

フェーズフィールド法に対するParareal法による時間並列計算の収束挙動

飯塚, 幹夫
九州大学情報基盤研究開発センター

小野, 謙二
九州大学情報基盤研究開発センター

<https://hdl.handle.net/2324/1916254>

出版情報：数値流体力学シンポジウム講演要旨集. 31 (C02-3), pp.1-4, 2017-12-12. 日本流体力学会
バージョン：
権利関係：

フェーズフィールド法に対するParareal法による 時間並列計算の収束挙動

Convergence of Parallel-in-time Integration for Phase field method by Parareal Method

○飯塚幹夫*¹、小野謙二*^{1,2}

* 1:九州大学 情報基盤研究開発センター

* 2:理化学研究所計算科学研究機構、

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:hp170238)。

発表内容

1. 本研究の背景 (PinT: 時間並列計算) と目的
2. フェーズフィールド法へのParareal法の適用法
3. 収束性調査のための解析条件
4. 収束データの測定結果と考察
5. まとめと今後

1:本研究の 背景 (PinT: 時間並列計算) と目的

PinT: 本来逐次処理である時間積分を並列で行う技術

PinTを必要とする背景:

- Time to solution 的視点:
長時間の時間積分計算が課題のもの
 - ・第一原理分子動力学
 - ・核融合炉設計のためプラズマ計算
 - ・心臓シミュレータ
 - ・フェーズフィールド法
- 空間並列性能飽和後もさらに加速が必要
- 増え続けるコア数を背景に、並列演算器の使い道を広げる手段を増やす

計算のスピードアップ

時間並列
Parallel-in-time

飽和: 限界

空間並列
Parallel-in-space

並列数

Parareal法:

予測計算: 細かい
積分演算: δt で
計算(Fine solver)

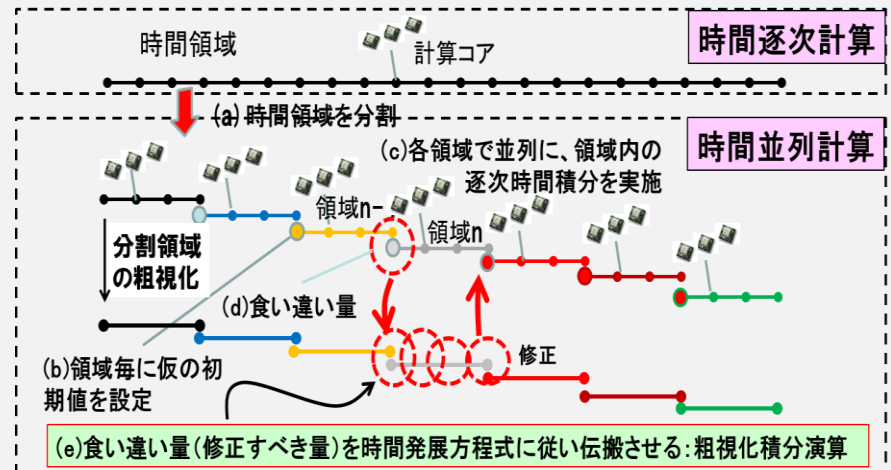
修正計算: 粗視化積
分演算: δT で計算
(Coarse solver)

どう計算するか:

- 1980年代後半～: **時空間法**: 時空間の巨大な行列を作成し、それを解く⇒メモリ容量・超大規模ソルバーがネック
- 2001年～: メモリ容量・超大規模ソルバーが不要な**Parareal法**が提案: 時間方向を領域分割し、分割領域の仮の初期値を使った既存の時間積分をおこない、その仮の初期値を領域間ですり合わせていく計算法

領域端点の値の更新計算

$$U_n^k = \varphi_{\Delta T_n} (U_{n-1}^{k-1}) \quad \text{Fine solver} \\ + \varphi_{\Delta T_n}^C (U_{n-1}^k) - \varphi_{\Delta T_n}^C (U_{n-1}^{k-1}) \quad \text{Coarse solver}$$



Parareal法の研究開発動向

	2001 ~	2008 ~ 2017
全体傾向	<ul style="list-style-type: none"> ● 数学分野の領域分割法研究者による理論的基礎研究 ● 研究量は微増傾向 ● 多くの分野への適用と落胆 	<ul style="list-style-type: none"> ● 課題解決による応用へむけた研究が活発化 ● 研究量の急増
放物型 PDEs	<ul style="list-style-type: none"> ● 理論と基礎的なテストで、収束性が比較的良いことがしめされた 	<ul style="list-style-type: none"> ● 実用的応用への動き <ul style="list-style-type: none"> ● 核融合プラズマ(ダイバータ) ● フェーズフィールド法 ● 地下資源・環境解析
双曲型 PDEs	<ul style="list-style-type: none"> ● 理論と基礎的なテストで、収束が困難であることが示された 	<ul style="list-style-type: none"> ● 収束性向上の研究開発活発化 <ul style="list-style-type: none"> ● 基底次元低減法、随伴方程式法、漸近展開法等 ● 2017: 高精度・高負荷時間積分問題に対して効果の傾向があることが実問題(津波解析, 高次WENO)で示された(Coarse solver-Roe's 法).

今後、実用へ向けた研究開発が増々活発化すると推定

これまでの本研究

第1ステップ: 大規模拡散問題を対象

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \Delta \mathbf{u} \quad \nu = \frac{1}{3}$$

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

計算条件

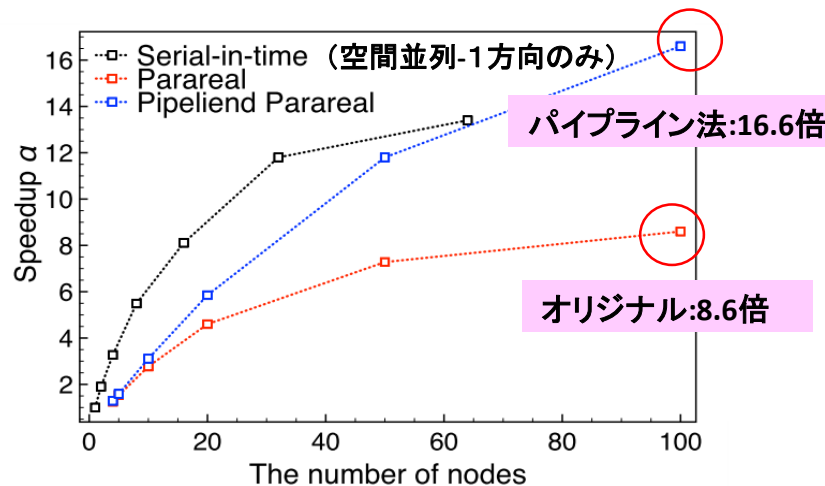
$$u(\mathbf{x}, 0) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$$

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

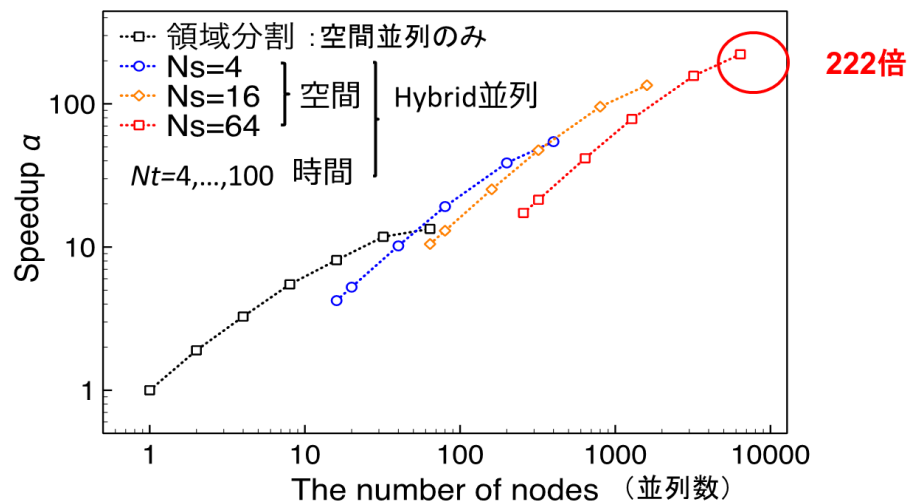
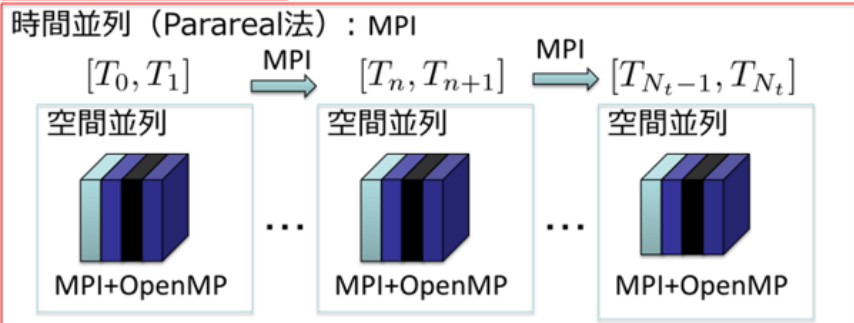
解析解

$$u(\mathbf{x}, t) = \sin(\pi x) \sin(\pi y) \sin(\pi z) \exp(-\pi^2 t)$$

加速可能性を確認してきた



時空間のHybrid並列



目的: フェーズフィールド法に対するParareal法による時間並列計算の収束挙動をまず調べる

2: フェーズフィールド法への Parareal法の適用法

フェーズフィールド法の方程式と離散化

- Allen-Cahn方程式: **AC**、Cahn-Hilliard方程式: **CH**

今回は、1次元の単純な問題を扱います

$$\frac{\partial \phi}{\partial t} = \begin{cases} D\nabla^2 \phi - k\phi(\phi - 1.0)(\phi - 0.5 + \beta), & : AC \\ (p\partial_x^2 + q\partial_x^4)\phi - r\partial_x^2 \phi^3 & : CH \end{cases} \quad (1a)$$

$$= \begin{cases} {}^{AC}RHS = {}^{AC}RHS_L + {}^{AC}RHS_N & (1b) \\ {}^{CH}RHS = {}^{CH}RHS_L + {}^{CH}RHS_N & (2b) \end{cases}$$

$${}^{AC}RHS_L = D\nabla^2 \phi$$

$${}^{AC}RHS_N = -k\phi(\phi - 1.0)(\phi - 0.5 + \beta)$$

$${}^{CH}RHS_L = (p\partial_x^2 + q\partial_x^4)\phi$$

$${}^{CH}RHS_N = -r\partial_x^2 \phi^3$$

- 離散化:

- 空間: 中心差分 2次

- 時間: Forward Euler 1次、

Crank-Nicolson 2次、構造保存2次

Backward Euler 4次

→ 次項参照

...陽解法

...陰解法

時間離散化式(陰解法): Coarse solver用

	Crank-Nicolson法 2次 (CN)	Backward Euler 法 4次 (BD4)
Allen-Cahn方程式: AC	$\frac{\phi_i^n - \phi_i^{n-1}}{\delta t} = 0.5 ({}^{AC}RHS_i^n + {}^{AC}RHS_i^{n-1})$ <p style="text-align: right;">(5a)</p>	$\frac{\sum_{l=0}^4 a_l \phi_i^{n-l}}{12\delta t} = {}^{AC}RHS_i^n$ <p style="text-align: right;">(a_l) = (25, -48, 36, -16, 3) (5b)</p>
Cahn-Hilliard方程式: CH	$\frac{\phi_i^n - \phi_i^{n-1}}{\delta t} = 0.5 ({}^{CH}RHS_i^n + {}^{CH}RHS_i^{n-1})$ <p style="text-align: right;">× (6a)</p>	$\frac{\sum_{l=0}^4 a_l \phi_i^{n-l}}{12\delta t} = {}^{CH}RHS_i^n$ <p style="text-align: right;">(6b)</p>
構造保存 Structure Preservation ×	$\frac{\phi_i^n - \phi_i^{n-1}}{\delta t} = 0.5 ({}^{AC}RHS_{L,i}^n + {}^{AC}RHS_{L,i}^{n-1}) + {}^{CH}RHS_{N,i}^{SP}$ <p style="text-align: right;">(7b)</p>	$\frac{\sum_{l=0}^4 a_l \phi_i^{n-l}}{12\delta t} = {}^{CH}RHS_{L,i}^n + {}^{CH}RHS_{N,i}^{SP}$ <p style="text-align: right;">(7b)</p>
	${}^{CH}RHS_{N,i}^{SP} = -r\partial_x^2 \left\{ \frac{(\phi^n)^3 + (\phi^n)^2\phi^{n-1} + \phi^n(\phi^{n-1})^2 + (\phi^{n-1})^3}{4} \right\}_i$ <p style="text-align: right;">(7c)</p>	

$$\begin{aligned} & \frac{U_k^{(n+1)} - U_k^{(n)}}{\Delta t} \\ &= \delta_k^{(2)} \left\{ p \left(\frac{U_k^{(n+1)} + U_k^{(n)}}{2} \right) + r \left(\frac{(U_k^{(n+1)})^3 + (U_k^{(n+1)})^2 U_k^{(n)} + U_k^{(n+1)} (U_k^{(n)})^2 + (U_k^{(n)})^3}{4} \right) \right. \\ & \quad \left. + q \delta_k^{(2)} \left(\frac{U_k^{(n+1)} + U_k^{(n)}}{2} \right) \right\} \quad \text{for } n \geq 0, 0 \leq k \leq N. \end{aligned} \quad (2.27)$$

Parareal法コードの構成方法

研究用のFrameworkコードを利用し、構成(逐次計算コードの組み込みは1日)

```
Framework() {
```

```
  • activator-get_solver_type  
  open_solver_mpi  
  inp_solver ←  
  init_solver ←  
  inp_pit  
  init_pit
```

```
  • initializer-coarse-solver  
  k=0  
  while (error > tol) {  
    k = k + 1  
    • set initial value  
    for kth iteration  
    • Time Slice loop: parallel  
    fine-solver  
    • get Time Slice edge data  
    from previous process  
    • Time Slice loop: serial  
    coarse-solver  
    • put Time Slice edge data  
    to next process  
    • error_cal }  
}
```

interface

```
Input part  
Init. part
```

PFM Solver

3つの部分に分割

```
Time loop part
```

Fine-solver(そのまま)

Coarse-solver(陰解法化)

UTIL:

Put/Get (Send/Receive) tools
SDC tools and template,
CRK4 subroutine, etc.

Parareal algorithm part

Parareal法コードの構成概要

	Fine solver	Coarse solver			
	Forward Euler 1次.	Crank-Nicolson 2次	Backward Euler 4次	構造保存 2次	Backward Euler 4次+構造保存
Allen-Cahn方程式: AC	○	○	○	×	×
Cahn-Hilliard方程式: CH	○	×	○	×	×

- 以上の○の付くものの組み合わせで行う
- 構造保存はParareal法の収束の改善が見られなかったため、今回は除外
- 今回、Parareal法の修正過程(G)の緩和が有効であることが分かったため導入

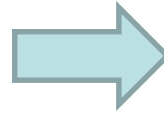
$$\begin{aligned}
 \mathbf{U}_n^k &= F(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}) \leftarrow \text{Fine solver} \\
 &+ \gamma \{ G(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) - G(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}) \} \\
 &\quad \leftarrow \text{Coarse solver}
 \end{aligned}$$

1.0 \geq 緩和係数 γ

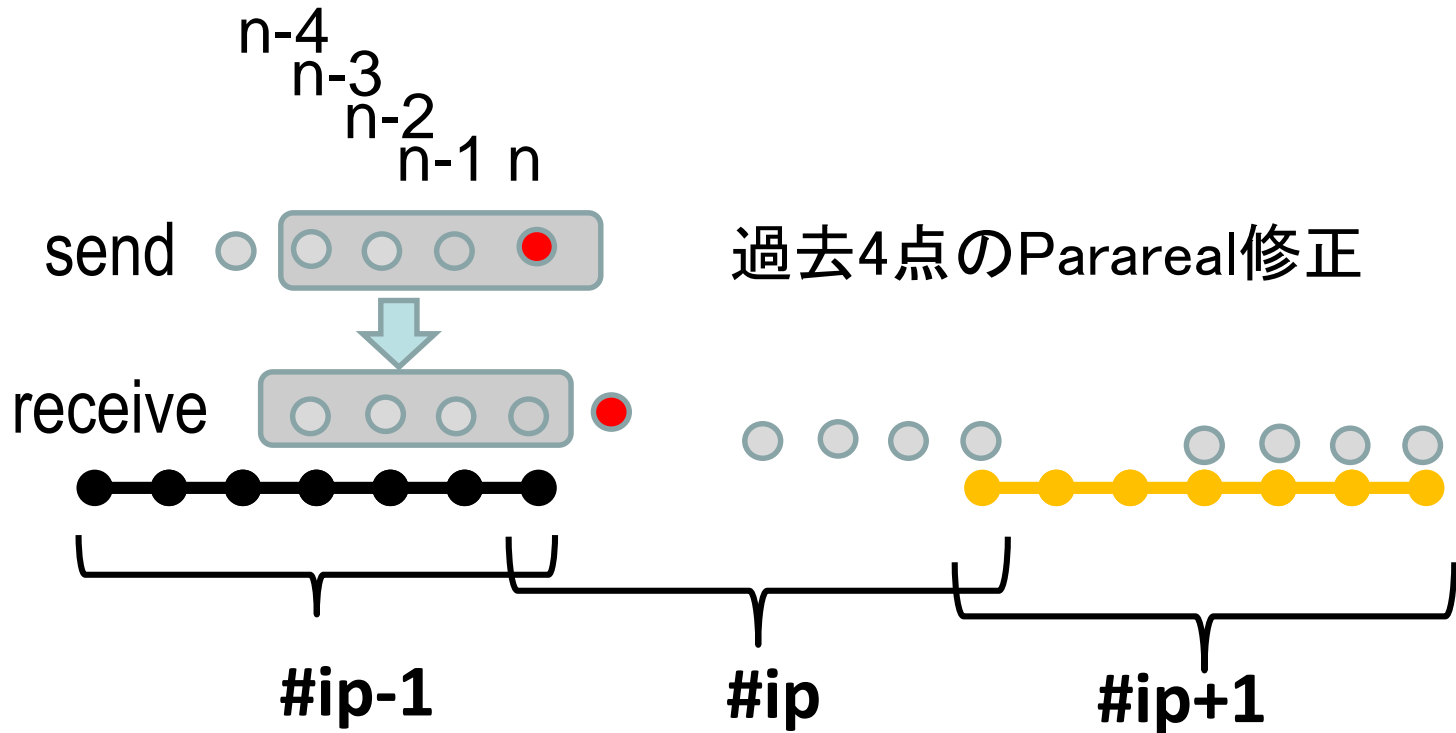
Backward Euler 4次のParareal法の修正過程

式(5b)

$$\frac{\sum_{l=0}^4 a_l \phi_i^{n-l}}{12\delta t} = {}^{AC} RHS_i^n$$

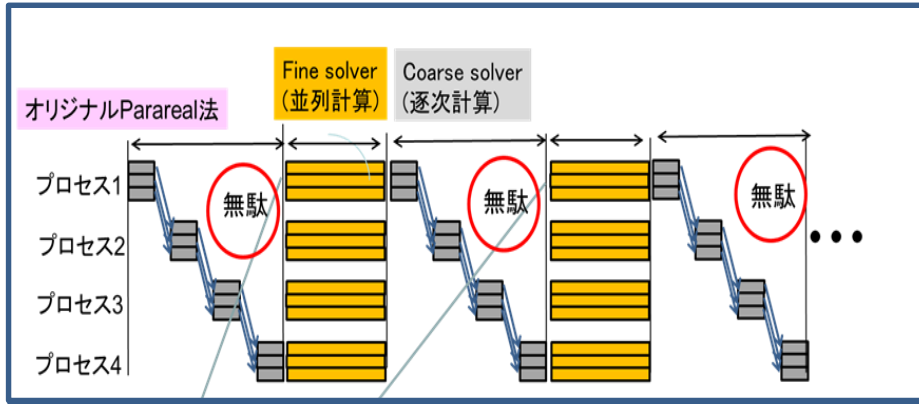


過去の4点が必要となるので、
領域境界にて、過去4点のParareal
修正を同時行い次の時間領域(time
slice)へおくる



性能評価の式

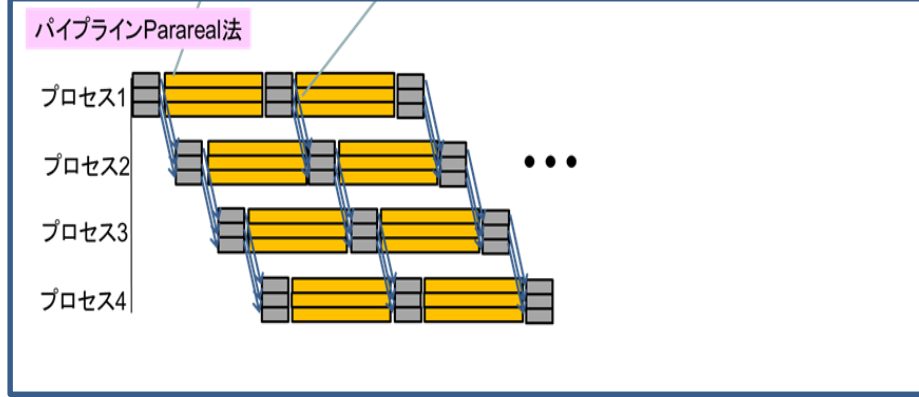
加速率 $\alpha \left(N_{ts}, R_{fc}, K^{par}, \frac{t_{com}}{T_G} \right) = \frac{\text{逐次計算時間}}{\text{Parareal法の並列計算時間}}$



$$\frac{T_F N_{ts}}{(T_F + T_S N_{ts}) K^{par} + T_G N_{ts}}$$

$$\approx \frac{N_{ts}}{\left(1 + \frac{N_{ts}}{R_{fc}}\right) K^{par} + \frac{N_{ts}}{R_{fc}}}$$

$$= \frac{N_{ts}}{\left(1 + K_0^{par}\right) K^{par} + K_0^{par}} = \frac{N_{ts}}{\tilde{K}^{par}}$$



$$\frac{T_F N_{ts}}{(T_F + T_S) K^{par} + T_G N_{ts}}$$

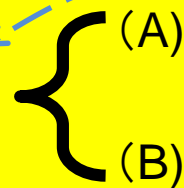
$$\approx \frac{N_{ts}}{K^{par} + \frac{N_{ts}}{R_{fc}}} = \frac{N_{ts}}{K^{par} + K_0^{par}} = \frac{N_{ts}}{\hat{K}^{par}}$$

以下、パイプライン法で考察

並列数

実効反復数

実効反復数の低減が重要



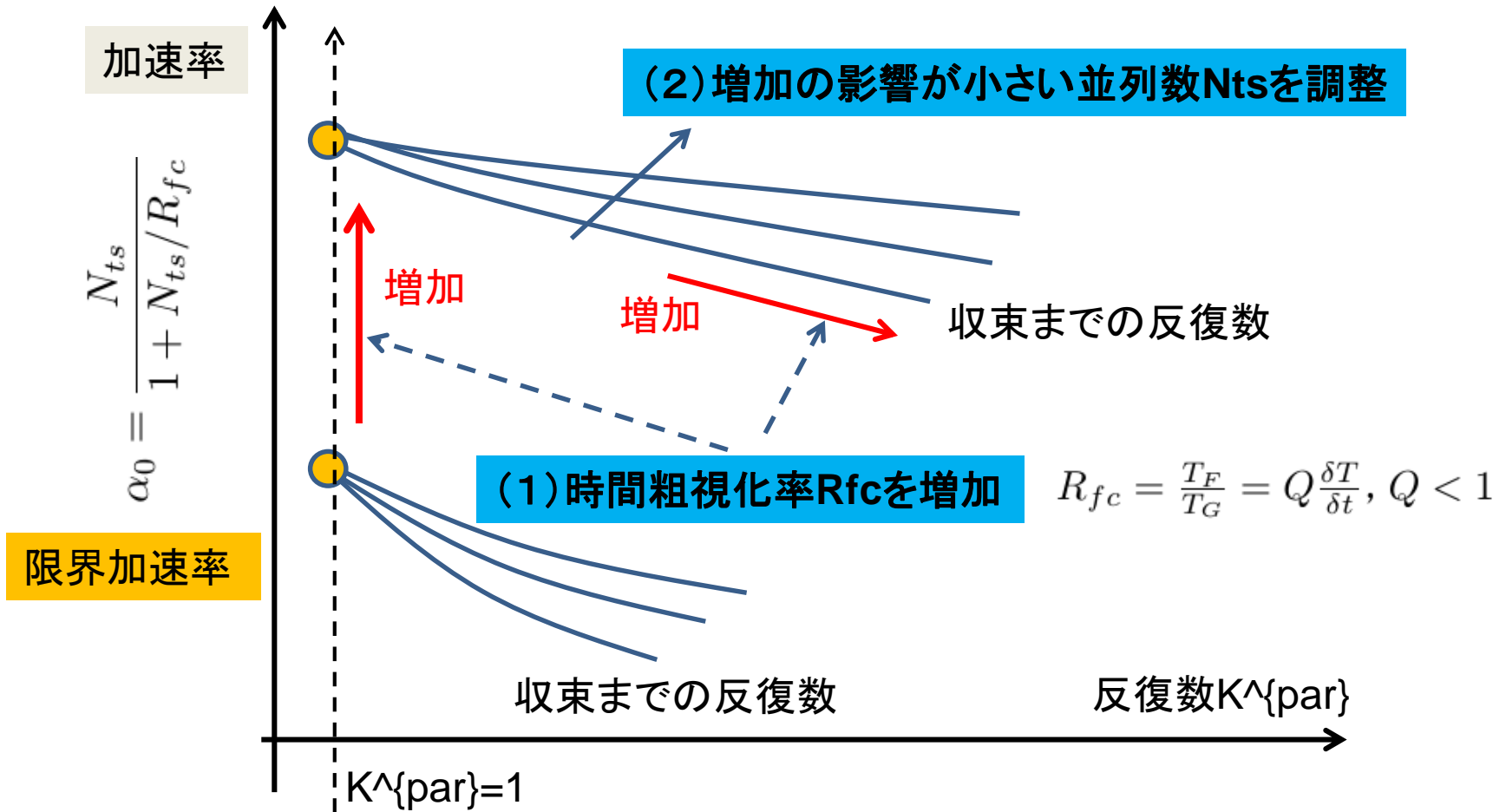
(A) $K_0^{par} = \frac{N_{ts}}{R_{fc}}$: 初期反復数

(B) $\alpha_0 = \frac{N_{ts}}{1 + N_{ts}/R_{fc}}$ 反復数=1の時の最大値: 限界加速率

削減

増加

性能調整の方針の考察：限界加速率から



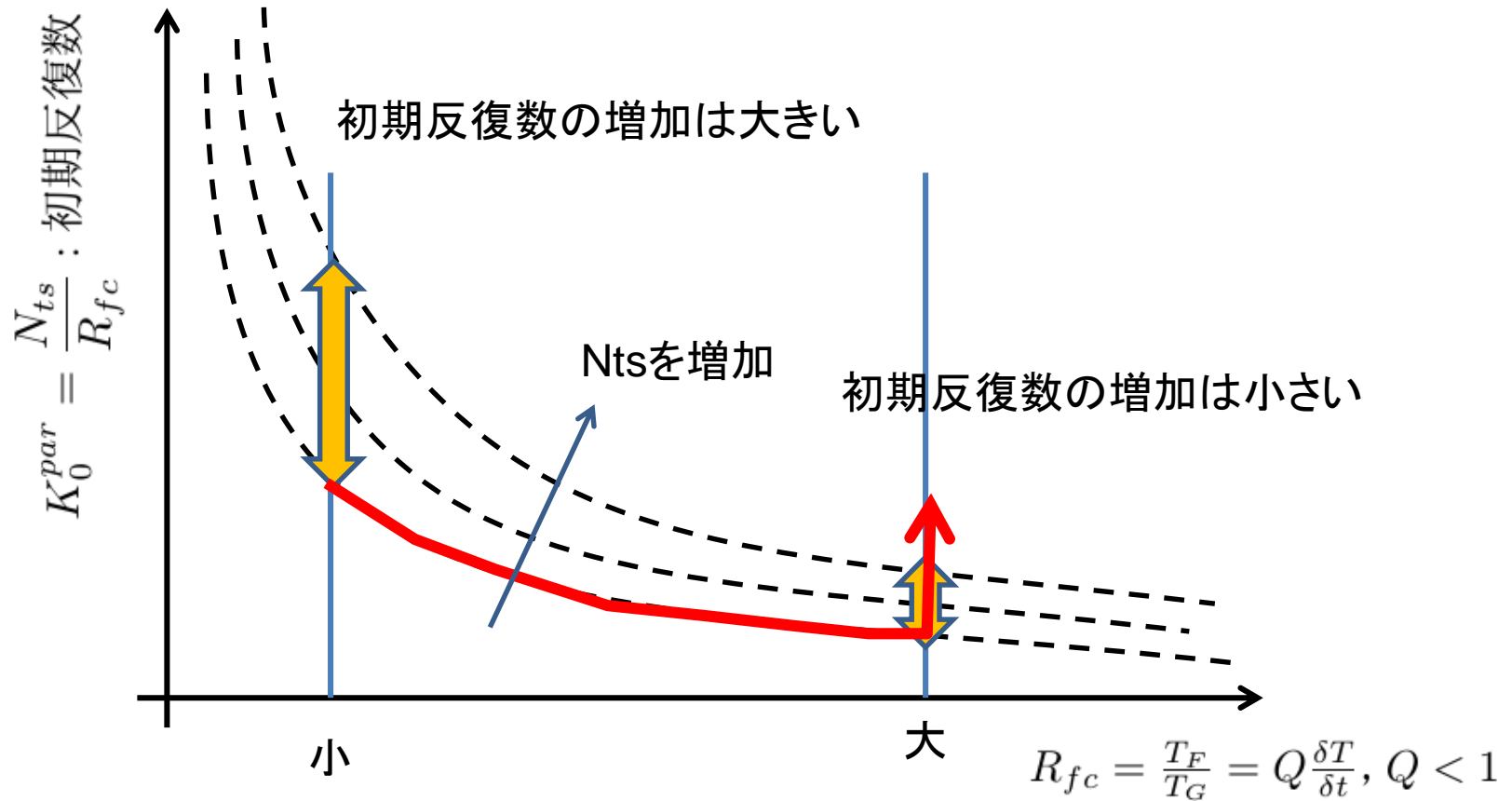
並列数 N_{ts} の増加の影響は小さい

→ R_{fc} を反復数の増加が少ない範囲で増加させて→限界加速率を上げるよう調整

→その後で増加の影響が小さい並列数 N_{ts} を調整

→この方が得！

性能調整の方針の考察：初期反復数の挙動から



時間粗視化率を反復数があまり増加しない範囲で増加する

時間粗視化率を反復数があまり増加しない範囲で増加して、その後で
並列数Ntsお大きくすれば、初期反復数の増加は抑制できる

今回の検討法

フェーズフィールド法に対する初期の検討として：
Parareal法による時間並列計算の収束挙動を調査

影響の小さい N_{ts} を100程度で、調査をはじめて

$\delta T / \delta t$ を反復数の増加が少ない
範囲で増加させて
→限界加速率を上げる

$$\alpha_0 = \frac{N_{ts}}{1 + N_{ts}/R_{fc}}$$

あらかじめ、 Q を測っておく

$$R_{fc} = \frac{T_F}{T_G} = Q \frac{\delta T}{\delta t}, Q < 1$$

収束データ取得と傾向の把握（今
回の主要な調査項目）

収束反復数 $K^{\{par\}}$ の選択

加速率 α を推定・評価

$$\approx \frac{N_{ts}}{K^{par} + \frac{N_{ts}}{R_{fc}}} = \frac{N_{ts}}{K^{par} + K_0^{par}} = \frac{N_{ts}}{\hat{K}^{par}}$$

3: 収束性調査のための解析条件

収束性に影響を与えるパラメータ群の整理

収束性

(A) 物理パラメータ

物性値 AC(dv,kval,beta)、CH(p,q,r)
T: 解析時間領域の幅、L: 解析空間両機の幅
初期値分布、境界条件

(B) 細かい積分演算(数値解析)パラメータ(fine solverパラメータ)

空間メッシュ数Nx, 幅dx (=L/Nx)
時間メッシュ幅δt (拡散数 $0.5 * dx * dx / dv \rightarrow \eta$)

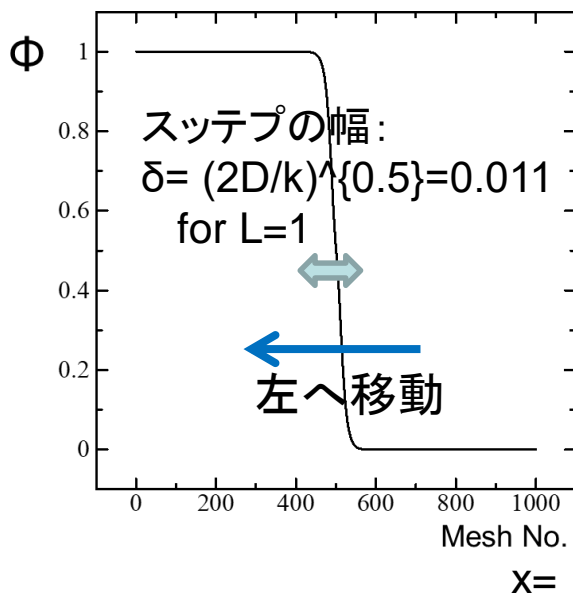
(C) 時間並列パラメータ 粗視化積分演算パラメータ (coarse solverパラメータ)

並列数 = Time Slice数: Nts
粗視化時間メッシュ幅: δT
時間粗視化率 : $R_{fc} = \frac{T_F}{T_G} = Q \frac{\delta T}{\delta t}$, $Q < 1$
粗視化時間積分法θ, ω, ε_solver, ε_nr

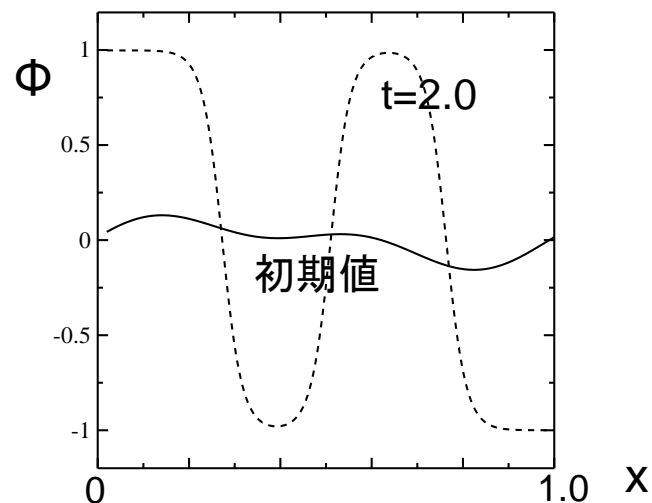
計算条件

物理パラメータ	Allen-Cahn方程式: AC	Cahn-Hilliard方程式: CH
物性値	D=1, k=16, 000, beta=-0.128 (T=0.3程度でstepは移動)	(p,q,r)=(-1,-0.001,1)
解析領域	x : [0, 1.0], t : [0, Tspan=0.5]	x : [0, 1.0], t : [0, Tspan=0.005] 最も変化の激しい時間領域を対象
初期条件 境界条件	<ul style="list-style-type: none"> ・ D=1, k=16, 000 を与えた時の定常の解析解の分布を与える ・ 傾きゼロ 	<ul style="list-style-type: none"> ・ 以下の式 ・ 傾き、3階微分ゼロ

初期値 D/k=1/16,000

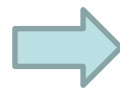


$$\Phi(x) = 0.1 \sin(2\pi x) + 0.01 \cos(4\pi x) + 0.06 \sin(4\pi x) + 0.02 \cos(10\pi x)$$



細かい積分演算(数値解析)パラメータ	Allen-Cahn方程式: AC	Cahn-Hilliard方程式: CH
空間メッシュ数 N_x	$N_x=1000 (\Delta x=0.001) \rightarrow$ 実問題の規模を想定	
時間ステップ幅	拡散数からの安定条件を $d = \delta t D / \Delta x^2 < 1/2 \rightarrow \delta t < 0.5 \Delta x^2 / D$ を基に考える	
Fine solver (Forward Euler 1次で安定計算できる安全率を調査して決定)	$\delta t = \eta * 0.5 d_0 * dx * dx / D$ $= \eta * 5.0 * 1.0E-7$ $\rightarrow 1.0E-7$ (安全率 $\eta = 0.2$) $\rightarrow 5,000,000$ ステップ	$\delta t = \eta * 0.5 d_0 * dx * dx / D$ $= \eta * 5.0 * 1.0E-7$ $\rightarrow 2.5E-11$ (安全率 $\eta = 5.0E-5$) $\rightarrow 20,000,000$ ステップ

ACの時は、メッシュ数を50から1000に増やしても安全率は $\eta = 0.2$ で変化無



$N_x=50$ の時は、安全率 $\eta = 5.0E-3$
 \rightarrow メッシュ数を増やすと時間積分を安定性が急激に減少する

時間並列 パラメータ	Allen-Cahn方程式: AC	Cahn-Hilliard方程式: CH
	並列数 (Time slice 数) Nts: 20, 40, 100, 200, 400 時間粗視化率 : $R_{fc} = \frac{T_F}{T_G} = Q \frac{\delta T}{\delta t}$, $Q < 1$ 時間ステップ幅比 $R_{fc}^{\wedge}\{0\} = \delta T / \delta t = 20, 40, 100, 400, 1000$	
	Fine solver Euler 1 st + Coarse solver CN → Q ≈ 20 BD4 → Q ≈ 30	Fine solver Euler 1 st + Coarse solver BD4 → Q ≈ 40
	Fine solver と Coarse solver 同じ (陰解法) → Q ≐ 1	

● 残差の計算

$$res^{(K^{par})} = \max_n \left[\sqrt{\frac{\sum_{i=1}^{N_{DOF}} |U_{i,n-1}^{K^{par}} - U_{i,n-1}^{K^{par}-1}|^2}{N_{DOF}}} \right]$$

10⁻⁷として考察する

● Coarse solverのQの抑制、収束の安定化

→SOR, Newton Raphosn法の収束条件の緩和

・SORの収束判定値を徐々に厳しく: `eps_itr = 1.0d0/(1.5d0**itr)`
`if(eps_sor .ge. eps_itr) then`
`eps_itr = eps_sor`
`endif`

・Newton Raphosn法の収束条件を緩くしても収束: `eps_nrm = 1.0d-3`

4: 収束データの測定結果と考察

Allen-Cahn方程式: ACの場合

時間並列パラメータの計算範囲

$\delta T / \delta t \backslash Nts$	20	40	100	200	400
20	×	×	○	×	×
40	×	×	○	×	×
100	○	○	○	×	×
400	×	×	○	○	○
1000	×	×	○	○	○



○: 実施
×: 非実施

緩和係数 $\gamma = 0.1, 0.2, 0.5, 1.0$

限界加速率 α_0 を上げつつ反復数の挙動を見る

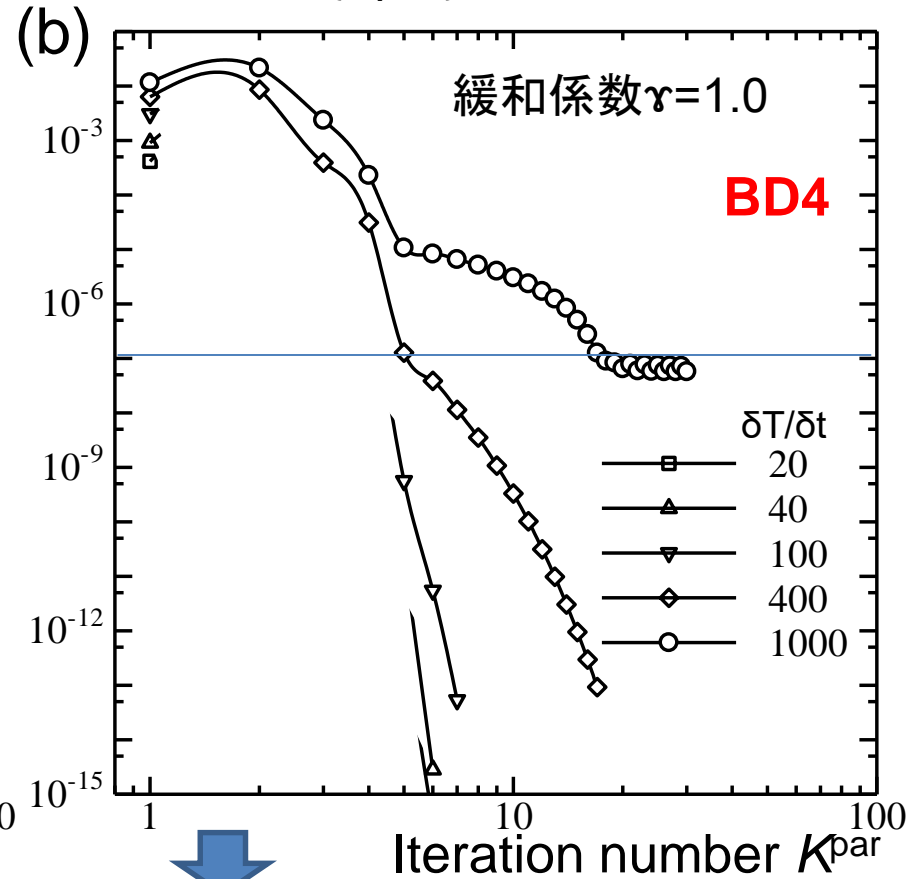
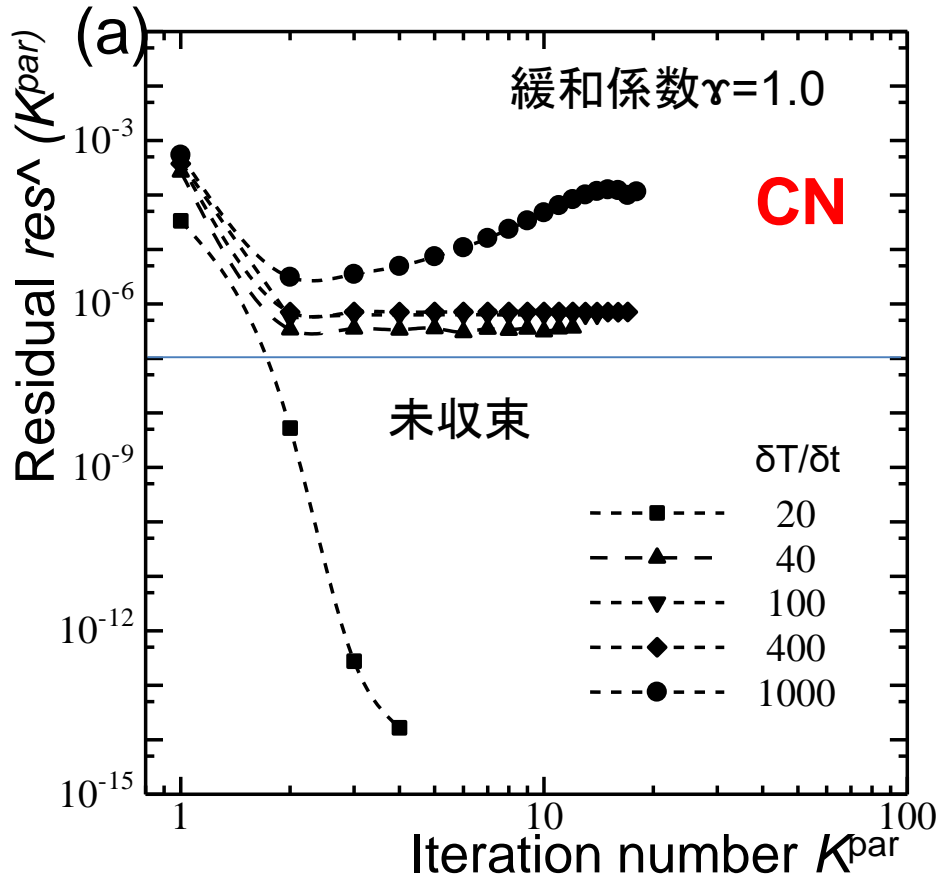
F/C solverのタイムステップ幅の比を上げる

$$\alpha_0 = \frac{N_{ts}}{1 + N_{ts}/R_{fc}} \quad R_{fc} = \frac{T_F}{T_G} = Q \left(\frac{\delta T}{\delta t} \right) \quad Q < 1$$

$$\alpha \approx \frac{N_{ts}}{K^{par} + \frac{N_{ts}}{R_{fc}}} = \frac{N_{ts}}{K^{par} + K_0^{par}} = \frac{N_{ts}}{\hat{K}^{par}}$$

並列数 $N_{ts}=100$ 一定、

収束判定 $res\{K^{par}\}=1.0E-7$



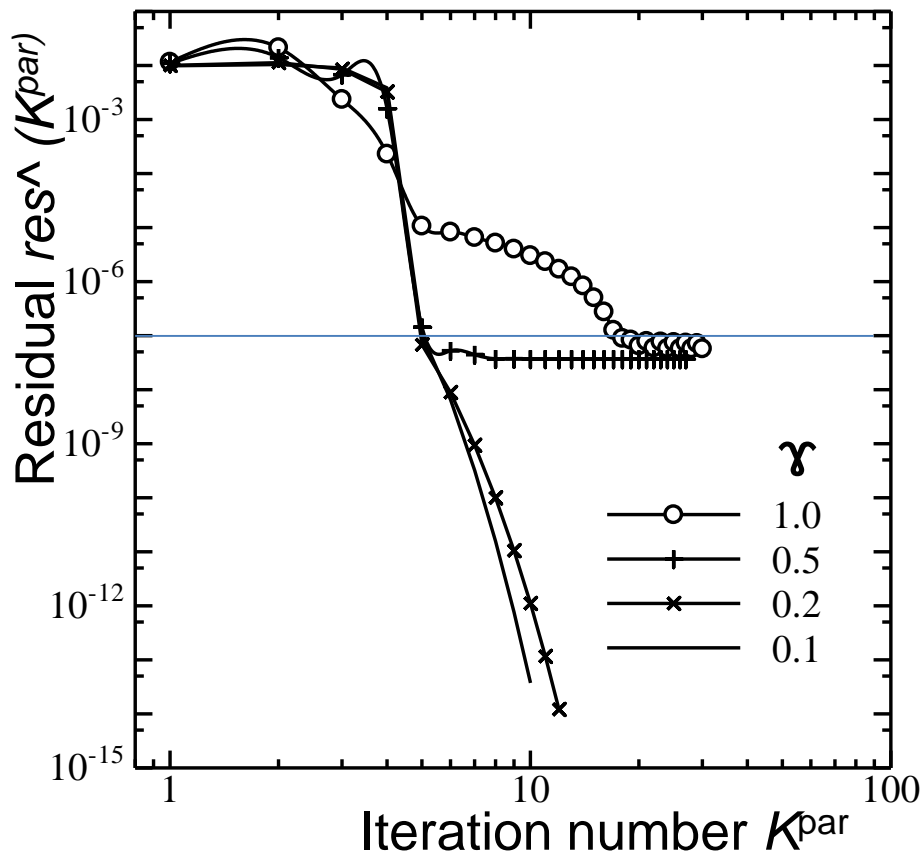
最悪値 $Q=30$
で推定

$(\delta T/\delta t)/\text{収束}K^{par} = 20/4, 40/5, 100/5, 400/6, 1000/18$
 加速率 α 推定 = 0.65, 1.25, 2.86, 7.4, 4.76

$\delta T/\delta t=400$ あたりに飽和点

修正の緩和効果を見る

並列数 Nts=100 一定/ 収束判定 $res^{Kpar}=1.0E-7$ 、 $\delta T/\delta t=1000$ を対象



$$U_n^k = F(T_n, T_{n-1}, U_{n-1}^{k-1}) \leftarrow \text{Fine solver}$$

$$+ \gamma \{ G(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^{k-1}) \}$$

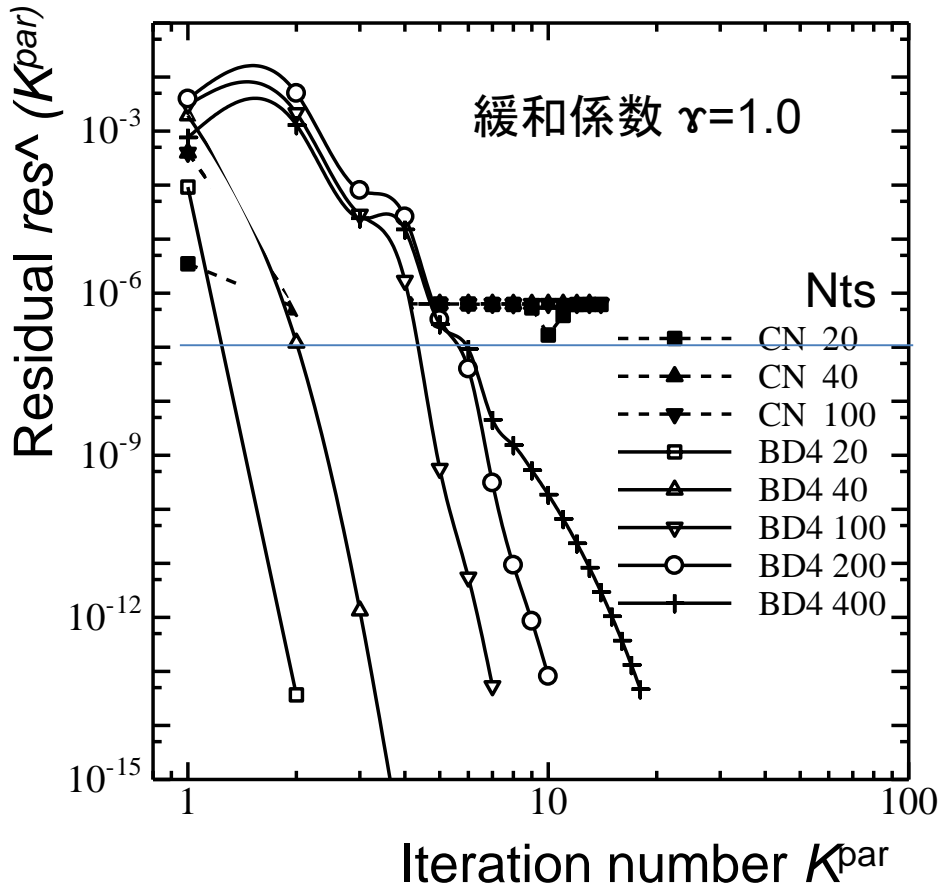
$1.0 \geq \text{緩和係数 } \gamma$ \leftarrow Coarse solver

$(\delta T / \delta t) / (\text{緩和係数 } \gamma : \text{収束 } K^{par}) = 1000 / (0.5:5, 0.2:5, 0.1:6)$
 加速率 α 推定 = 12.5, 12.5, 11.1

修正の緩和は効果あり

並列数 Nts (time slice数) の影響を見る

$\delta T/\delta t=100$ で一定/収束判定 $res^{Kpar}=1.0E-7$



BD4

Nts/収束Kpar = 20/2, 40/3, 100/5, 200/6, 400/7
加速率 α 推定 = 2.5, 2.7, 2.86, 3.0, 3.15

並列数増加の効果は小さい (微増、予想通り)

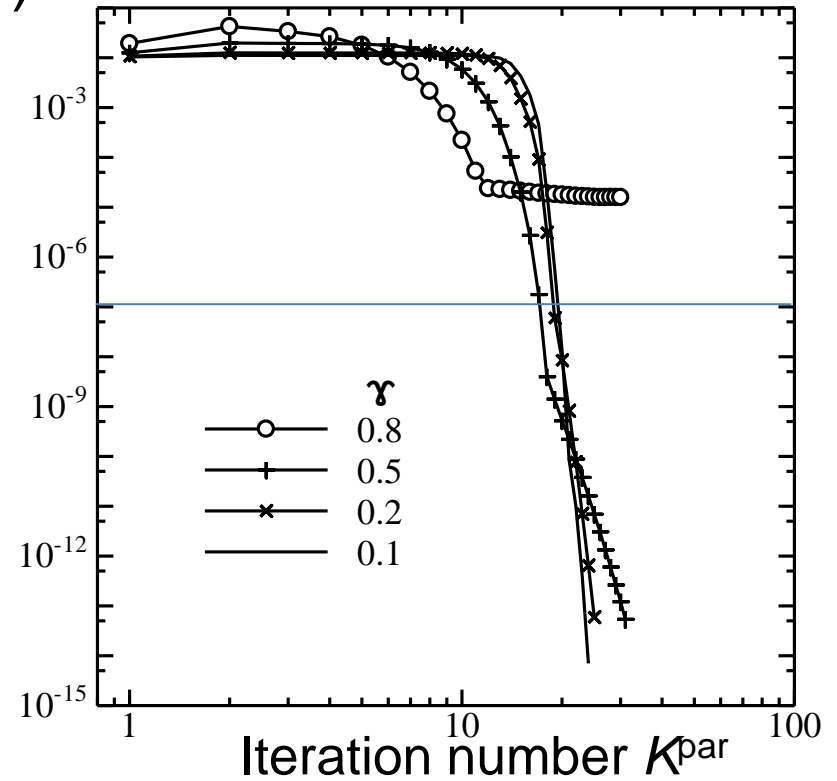
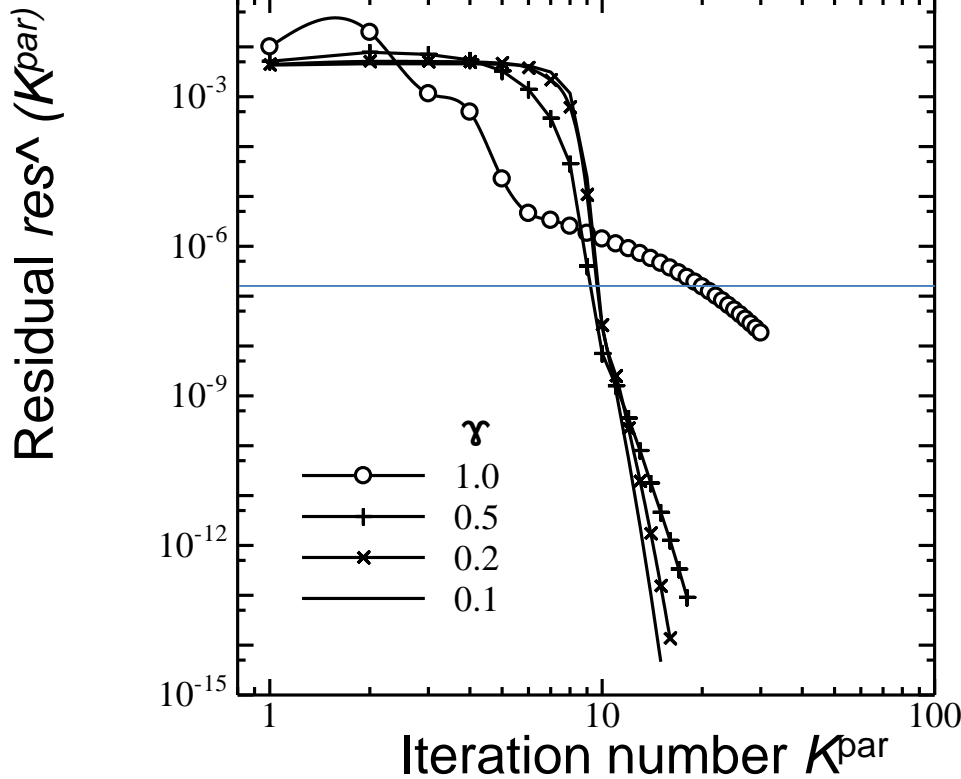
並列数 Nts と $\delta T/\delta t$ の同時変更の影響を見る

Nts=200, $\delta T/\delta t=400$

Nts=400, $\delta T/\delta t=1000$

(a) 修正の緩和あり ($\gamma < 1.0$) (b)

修正の緩和あり ($\gamma < 1.0$)



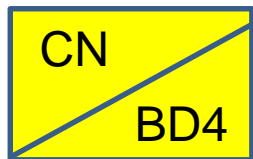
(a) $(\delta T/\delta t) / (\text{緩和係数 } \gamma : \text{収束 } K^{\text{par}}) = 400 / (0.5:10, 0.2:10, 0.1:10)$
 加速率 α 推定 = 13.3, 13.3, 13.3

(b) $(\delta T/\delta t) / (\text{緩和係数 } \gamma : \text{収束 } K^{\text{par}}) = 1000 / (0.5:18, 0.2:19, 0.1:20)$
 加速率 α 推定 = 13.3, 12.9, 12.5

やはり並列数を上げて (計算資源を投入) しても効果は小さい

加速率のまとめ AC

Nts \ δT/δt	20	40	100	200	400
20	×	×	○ α=0.7	×	×
40	×	×	○ α=1.3	×	×
100	○ α=2.5	○ α=2.7	○ α=2.9	×	×
400	×	×	○ α=7.4	○ α=13.3	
1000	×	×	○ α=12.5		○ α=13.3



○ : 実施 (CNは未収束)
 × : 非実施

$$R_{fc} = \frac{T_F}{T_G} = Q \frac{\delta T}{\delta t}, \quad Q < 1$$

加速率を効率に増加させるためには、
 並列数をほどほどに取り反復数を抑制する努力をしてRfcを上げる
 Rfcが大きい時は、Parareal法の修正計算を緩和するのは効果あり

Cahn-Hilliard方程式: CHの場合

時間並列パラメータの計算範囲

(収束の見込めるBD4を使い、加速率が見込める範囲のみを計算)

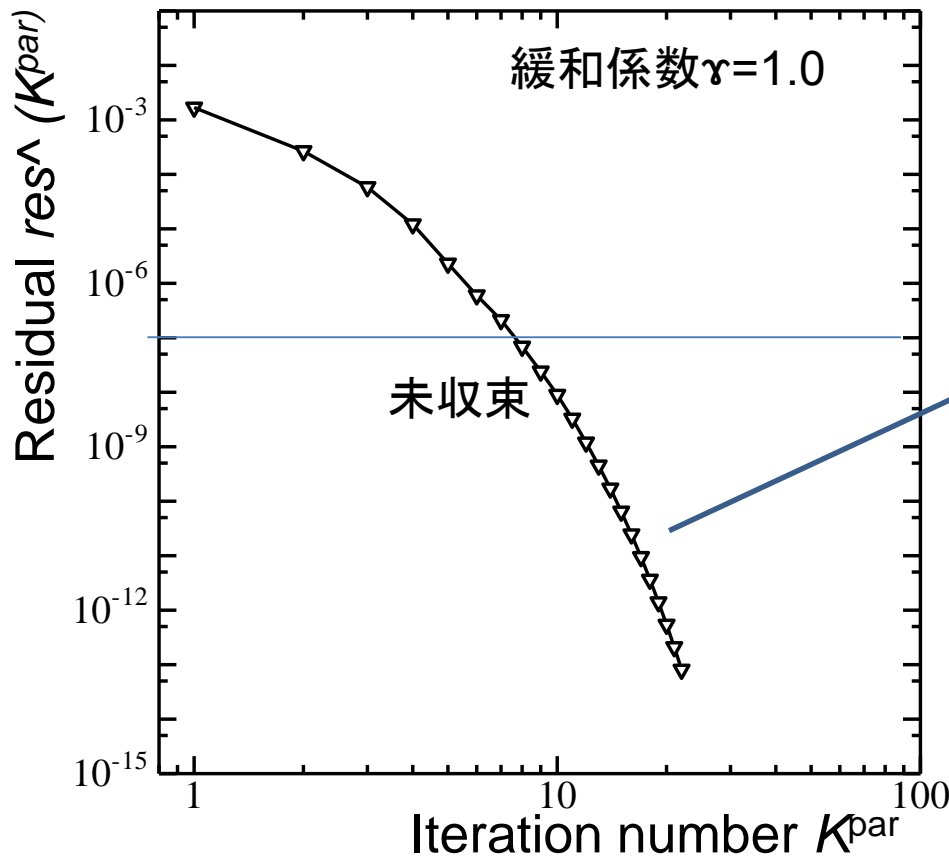
Nts \ $\delta T / \delta t$	20	40	100	200	400
20	×	×	×	×	×
40	×	×	×	×	×
100	×	×	○ $\gamma=1.0$	×	×
400	×	×	○ $\gamma=0.1, 0.2, 0.5, 1.0$		×
1000	×	×	○ $\gamma=0.1, 0.2, 0.5, 1.0$	×	×

○ : 実施

× : 非実施

$\delta T/\delta t=100$ で、反復数の挙動を見てみる

Time slice Nts=100 一定/ 収束判定 $res^{Kpar}=1.0E-7$



ACの場合に比べ
残差の下がり方は緩慢

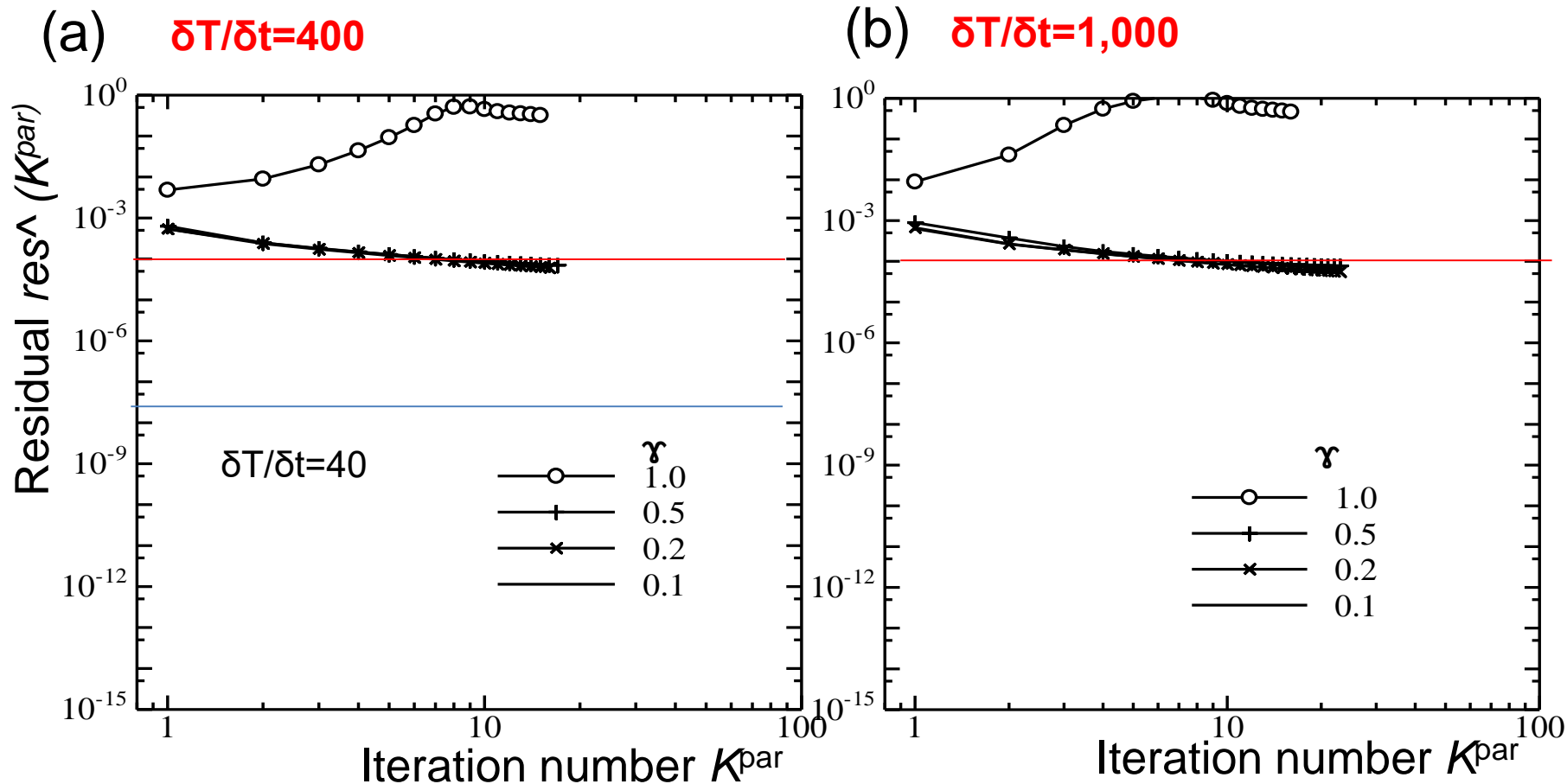
最悪値 $Q=40$
で推定

$(\delta T/\delta t)/$ 収束 $Kpar=100/8$
加速率 α 推定 = 2.1,

$\delta T/\delta t$ をさらに上げたいが . . .

$\delta T/\delta t$ をあげられるか？

上げると、収束は急速に悪化、修正の緩和の効果が薄い →
 収束判定 $res^{Kpar}=1.0E-7 \rightarrow 1.0E-4$ まで緩和



最悪値 $Q=40$
 で推定

$(\delta T/\delta t)/$ 収束 $K^{par} = 400/8 \quad 1,000/8$
 加減率 α 推定 $= 5.6, \quad 8.3$

Cahn-HilliardではPareal法収束は比較的困難 → さらに検討必要

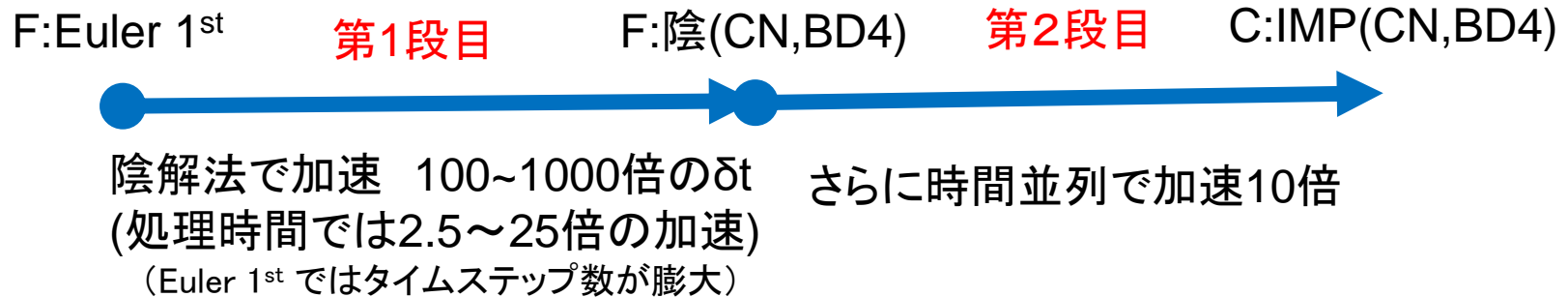
5:まとめと今後

まとめ

- Cahn-HilliardはAllen-Cahn に比べ、Parareal法の収束性が悪い
- Allen-Cahn: 10倍以上の時間方向加速が期待できる
- Cahn-Hilliard: 2倍程度か・・・
- Parareal法のフェーズフィールド法への応用は始まったばかり、今後のさらなる検討が必要

今後

- Cahn-Hilliard に対して、2段加速法を検討する



- Allen-Cahn, Cahn-Hilliard
→ 3次元化、詳細な性能データ取得・評価