疎行列ベクトル積の通信はそのままだった
  →これの通信削減が「行列冪カーネル」



## 行列冪カーネル

- 疎行列 $A$, ベクトル $\boldsymbol{x}$, 正整数 $k$ が与えられて
  第1型：$\boldsymbol{x}_k = A^k \boldsymbol{x}$
  第2型：$X_k = (A\boldsymbol{x}, A^2\boldsymbol{x}, \ldots, A^k\boldsymbol{x})$
  のいずれかを計算する
  - CBCG では第2型
- 通信の回数を減らしたい



## 注1

- 行列の冪 $A^k$ などは、作らない
  – 疎行列 $A$ に対して、$A^2$ は非零要素が増える
    → 計算量が激増

- 第1型と第2型は、計算は同じで、途中のデータを残すかどうかだけが違う

第1型：$x_k = A^k x$
第2型：$X_k = (Ax, A^2x, \ldots, A^k x)$



## 注2

- CBCG 法ではチェビシェフ多項式を使う

$$\boldsymbol{s}_{ij} = T_{j-1}(A)\boldsymbol{r}$$

が通信・計算は単純な冪カーネルとほぼ同一

| 行列冪カーネル | チェビシェフ基底 |
|---|---|
| $\boldsymbol{x}_0 = \boldsymbol{x}$ | $\boldsymbol{s}_{i1} = \boldsymbol{r}$ |
| $\boldsymbol{x}_1 = A\boldsymbol{x}$ | $\boldsymbol{s}_{i2} = \eta A\boldsymbol{r} + \zeta\boldsymbol{r}$ |
| $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$ | $\boldsymbol{s}_{ij+1} = 2\eta A\boldsymbol{s}_{ij} + 2\zeta\boldsymbol{s}_{ij} - \boldsymbol{s}_{ij-1}$ |

– 違いは、スカラー倍やベクトルの加減算のみ

# 目標

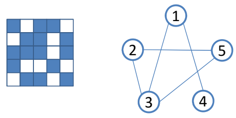
- 時空間ブロック化の手法を参考にして、
  行列冪カーネルの実装方法を設計し、
  通信回数を削減したい



# 本研究での仮定

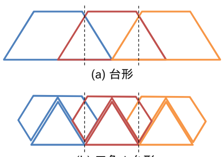
1. 非零要素は対称な位置にある
   $$a_{ij} \neq 0 \quad \Leftrightarrow \quad a_{ji} \neq 0$$
2. 対角要素は非零である
   $$a_{ii} \neq 0$$

- このとき、疎行列と無向グラフが対応する






# 時空間ブロック化



(a) 台形

(b) 三角＋台形

(c) 菱形

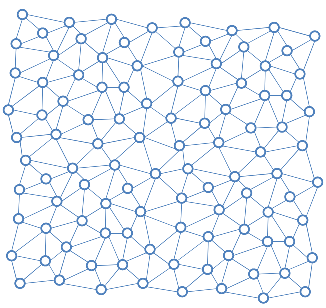
| 通信回数 | 計算重複 | 一般化 |
|---|---|---|
| 1 | 大 | 済 |
| 1 | 中 | 済 |
| 複数 | なし | 未<br>本研究 |



「台形」方式による
行列冪カーネル

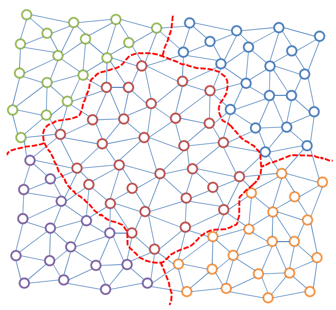


# 疎行列に対応するグラフ

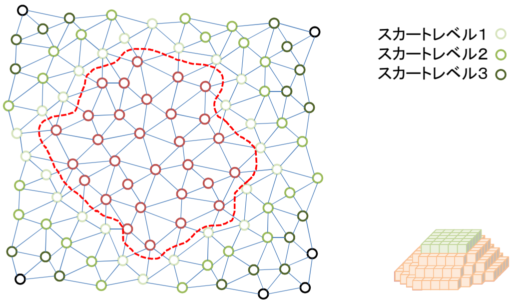




# 領域分割

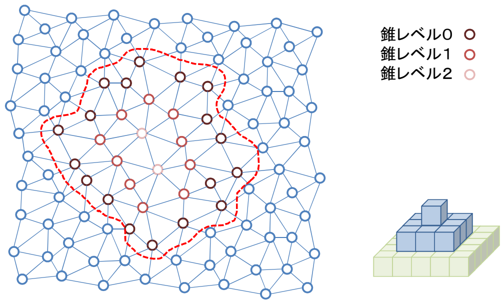


## スカートの深さ



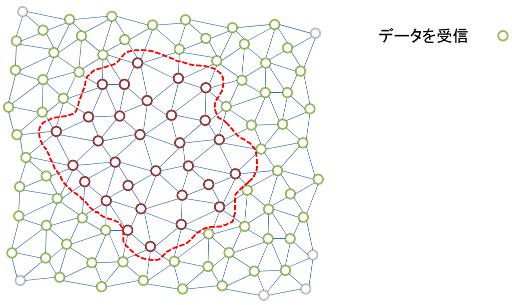

17

## 錐の高さ



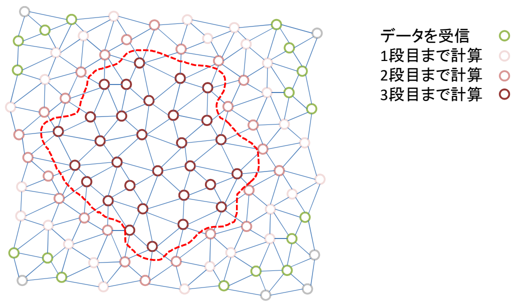

18

## 台形方式3段




19

## 台形方式3段




22

## 「台形」行列冪カーネル

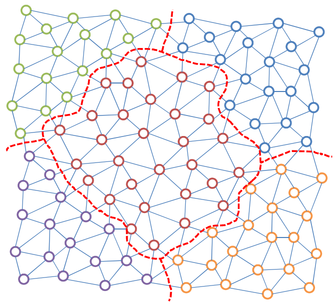
- あらかじめ，スカートレベルを計算しておけばよい

23

## 「三角＋台形」方式による行列冪カーネル

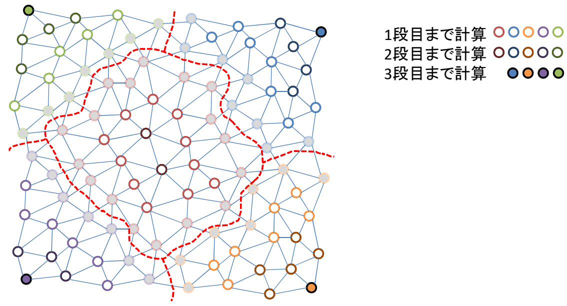
24

## 領域分割



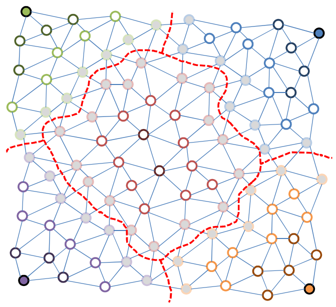
25

## 三角＋台形：ステップ1



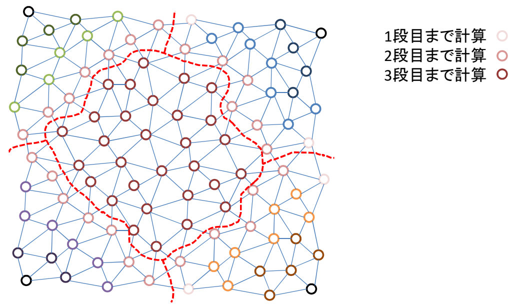

28

## 必要な通信をした上で・・・



29

## 三角＋台形（3段）：ステップ2




32

## 「三角＋台形」冪行列カーネル

- 事前に，スカートレベルと錐レベルを計算しておけばよい
  – 重複が避けられる計算を判定し，通信の必要な範囲も求めておく必要

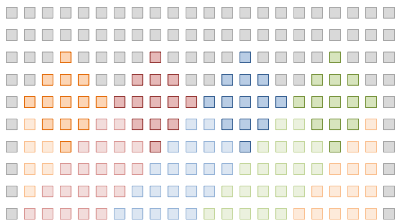
33

## 「菱形」方式による行列冪カーネル

34

## 菱形は自明でない

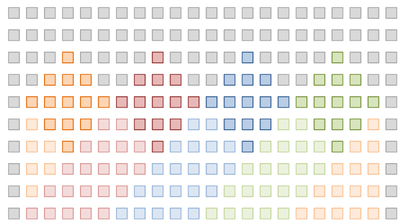
- ステンシルでは、担当領域が斜め(例:右下)方向に徐々にずれていく
  - 一般の疎行列では「方向」が定義されない





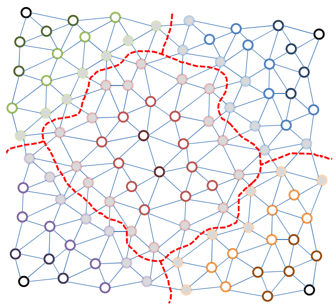
## 菱形はどうするか

- 最初は「三角」
- その次に担当範囲がずれる





## 菱形:最初の計算

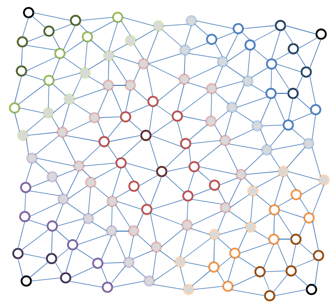


1段目まで計算
2段目まで計算
3段目まで計算

「三角」と同じ

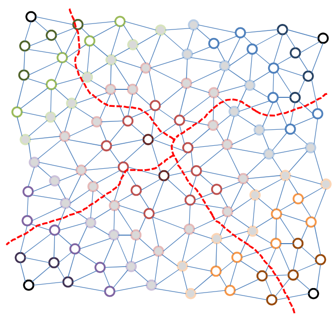


## 領域分割を一旦忘れて・・・





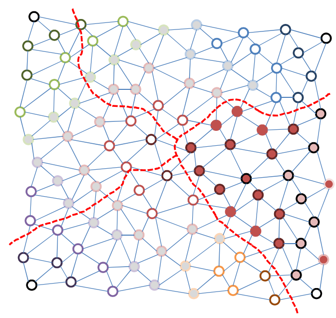
## 新しい領域分割を導入



どうやって新しい領域分割を決めるかは、後ほど説明



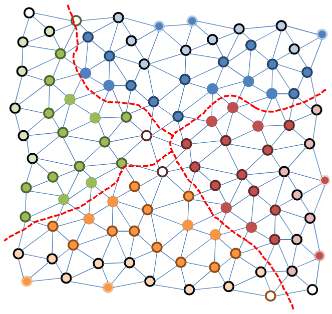
## 必要な通信をしたうえで、計算



1段目まで計算
2段目まで計算
3段目まで計算
4段目まで計算

※何段目まで計算するかの決め方は後ほど説明

## 他のプロセッサも同様

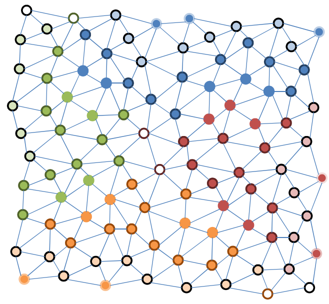


1段目まで計算
2段目まで計算
3段目まで計算
4段目まで計算

※何段目まで計算するかの決め方は後ほど説明

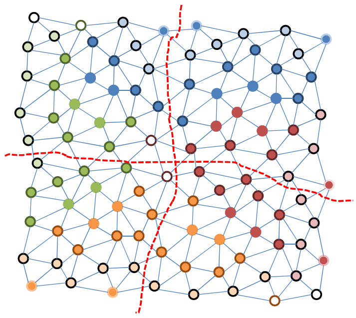
41

## また、領域分割を一旦忘れて・・・



42

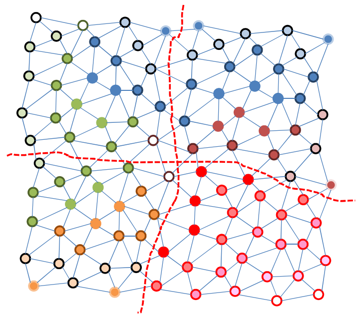
## 改めて領域分割



どうやって新しい領域分割を決めるかは、後ほど説明

43

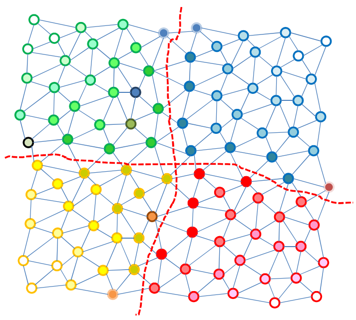
## 必要な通信をしたうえで、計算



3段目まで計算
4段目まで計算
5段目まで計算
6段目まで計算
7段目まで計算

44

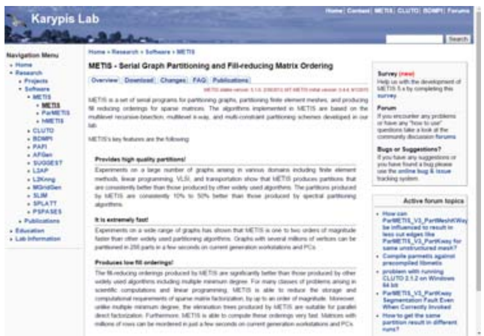
## 他のプロセッサも同様に



3段目まで計算
4段目まで計算
5段目まで計算
6段目まで計算
7段目まで計算

47

## グラフ分割ソフト：例えば METIS




60

## 既存のグラフ分割ソフトを利用

- 通常、節点と辺に「重み」をつけられる
  - 各領域の節点の重みの和はだいたい同じで
  - 領域をまたぐ辺の重みの和をできるだけ小さく
  
  するような領域分割をしてくれる

61

## 既存のグラフ分割ソフトを利用

- 通常、節点と辺に「重み」をつけられる
  - 各領域の節点の重みの和はだいたい同じで
  - 領域をまたぐ辺の重みの和をできるだけ小さく
  
  するような領域分割をしてくれる

- 辺が接する点の計算段数が低いとき、辺の重みを重くすればよい
  - 計算段数が低い点は境界に来にくくなる

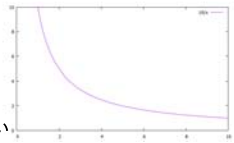
62

## テスト

- 100 x 100 の 5 点差分格子メッシュ
- グラフ分割：METIS
- 辺の重み

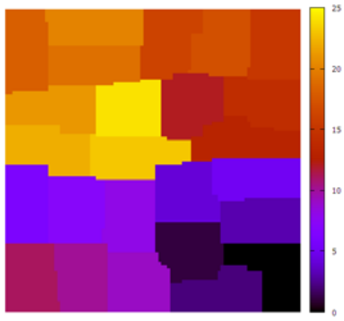
$$w = \frac{10^6}{l_0 + l_1 - 2l_{min} + 1}$$

  - $l_0, l_1$: 辺の両端の頂点の段数
  - $l_{min}$: 頂点の段数の最小値



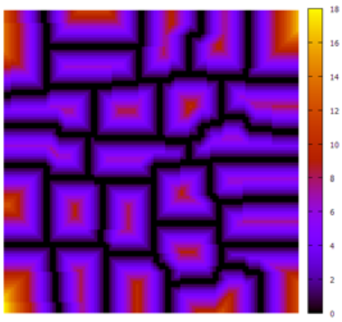
※学生君によれば、指数関数的に減少させたほうがよい

## 最初の領域分割

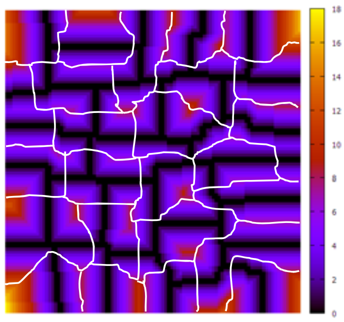


64

## （通信しないで計算できる）錐



## 2つめの領域分割



# 行列冪カーネルのまとめ

- 「台形」方式、「三角＋台形」方式は、一般の行列冪カーネルに単純に拡張できる
  - 計算の重複が多いのが問題
- 「菱形」方式の行列冪カーネルへの拡張方法を考案した
  - 領域分割をやりなおす：METIS を使ってできる
  - 計算重複を効果的に削減できた
- 冪カーネルが有効でない行列もある（当然）

103

# 関連研究


- M. Hoemmen, **Communication-Avoiding Krylov Subspace Methods**, Ph.D thesis, UC Berkeley, 2010.
- I. Yamazaki, et al., **Domain decomposition preconditioners for communication-avoiding Krylov methods on a hybrid CPU/GPU cluster**. Proc. SC 2014: 933-944
- 南武志, 岩下武史, 中島浩：**冗長な計算を伴わない 3 次元 FDTD 法の時空間タイリング グ**, 情報処理学会論文誌ACS, Vol. 6, No. 1, (2013), pp. 56–65.
- T. Muranushi and J. Makino, **Optimal Temporal Blocking for Stencil Computation**, Procedia Computer Science, Vol. 51, 2015, pp. 1303-1312 (Proc. ICCS 2015).


104

# 研究発表


- 須田礼仁，本谷徹，**チェビシェフ基底共役勾配法**，情報処理学会HPCS 2013, 2013
- 須田礼仁ほか、**数値的に安定性な通信削減クリロフ部分空間法**, 第19回計算工学講演会、2014
- 熊谷洋佑ほか，**超高並列環境での通信削減を目的としたChebyshev基底共役勾配法の特性評価**，情報処理学会 145 回 HPC 研究会，2014
- 渡邉大地，須田礼仁，**通信削減共役勾配法における基底ベクトル拡大数の選択**，第20 回計算工学講演会，2015
- Y. Kumagai, et al., **Performance Analysis of Chebyshev Basis Conjugate Gradient Method on the K Computer**, Proc. PPAM, 2015
- 須田礼仁，**一般の行列冪カーネルに向けて**，日本応用数理学会2015 年年会，2015


105