

大規模時空間並列計算でのParareal法の性能評価

今村, 成吾
神戸大学大学院システム情報学研究科

飯塚, 幹夫
理化学研究所計算科学研究機構

小野, 謙二
九州大学情報基盤研究開発センター

横川, 三津夫
神戸大学大学院システム情報学研究科

<https://hdl.handle.net/2324/1916268>

出版情報 : Proceedings of the Conference on Computational Engineering and Science. 22, 2017-06-01. 日本計算工学会
バージョン :
権利関係 :



大規模時空間並列計算でのParareal法の性能評価

今村 成吾(神戸大), ○飯塚 幹夫(九州大学), 小野 謙二(九州大学),
横川 三津夫(神戸大)

謝辞

本研究は, 文部科学省ポスト「京」重点課題⑧「近未来型ものづくりを先導する革新的設計・製造プロセスの開発」—「サブ課題A 上流設計プラットフォームの開発整備と産業利用るものづくり革新」の一環として実施し、JSPS26390130 科研費の助成を受けたものです。また本論文の結果は、理化学研究所のスーパーコンピュータ「京」を利用して得られたものです(課題番号:hp160203)。

内容

- 時間並列計算法への取り組みの背景
 - (a)応用から、(b)計算環境から
- 時間並列計算法とは
 - (a)仕組み、(b)動向
- 基本的な拡散問題でのParareal法の大規模並列性能
 - (a)計算法、(b)独立に加速可能か、(c)時間並列を使うタイミング、(d)残る課題の抽出
- まとめ

時間並列計算法 (PinT) への取り組みの背景

PinT: Parallel-in-Time Integration

背景: 応用から

例: アンサンブル計算をより高度化、使いやすいものへ→膨大なパラメータ空間の探索による新現象の発見、最適設計へ寄与

計算開始

数か月

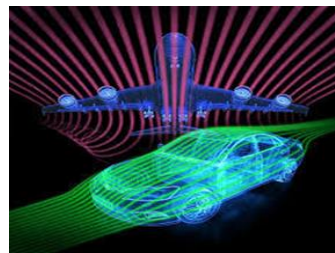
空間並列のみによるバカ並列(トコロテン式)

Ex. 数ヶ月後に1000個の結果、分析を開始: 待ち時間が無駄、計算機を占有してしまう



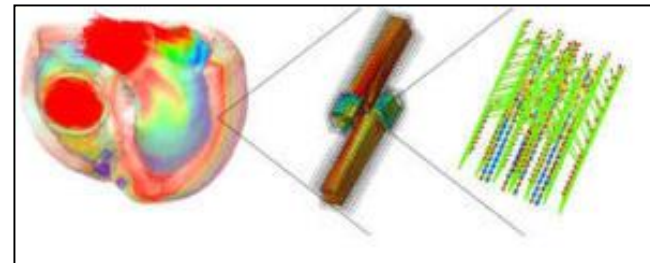
時間並列を併用し俊敏なバカ並列(機関砲式)

Ex. 短時間で次から次に結果を得る→すぐに分析→探索方向の最適化→繰り返し: 待ち時間の無駄無、計算機の柔軟な運用可能



例: ITERのダイバータ部の研究開発の加速

ITER→ダイバータ: 熱粒子からエネルギーを取り出す最も重要かつ過酷な部分⇒プラズマ負荷の軽減設計のための計算→現在は数か月(空間並列の限界)必要→これを数日に短縮して研究開発の加速に寄与したい(SIAM-PP2016/Debasmita Samaddar)

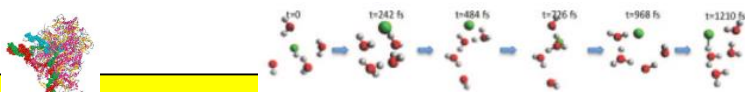


例: 第一原理分子動力学の実用化

シミュレーションの時間スケールを現状のピコ秒から材料特性の評価が可能となるナノ秒へ拡大したい→すでに、MP2(高精度計算)で14.3倍/50並列の加速を達成(Eric J. Bylaska, THE JOURNAL OF CHEMICAL PHYSICS, (2013))

例: 心臓シミュレータの臨床への実用化

短時間で治療方針を決断するためのツールとしたい(病理の解明、治療法(手術法)の効果の予測・評価)⇒そのため、解析時間を現状の数日(空間並列の限界)から数時間(時間並列を加えて)へと短縮したい(SIAM-PP2016/鷲尾)



時間方向の加速が必要な問題はたくさんあるだろう

背景：計算環境から

...

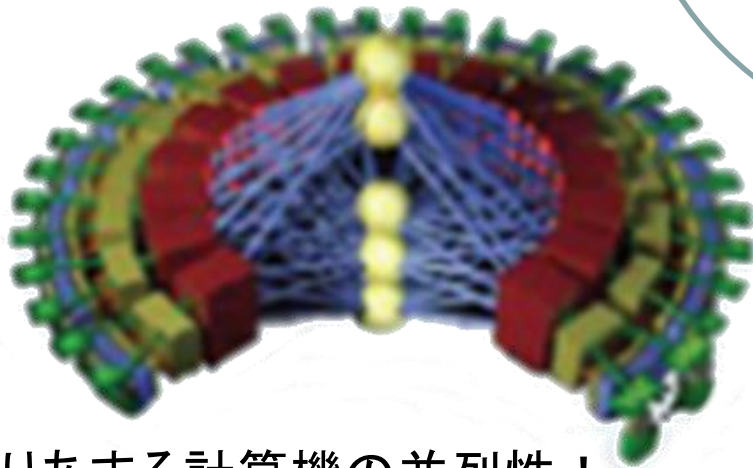
■ ビッグデータ

■ ...

■ アンサンブル計算

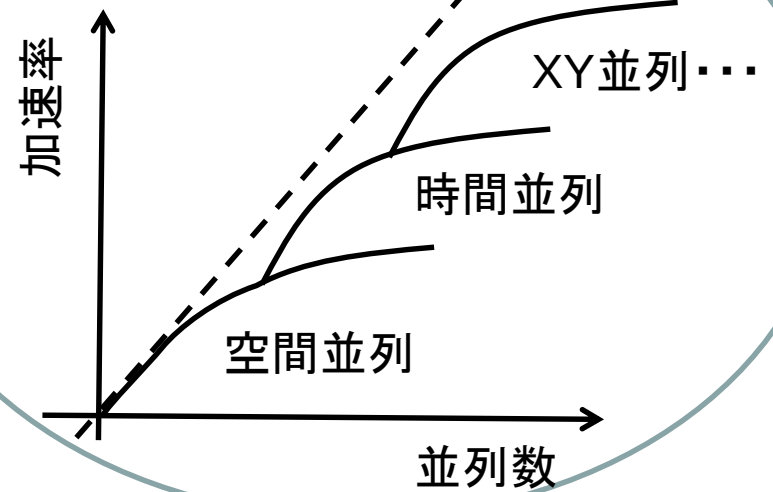
膨大なパラメータ空間の探索による
新現象の発見、最適設計へ寄与

Exa-scale 計算資源



ありあまる計算機の並列性！
どうやって使うのか？

さらなるtime to solutionの短縮



時間方向並列も、寄与できると期待

時間並列計算法とは

その仕組

1つの時間領域の時間発展方程式

$$\frac{\partial \mathbf{u}(t, \mathbf{x})}{\partial t} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}(t, \mathbf{x})),$$
$$\mathbf{u}(0, \mathbf{x}_B) = \mathbf{u}_0, \quad t \in \Omega = [0, T]$$

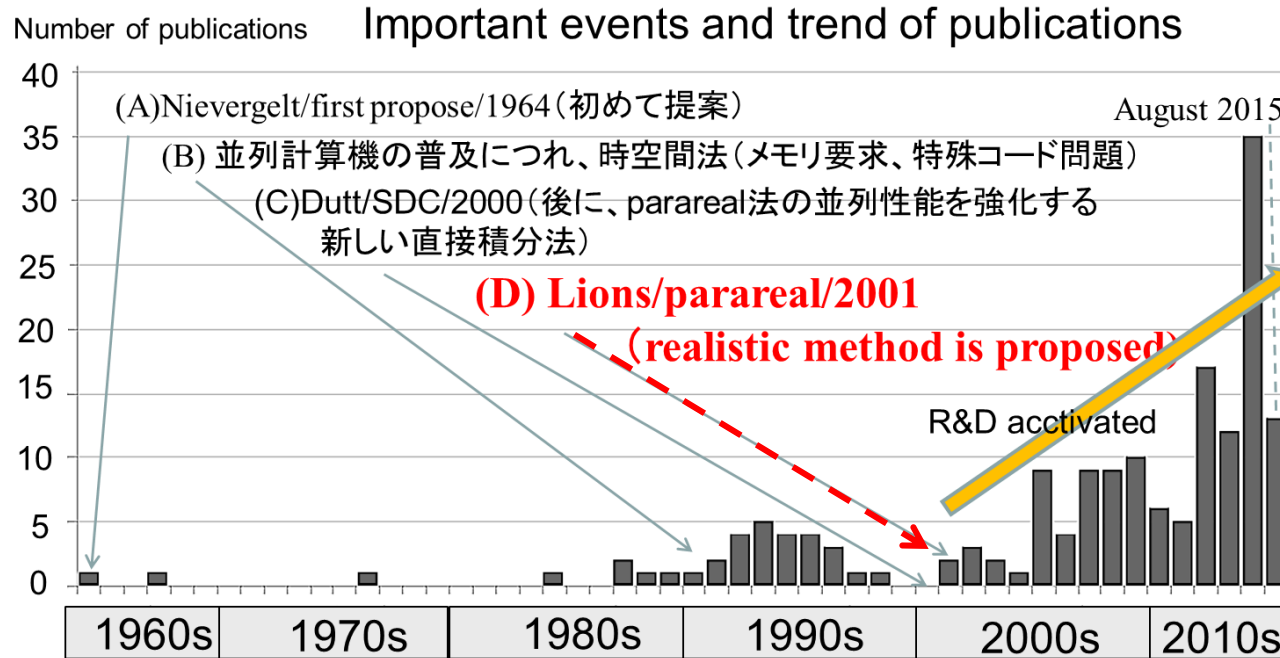


複数の時間領域の時間発展方程式

$$\tilde{\mathbf{Q}}(\mathbf{U}_{n-1}, \mathbf{U}_n) = 0$$
$$\mathbf{U}_0 = \mathbf{u}(0), \quad \mathbf{U}_{n-1} = \mathbf{u}(T_{n-1}), \quad \mathbf{U}_n = \mathbf{u}(T_n)$$

分割された1つの時間領域を Time Sliceと呼ぶ

各 time slice に、同時にプロセッサを割り当て、並列計算をすること

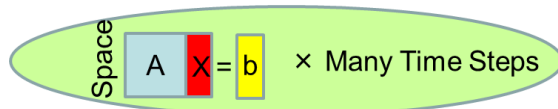


<http://www.parallelintime.org/references/index.html>

2001年にLionらが時間方向マルチグリッド法的Parareal法を提案、
それを契機にPinT研究が活性化

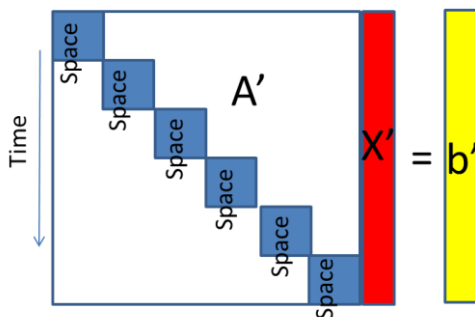
時間並列計算の大きな流れが、2つあり

逐次時間計算



1つのステップの行列計算 × 多数ステップ

行列型
時空間
MG 法



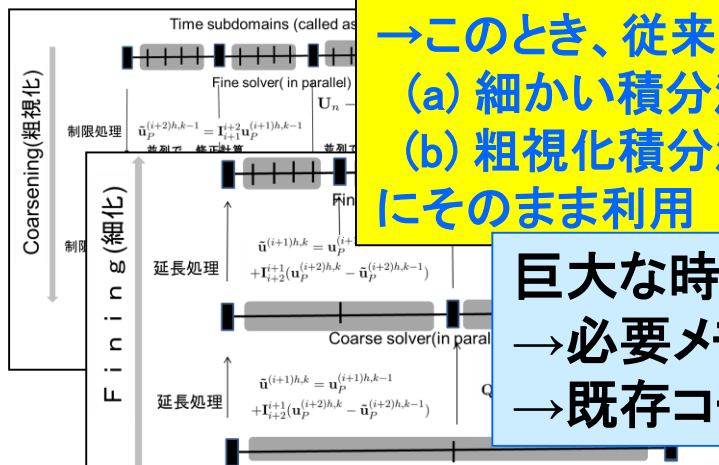
巨大な時空間行列を利用

- ・メモリがネック
- ・非対称性が強く、解くのが難しい

1990年代に流行った、時空間法

これは実用的ではない

時間積
分型時
空間
MG 法



反復法により時間領域間の境界値を擦り合わせる:
→このとき、従来の時間積分法・コードを、Time Slice内の
(a) 細かい積分演算(残差計算: Fine solver)
(b) 粗視化積分演算(修正・緩和計算: Coarse-solver)
にそのまま利用

巨大な時空間行列を必要としない

→必要メモリを削減、

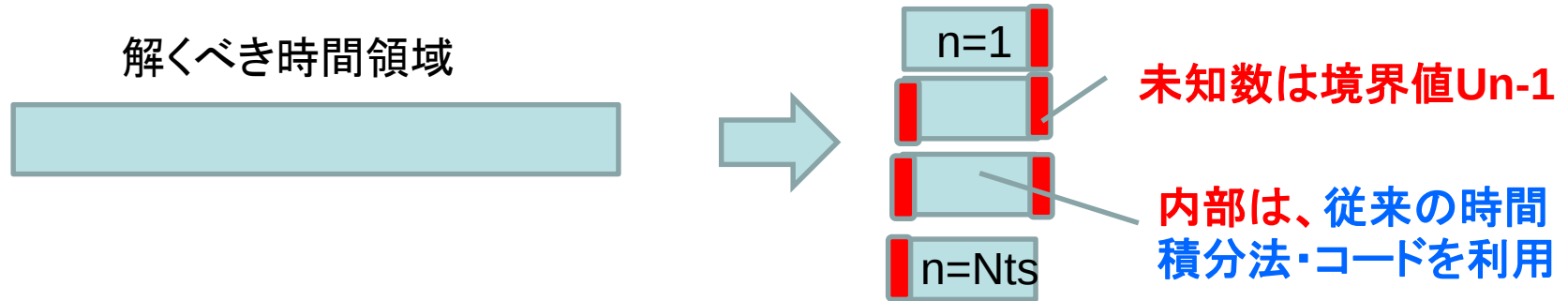
→既存コードの流用を可能とする

これは、実用的

Parareal法の説明

- 分割時間領域境界値 U_{n-1} を未知数とする N_{ts} 本の非線形方程式を解いて求める問題として定式化

N_{ts} 個の分割時間領域



- Newton-Raphson法を使ってParareal法が以下のように求められる

$$U_n^k = \varphi_{\Delta T_n}(U_{n-1}^{k-1}) + \partial_{U_{n-1}} \varphi_{\Delta T_n}(U_{n-1}^{k-1}) dU_{n-1}^k$$

この修正項の計算の方法の工夫が新しい発想だった

反復更新値

↓

 $U_n^k = \varphi_{\Delta T_n}(U_{n-1}^{k-1}) + \varphi_{\Delta T_n}^C(U_{n-1}^k) - \varphi_{\Delta T_n}^C(U_{n-1}^{k-1})$

残差計算

↓ Fine solver

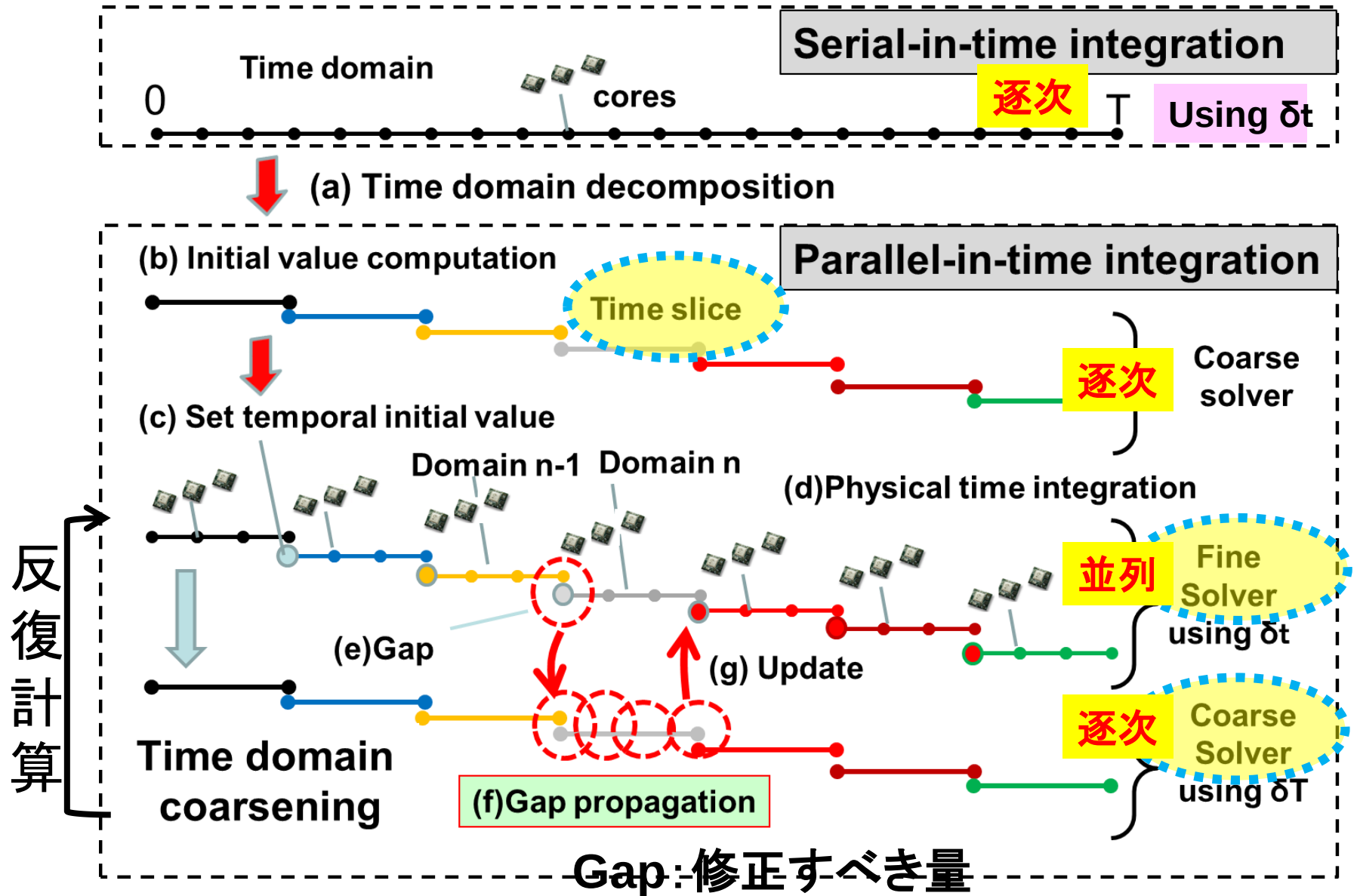
修正計算 (Jacobian計算は困難)

↓ Coarse solver (時間粗視化による粗い計算)

通常はF,Gを使いこのように書く

$$U_n^k = F(T_n, T_{n-1}, U_{n-1}^{k-1}) + G(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^{k-1})$$

参考: Parareal法の計算の流れ



Parareal法の並列性能: 並列加速率 α の挙動特性

Parareal法 → 人工的に並列性を作成

$$\alpha = \frac{\text{逐次計算時間}}{\text{Parareal 法による並列計算時間}}$$

$$= \frac{T_F N_{ts}}{(T_F + T_S) K^{par} + T_S} \quad : T_S = -t_{com} + (t_{com} + T_G) N_{ts}$$

$$= \frac{1}{K^{par} \left(\underbrace{\frac{1}{N_{ts}}}_{\text{並列部分}} + \underbrace{\frac{1}{R_{fc}} \left(1 + \frac{1}{K^{par}} \right)}_{\text{逐次部分}} \right)}$$

(1) この部分は、これまでの加速率の式と同等

Parareal法の反復数(反復数の削減がとても重要)

(2) 逐次部分は、時間粗視化率: 修正計算の負荷に影響される
 $R_{fc} = T_F / T_G \doteq \delta T / \delta t$: 時間粗視化率(逐次計算負荷の指標)

人工的に作られた並列性の利用の影響がある

動向：期待が大きく、すでに、理論研究と多くの分野での試行 →それを受けて、どう研究開発を進めているか

ODEs 放物型PDE(拡散等)

実用化のレベル、が、問題適用での性能挙動の調査が手薄⇒実際の問題で腕を磨いて、レベルアップを継続する段階



双曲型PDEs(移流、波動)

時間粗視化による
Fine solverとCoarse
solverの波の位相の差
が課題

問題に応用しレベルアップを図る

- 基本的拡散方程式で大規模並列性能評価・課題出しと解決へ
- フェーズフィールド法へ応用し、評価・課題出しと解決へ
- その他へ拡大

優先
順位2

課題解決の計算法の開発から行う

- 高精度・低計算負荷計算法の研究開発
 - ・移流：時空反転対称性スキーム(渡部)
 - ・波動：修正Newmark- β 法(水野)
 - ・汎用的：WATMUS(多重解像度法の応用)
- 応用・課題出し・評価...

優先
順位1

動向とそれを受けて、どう研究開発を進めているか

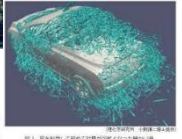
期待が大きく、すでに、理論研究と多くの分野での試行→

ODEs 放物型PDE(拡散)
実用化のレベル、が、問題適
性能挙動の調査が手薄⇒
題で腕を磨いて、レベルア
する段階



が、今回は、ここ
に着目した話

Es(移流、波動)



問題に応用しレベルアップを図る

- 基本的拡散方程式で大規模並列性能評価・課題出しと解決へ
- フェーズフィールド法へ応用し、評価・課題出しと解決へ
- その他へ拡大

優先順位1

優先順位2

課題解決の計算法の開発から行う

- 高精度・低計算負荷計算法の研究開発
 - ・移流:時空反転対称性スキーム(渡部)
 - ・波動:修正Newmark- β 法(水野)
 - ・汎用的:WATMUS(多重解像度法の応用)
- 応用・課題出し・評価...

基本的な拡散問題でのParareal法の大規模並列性能

■今回の目的

- (A) 時間並列は空間並列と独立に加速可能か
- (B) 放物型問題においても残る課題の抽出

基本的な拡散問題での大規模並列計算法

■ 解析問題: 3次元拡散問題

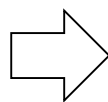
$$u_t(\mathbf{x}, t) = \nu \Delta u(\mathbf{x}, t), \quad \nu = \frac{1}{3},$$

$$\mathbf{x} \in \Omega = [0, L = 1]^3, \quad t \in [0, T = 1]$$

初期条件と境界条件は以下である。

$$u(\mathbf{x}, 0) = \sin(\pi x) \sin(\pi y) \sin(\pi z),$$

$$u(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \partial\Omega.$$



■ 空間離散化式

(i,j,k方向の2次中心差分法)

$$\mathbf{u}_t(ijk, t) = \nu \Delta \mathbf{u}(ijk, t)$$

$$= \mathbf{Q} \mathbf{u}(ijk, t)$$

■ 時間離散化: Crank-Nicolson法

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

Fine, Coarse solver内: 各time step で、
Red-BlackSOR前処理付Bi-CGstab法で解く

時空間のHybrid並列

時間並列 (Parareal法) : MPI

$[T_0, T_1]$

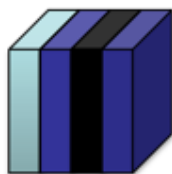


$[T_n, T_{n+1}]$

MPI

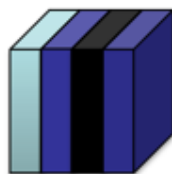
$[T_{N_t-1}, T_{N_t}]$

空間並列



MPI+OpenMP

空間並列

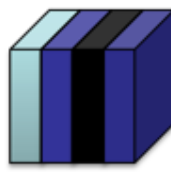


MPI+OpenMP

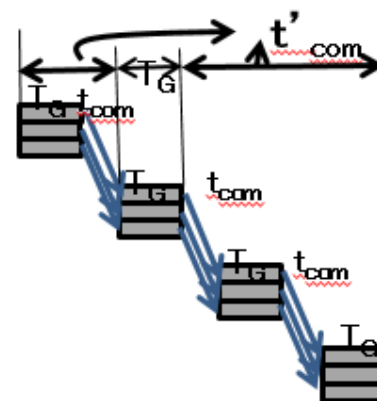
...

...

空間並列



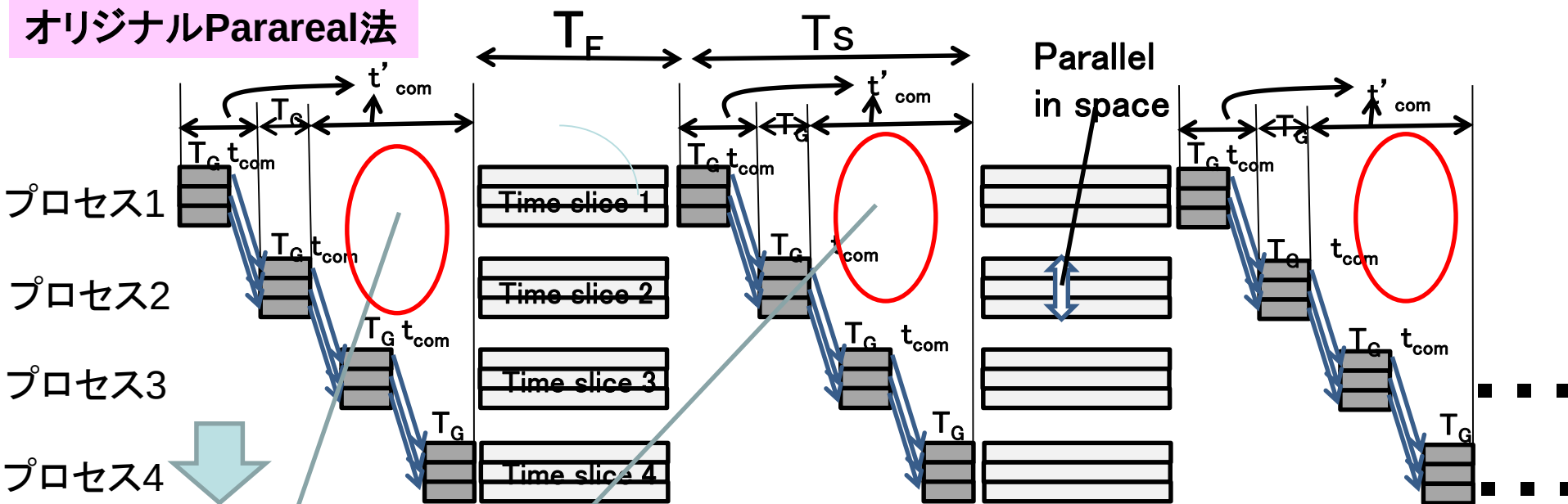
MPI+OpenMP



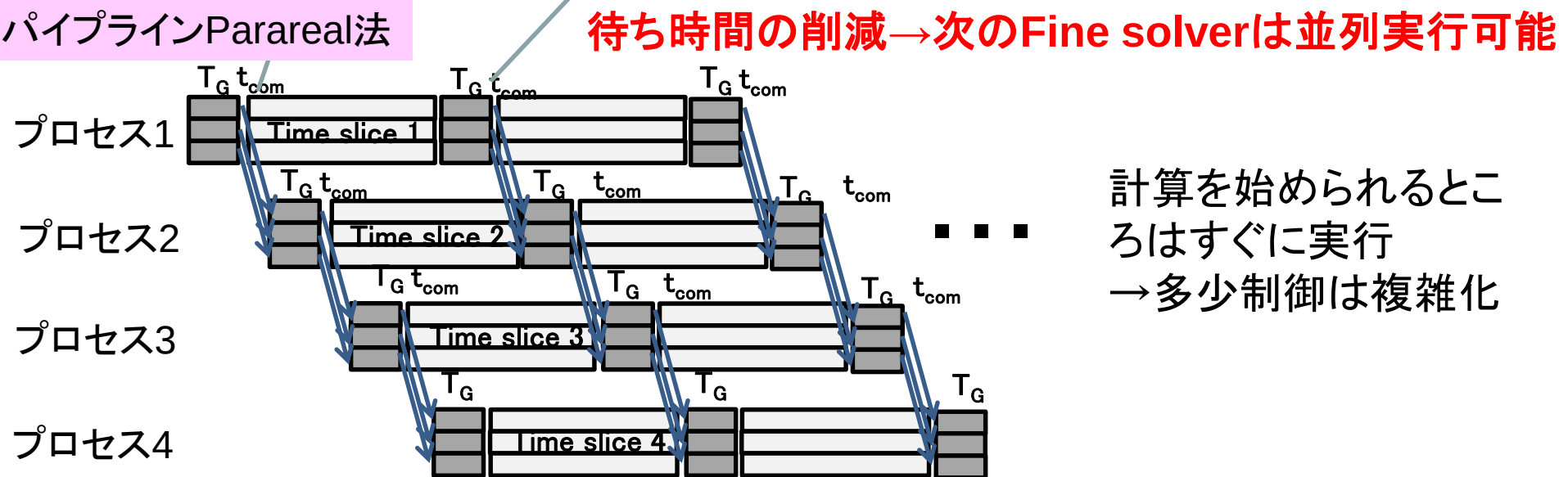
Time sliceの終点の値を次のtime sliceの始点として送る

Pipeline化: Pipelined parallel法

オリジナルParareal法

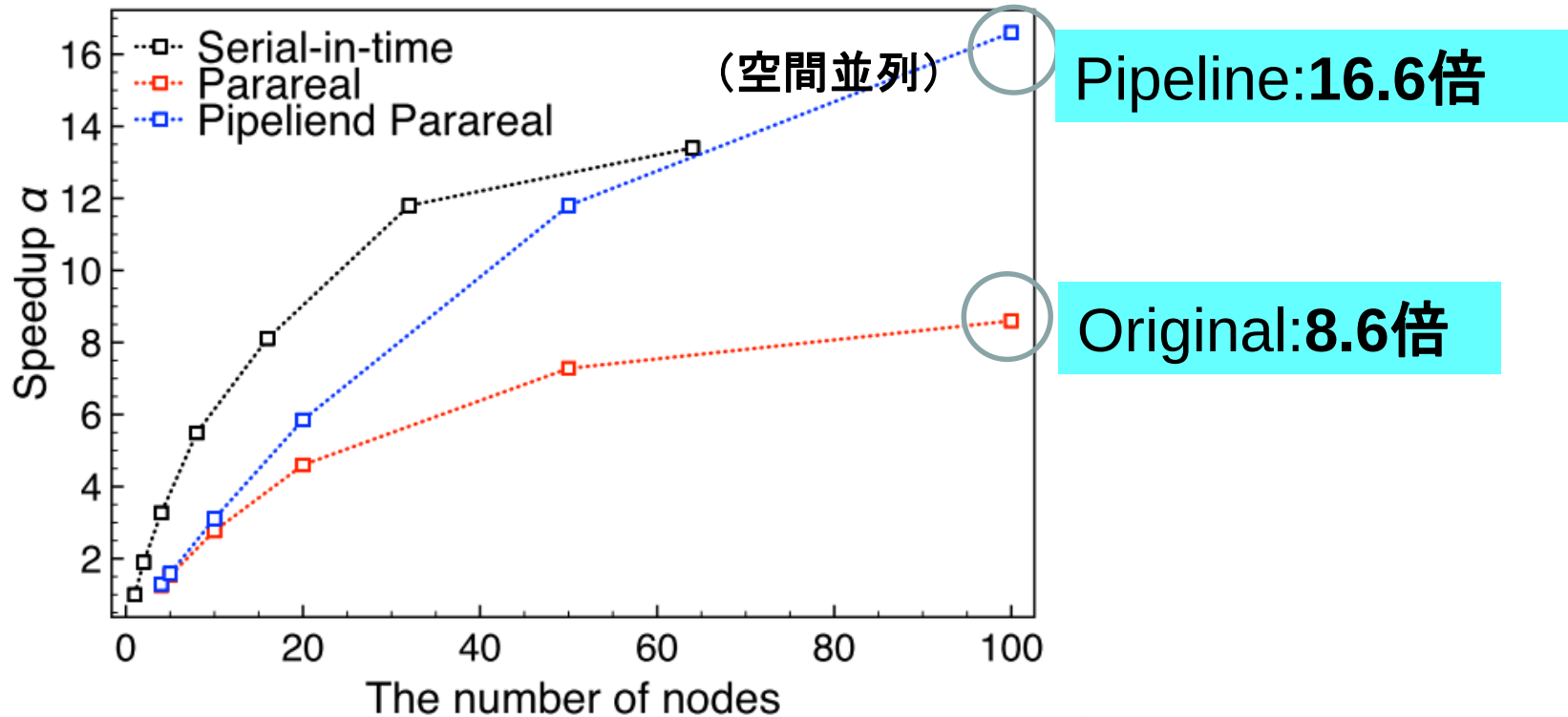


パイプラインParareal法



空間、時間方向それぞれの並列性能

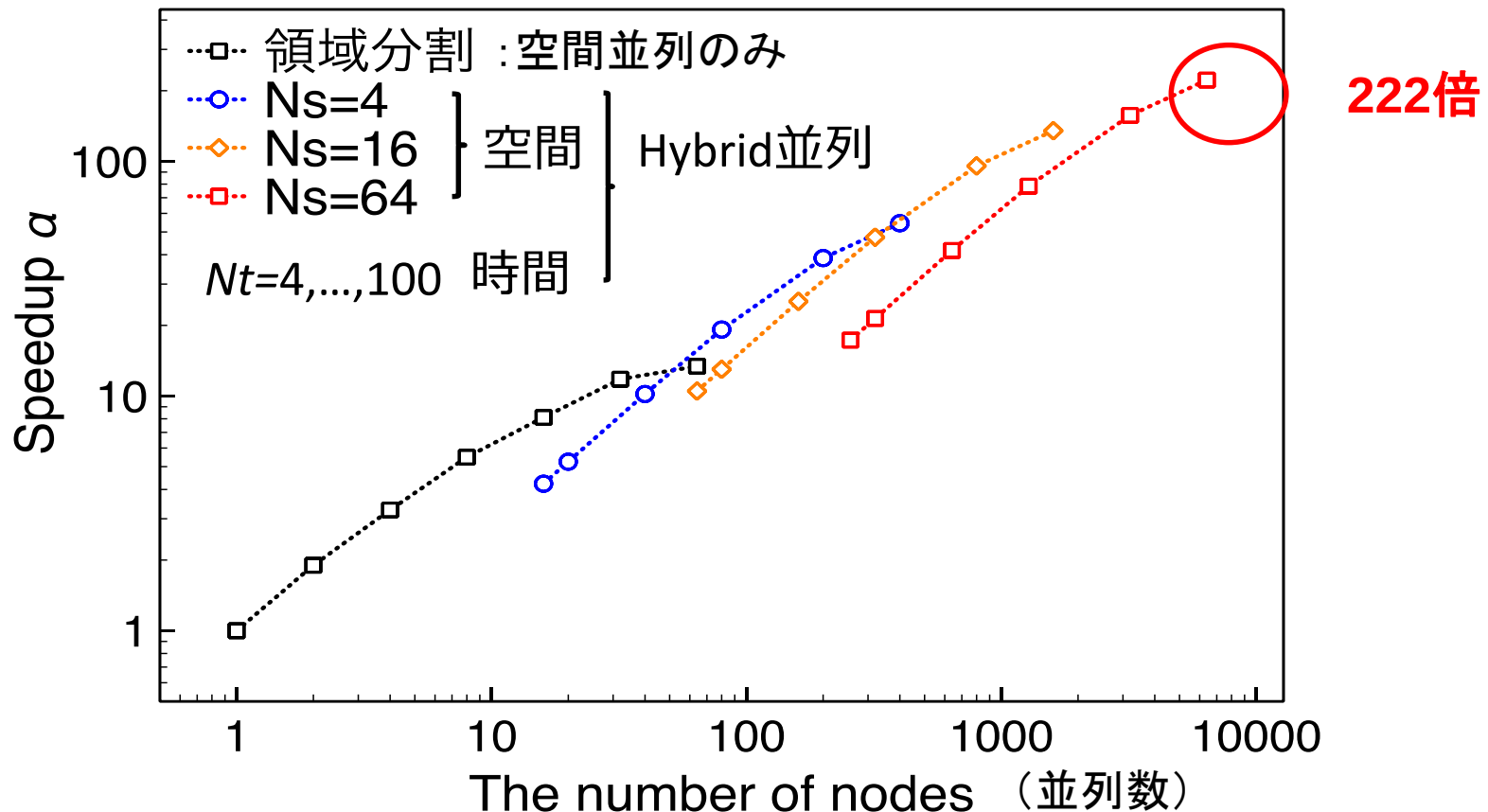
Grid数	Ng	128 ³ (固定)
空間領域分割	Nd	1, 2, 4, 8, 16, 32, 64 (空間並列数)
Number of time slices	Nts	10, 20, 50, 100 (時間並列数)
δt		0.0001
δT		0.1 (時間粗視化率 Rfc=100)



Pipeline 化は大きな効果あり、
100～200時間並列(並列効率～20%)までは加速可能か

時間並列は空間並列と独立に加速可能か

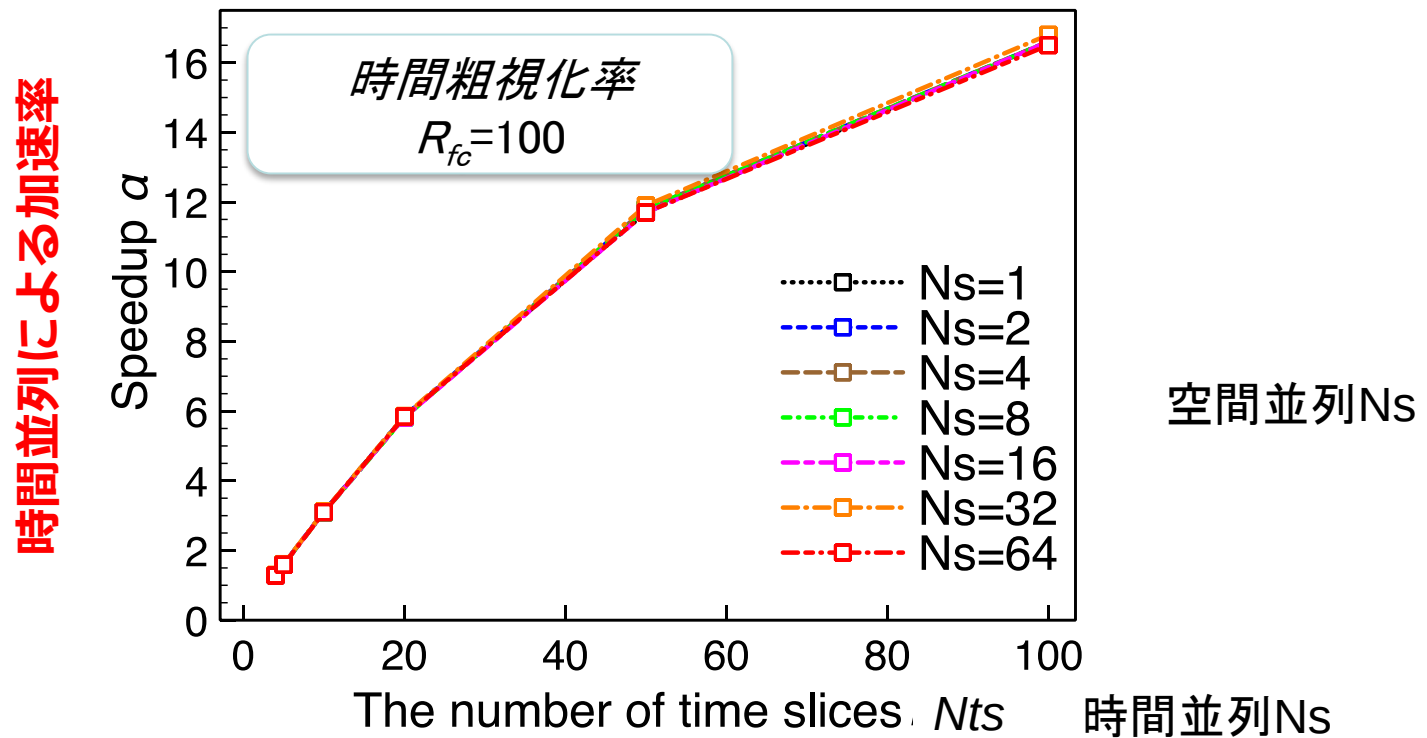
時空間Hybrid法 : Pipelined Parareal法, $\delta T=0.01$ ($R_{fc}=100=\delta T/\delta t$)



- 空間並列64、時間並列100で222倍の加速
- 独立に加速可能なように見える

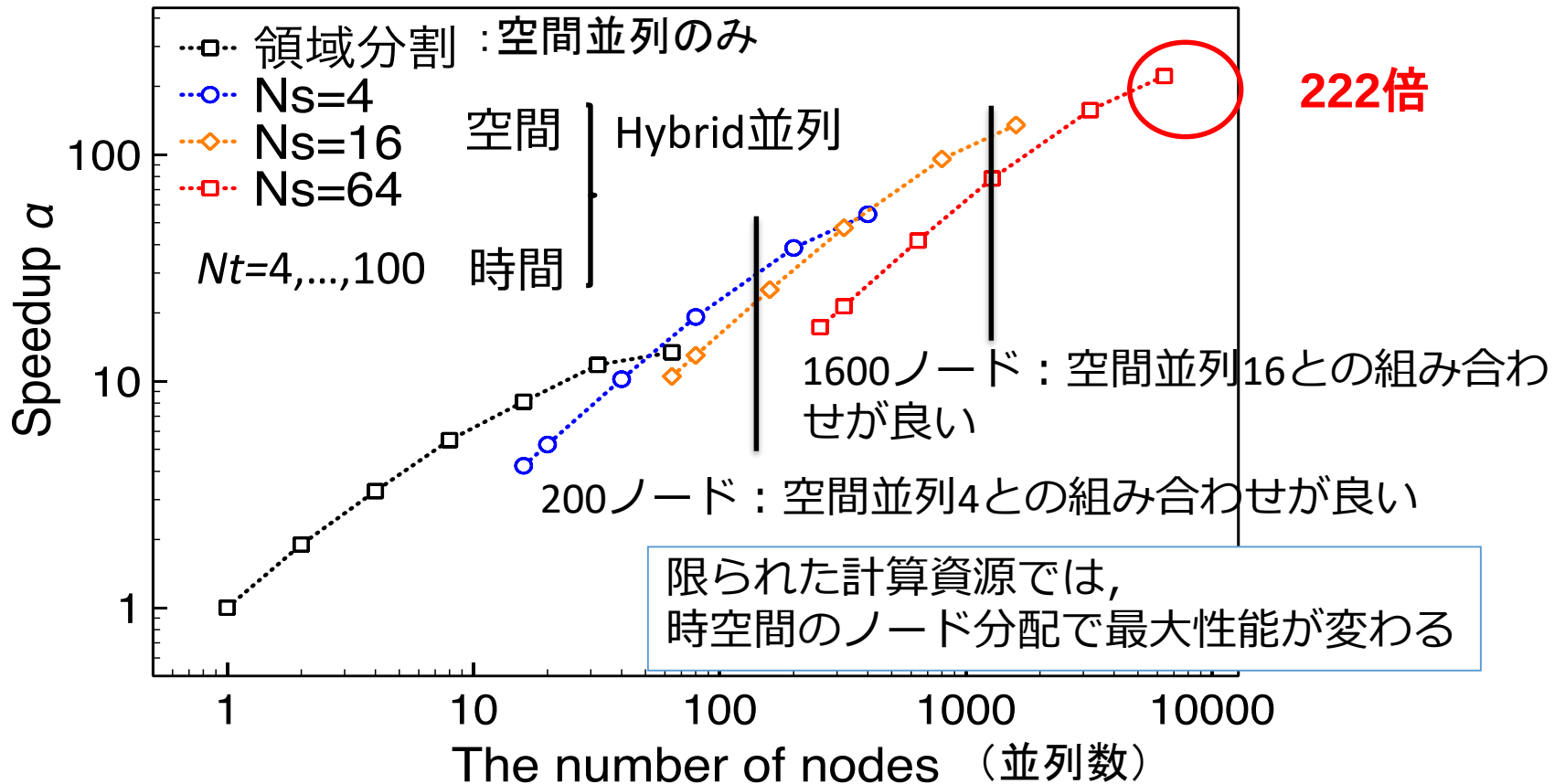
より明快に見るために、時間並列軸でまとめてみると

空間へ並列数 N_s に対して→時間並列数による加速率を見た



空間並列によらず、時間並列による加速が可能なが分かる

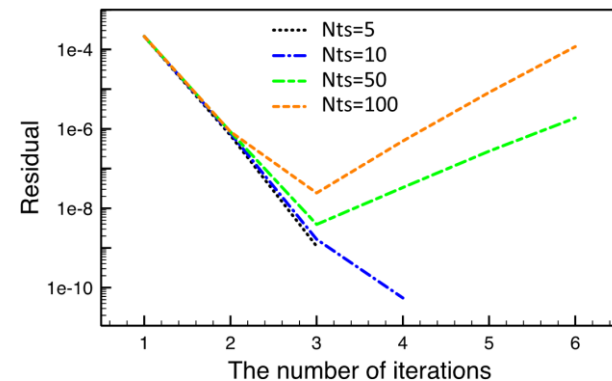
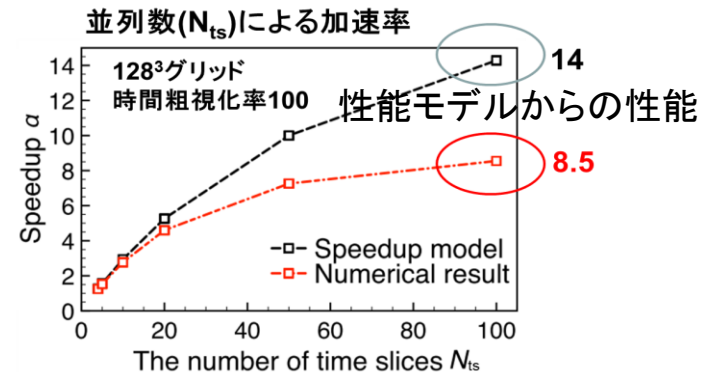
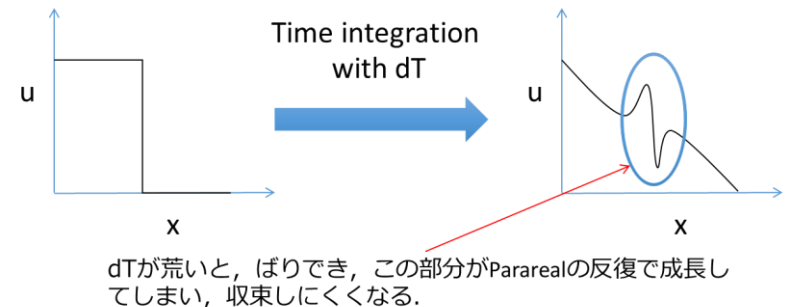
さらに、使える計算資源に制約があるとき、
時間並列を使うタイミングを調べて見ると...



使える計算資源量により使い方を吟味する必要あり

残る課題の抽出

- 粗い時間ステップで、急峻な分布(高波数モード)に対して、粗視化計算で非物理的振動→Parareal法の収束性を低下させる
- 性能モデルから予想される理想的な性能が出ない→粗視化時間積分による時間積分精度の差が、線形ソルバーの反復数を増加させるため
- 時間粗視化率、time slice数の増加で収束のある時点から誤差が成長し→収束残差が途中で増加に転じる場合がある



計算負荷の低い高精度時間積分法(4次後退オイラー、WATMUS...)で解決可能と予測→今後検討を続ける

まとめ

■ 時間並列計算は、放物型には使える

- しかし、課題があり、利用技術を磨いていく必要がある
- 応用分野の拡大を試行→フェースフィールド法へ

■ 2つの基本式

- 系の全自由エネルギーの減少の要請から

・保存量の式 $\frac{\partial c}{\partial t} = \nabla \cdot \left\{ M_c(c, \phi) \nabla \left(\frac{\delta G}{\delta c} \right) \right\}$ Cahn-Hilliard方程式
→空間4階微分項あり

・非保存量の式 $\frac{\partial \phi}{\partial t} = -M_\phi(c, \phi) \left(\frac{\delta G}{\delta \phi} \right)$ Allen-Cahn方程式
→空間2階微分項

■ 双曲型への取り組み

- 近年、工学分野で進んでいる、大きな時空間のメッシュ幅(低計算負荷)で高精度計算を可能とする技術(WATMUS, FIC, 時空間反転対称性等)を試行、改良の予定

- 逐次計算の並列計算化研究の道標となる可能性がある(「逐次問題の並列計算の数理とフレームワーク研究開発・実証」—科研費課題番号17H01750で提案・受託、実施中)