

## 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 法での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, 九州大学 情報基盤研究開発センター, 819-0385 福岡県福岡市西区元岡 744,  
E-mail : iizuka.mikio.903@m.kyushu-u.ac.jp
- 小野謙二, 九州大学 情報基盤研究開発センター, 819-0385 福岡県福岡市西区元岡 744,  
E-mail : keno@cc.kyushu-u.ac.jp

Mikio Iizuka, Research Institute for Information Technology, 744 Motooka, Nishi-ku, Fukuoka, 819-0385

Kenji Ono, Research Institute for Information Technology, 744 Motooka, Nishi-ku, Fukuoka, 819-0385

The parareal method is useful for solving parabolic PDEs. Phase field equation is one of parabolic PDEs and has many applications. However, there is no report of an evaluation of performance of parareal method to phase field equation. Therefore, in this study, we apply the parareal method to a 1D phase field equation and show the convergence data to consider the efficient use of parareal method. The results of this study show the following. The parareal method is useful for the phase field equation. The number of the parareal iteration is almost 10 for the number of time slices 100. The number of parareal iteration does not increase even if we increase the time coarsening ratio to 400. Backward-difference 4th method enables an accurate convergence. In the future, we intend to widen this study for 3D phase field equation and the Cahn-Hilliard equation. Then, we will continue to evaluate the performance of the parareal method.

## 1. 初めに

現在開発が進むポストベタスケール級スーパーコンピュータは、アプリケーションに対しさらに数値上の並列性を要求すると予想されている。一方、アプリケーションの空間方向での並列性は飽和傾向にある。そのため新たな方向で並列性を生み出す計算法として、複数時間ステップに演算プロセスを割り当てる時間並列計算法が着目されている。このような背景のもと、本研究では、Lionらにより提案された Parareal 法<sup>(1)</sup>をベースに時間並列計算法の研究開発を進めている。

Parareal 法は移流方程式や波動方程式等の双曲型偏微分方程式に対しては収束性に課題があり、現在その課題を解決するための研究が活発に実施されている<sup>(2)</sup>。一方、拡散方程式等の放物型偏微分方程式に対しては Parareal 法は収束性が良く実用的なレベルにあると考える。そこで我々は、初めに大規模な拡散問題についてその性能特性の研究を行い Parareal 法の拡散問題に対する有効性を示してきた<sup>(3,4,5,6)</sup>。

フェーズフィールド法は多数の応用が期待される計算手法であり、その手法は拡散項をベースとした非線形偏微分方程式である。そのため Parareal 法が有効に適用可能と推定される。しかし、現状では、Parareal 法をフェーズフィールド法に適用したときの性能に最も影響を与える収束性については報告がない。そこで今回は、その収束性について報告する。

本講演では、初めに Parareal 法の計算法となぜ収束性が並列性能に最も影響するのか述べる。特に、Parareal 法の収束性は収束のための修正計算を行う Coarse solver の時間積分法の精度に影響されることを述べる。次にフェーズフィールド法への Parareal 法の応用法を述べる。そして、数値実験を用いて Parareal 法はフェーズフィールド法に対しても有効であること、時間積分法の選択により収束性を改善することができることを示す。最後にまとめと今後について述べる。

## 2. Parareal 法による時間並列計算法

Parareal 法では、解析を行う時間領域  $\Omega_t \in [0 : T]$  を  $N_{ts}$  個の時間領域幅  $\Delta T_n = T_n - T_{n-1}$  の time slice と呼ばれる粗い時間領域  $\Omega_n \in [T_{n-1}, T_n]$  に分割する (Fig.1)。ここで  $n$  は time slice の番号である。以下、細かい積分演算とは時間発展現象を十分正確にシミュレーションするために必要な詳細な時間ステップ幅  $\delta t$  の演算で、十分

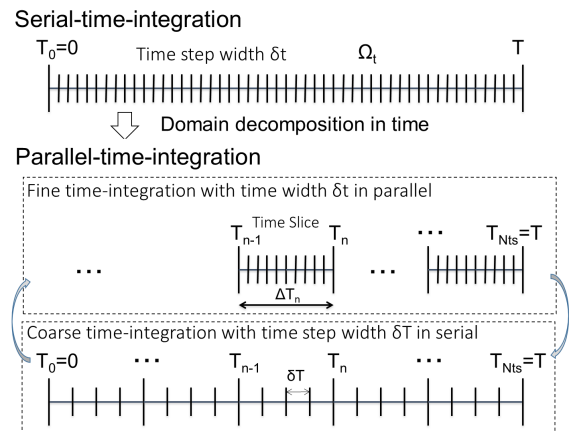


Fig. 1: Domain decomposition in time.

時間精度の高い時間積分演算のことである。粗視化積分演算とは、Parareal 法の収束計算に使われるもので、粗い時間ステップ幅  $\delta T$  での時間積分演算または粗いモデル等を用いた計算負荷の低い時間積分演算のことである。ただし、速い収束のためには高次の時間積分法が必要とされる。

Parareal 法は次のように計算される。(a) 予測計算ステップ: それぞれの time slice  $\Omega_n$  で、解くべき場の量の仮の始端値  $\mathbf{U}_{n-1}^{k-1}$  を使った時間ステップ幅  $\delta t$  の細かい積分演算  $\varphi_{\Delta T_n}(\mathbf{U}_{n-1}^{k-1})$  により初期値問題を解く。(b) 修正計算: この時、端点における前の反復との値に食い違いが出る。この食い違い量は修正すべき量であり、この食い違い量を粗視化積分演算  $\varphi_{\Delta T_n}^c(\mathbf{U}_{n-1}^{k-1})$  を用いて次の領域の始端値に伝搬させて修正量を得て修正計算を行う。そして各 time slice の端点値が収束するまで (a),(b) を繰り返す。(c) この反復計算の初期値設定: 粗い時間ステップ幅  $\delta T$  等を使い初期値問題を粗く計算する、ここで  $\Delta T \gg \delta T \gg \delta t$  である。このように異なる時間スケールの積分演算を用いるため、Parareal 法は時間方向

のマルチレベル解法となっている。細かい積分演算と粗視化積分演算を、 $\mathbf{U}_{n-1}^{k-1}$  の  $T_{n-1}$  から  $T_n$  までの積分であることを強調して、 $F(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1})$  (Fine solver, Fine popagtor ともいう)、 $G(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1})$  (Coarse solver, Coarse popagtor ともいう) と書くことが行われている。このような書き方をすると、良く知られた Parareal 法のアルゴリズムの式 (1) が得られる。

$$\mathbf{U}_n^k = F(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}) + G(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) - G(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}) \quad (1)$$

Parareal 法の並列計算では、各 time slic に演算プロセスを割り当て、各 time slic 間では time slice 番号が大きくなる方向に隣接通信を繰り返し修正量を伝搬させていく。

### 3. 並列性能向上における収束性の重要性

Parareal 法の処理手順概要とその中での処理時間を示す変数を Fig.2 に示す。Time slice 数  $N_{ts}$  と並列数  $N_p$  を

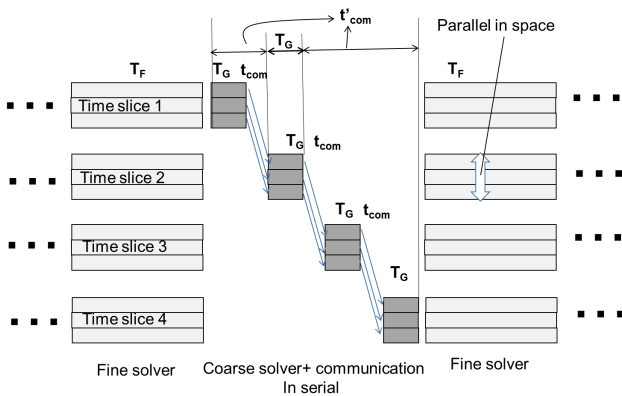


Fig. 2: Computational time in parareal

同数とし、逐次計算時間を並列計算時間で割り、さらに Time Slice 当たりの細かい積分演算の時間  $T_F$  で分子分母を割ると、Fig.2 を参考にして Parareal 法の並列加速率  $\alpha$  は式 (2) のように粗視化積分演算と細かい積分演算の時間比 (時間粗視化率)  $R_{fc} = \frac{T_G}{T_F}$ , time slice 数  $N_{ts}$ , 反復数  $K^{par}$ , 通信時間  $t_{com}$  の関数となる (7). 細かい積分演算と粗視化積分演算に同じ時間積分法を使う場合は  $R_{fc} = \frac{T_F}{T_G} = \frac{\delta T}{\delta t}$  となるが、本論文ではより一般的な方法を考える。つまり、 $R_{fc} = \frac{T_F}{T_G} = Q \frac{\delta T}{\delta t}$ ,  $Q < 1$  である。

$$\alpha \left( N_{ts}, R_{fc}, K^{par}, \frac{t_{com}}{T_G} \right) = \frac{\text{逐次計算時間}}{\text{Parareal 法の並列計算時間}} = \frac{T_F N_{ts}}{((T_F + T_S)N_{ts} + T_C N_{ts})K^{par} + T_G N_{ts}} = \frac{1}{K_{par} \left\{ \frac{1}{N_{ts}} + \frac{1}{R_{fc}} \left( 1 + \frac{1}{K_{par}} + \frac{t_{com}}{T_G} \right) \right\}} \quad (2)$$

ここで、 $T_S = -t_{com} + (t_{com} + T_G)N_{ts}$  は逐次計算時間である。  $R_{fc}$  が大きいと  $\alpha(N_{ts}, R_{fc}, K^{par}, \frac{t_{com}}{T_G}) \rightarrow N_{ts}/K^{par}$  となる。式 (2) から、性能は以下のような挙動となる。  $K^{par}$  が増加すると  $1/K^{par}$  で性能が低下する。  $R_{fc}$  の増加は逐次計算の比率を下げて性能を向上させる。一方、式 (3) から  $R_{fc}$  の増加、すなわち Coarse solver の時間ステップ幅  $\delta T$  の増加は、反復数  $K^{par}$  の増加となることに

注意が必要である。そのため Parareal 法の並列加速率を向上させるためには、 $R_{fc}$  の増加と  $K^{par}$  の減少は相反することを考慮し収束性を向上させて  $K^{par}$  の増加を抑制することが最も重要となる。このように収束性は性能向上のための最も重要な検討項目となる。

次に、その  $K^{par}$  の挙動について少し細かく述べておく。  $K^{par}$  は時間並列計算を行う方程式の特性、離散化手法、シミュレーションの問題設定 (規模等) の問題の特性、そして  $R_{fc}, N_{ts}$ , 粗視化積分演算法等の時間並列法の特性により影響を受けると推定される。この  $K^{par}$  は残差  $res$  の推定式が分かれば、Parareal 法の収束判定式  $res^{(K^{par})} \leq \varepsilon$  ( $\varepsilon$  は収束判定値) から推定可能である。現在この推定式は確立していない。現在研究中であるが分かる範囲で書けば以下のような式となる、

$$\frac{res^{(K^{par})}}{res^{(1)}} = \frac{(C'T\delta T^{mp})^{K^{par}-1}}{(K^{par}-1)!} \prod_{j=1}^{K^{par}-1} (N_{ts}-j), \quad (3)$$

ここで  $m$  はひとつの Time Slice の中の Coarse solver のステップ数、 $p$  は Coarse solver の次数である。この式は、 $\delta T < 1.0$  であるので Coarse solver の次数  $p$  が大きい、すなわち精度が高いほど Parareal 法の収束性は増加するということが示している。しかし精度を上げることにより Coarse solver 計算コストが増加してしまうと逐次計算負荷が増加し性能が劣化するので、どのような時間積分法を精度を上げるかは慎重に選択する必要がある。今回は Crank-Nicolson 法 (2 次) と後退差分法 (4 次) で精度向上を狙うが、このときそれらの計算負荷ほとんど変化しないことに注意する。

この研究の数値実験では、Parareal 法の反復数  $K^{par}$  での残差  $res^{(K^{par})}$  は次の式 (4) で評価する、

$$res^{(K^{par})} = \max_n \left[ \sqrt{\frac{\sum_{i=1}^{N_{DOF}} |U_{i,n-1}^{K^{par}} - U_{i,n-1}^{K^{par}-1}|^2}{N_{DOF}}} \right], \quad (4)$$

ここで  $N_{DOF}$  は問題の自由度である。すなわち、この残差の定義により問題規模が違っても 1 変数当たりの残差値を同等して計算が可能となる。

以上より、Parareal 法を利用する時には、問題特性 (今回は方程式中に拡散項と非線形項が混合していること)、空間離散化 (ここでは中心差分)、時間離散化 (ここでは Crank-Nicolson 法 (CN) と Backward-difference 4th (後退差分 4 次, BD4) が収束性に与える影響を把握し、それを基にスケールする並列数を勘案し適切な時間粗視化率と時間方向の並列数 (計算資源) を設定し実際の計算を実行することが必要となる。

### 4. フェーズフィールド法と Parareal 法の応用方法

ここでは、相の分布を表すフェーズフィールド変数  $\phi$  の方程式である以下の Allen-Cahn 方程式を対象とする。相間の境界領域の厚さ  $\delta$  は  $\sqrt{2D/k}$  であり、 $\beta$  は境界の移動する方向と速度を決める変数である。

$$\frac{\partial \phi}{\partial t} = RHS = D\nabla^2 \phi - k\phi(\phi - 1.0)(\phi - 0.5 + \beta), \quad (5)$$

ここで、 $\beta > 0$ : 境界が右に動く、 $\beta = 0$ : 境界は動かない、 $\beta < 0$ : 境界が左を動くパラメータ設定である。以下では空間を中心差分法で離散化する。Fine solver では時間を前進差分法 1 次で離散化する。Coarse solver で時間を Crank-Nicolson 法で離散化すると、時間発展式は次

の非線形方程式となる． $i$  番目の式は，

$$F_i^{CN}(\phi_{i-1}, \phi_i, \phi_{i+1}) = \phi_i^n - \phi_i^{n-1} - 0.5\delta T (RHS_i^n + RHS_i^{n-1}) = 0 \quad (6)$$

次に時間を Backward-difference 4th で離散化すると，時間発展の  $i$  番目の式は，

$$F_i^{BD4}(\phi_{i-1}, \phi_i, \phi_{i+1}) = 25\phi_i^n + (-48\phi_i^{n-1} + 36\phi_i^{n-2} - 16\phi_i^{n-3} + 3\phi_i^{n-4}) - 12\delta T P H F_{RHS,i}^n = 0, \\ RHS_i = \frac{D}{\Delta x^2}(\phi_{i-1} - 2.0\phi_i + \phi_{i+1}) - k\phi_i(\phi_i - 1.0)(\phi_i - 0.5 + \beta). \quad (7)$$

以上の時間発展式を Newton-Raphson 法で以下のように計算し、Coarse solver を実行する．

$$\left[ \frac{\partial F_i}{\partial \phi_{i-1}}, \frac{\partial F_i}{\partial \phi_i}, \frac{\partial F_i}{\partial \phi_{i+1}} \right] \begin{Bmatrix} d\phi_{i-1}^n \\ d\phi_i^n \\ d\phi_{i+1}^n \end{Bmatrix} = -F_i(\phi_{i-1}^{n,k-1}, \phi_i^{n,k-1}, \phi_{i+1}^{n,k-1}). \quad (8)$$

後退差分 4 次は高精度な計算であるが，4 つ前の時刻の値までを必要としている．そのため，Time slice 端点の値のみを更新して送る従来の Parareal 法では処理ができなくなる．そこで今回は  $n-2, 3, 4$  の時刻の値の変数を追加し， $n, n-1, 2, 3$  の 4 つの時刻の値の変数を同時に Parareal 法で更新していく手法を開発した．その概念図を Fig.3 に示す．

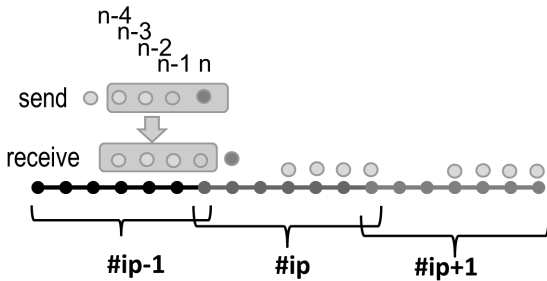


Fig. 3: Parareal edge data update for Backward-difference 4th.

## 5. 性能評価計算

ここでは，数値実験を用いて Parareal 法のフェーズフィールド法に対する有効性と時間積分法の選択の効果を，加速性能向上の主要因である収束性の観点から調べた．制御するパラメータとしては次数の異なる時間積分法 (CN, BD4) と逐次計算の比率に影響する時間粗視化率  $R_{fc}$  を使い，収束性の指標は Parareal 法の反復数  $K_{par}$  を用いた．これまでの拡散問題に対する時間並列計算法の研究から，時間並列計算での目標加速率 10 を達成するためのパラメータの目安は，Time slice 数 (並列数)  $N_{ts} = 100$  と時間粗視化率  $R_{fc} = 100$  程度である．そのためこの研究ではこのパラメータ値を中心に Time slice 数と時間粗視化率をふった計算を行った．その計算条件の全体を表-1 に示す．

物性値は，2 相の境界を表す変数  $\phi$  の分布が急峻なるように， $D/k = 1/16,000$  とした．その時の境界の厚み

Tab. 1: Parameters used in the numerical test.

D and k	1.0 and 16,000
Time span $[0, T]$	$[0, 0.5]$
Space span $[0, L]$	$[0, 1.0]$
Number of time slices $N_{ts}$	10, 50, 100
$\delta t$ for fine solver	$1.0 \times 10^{-7}$ : $5 \times 10^6$ steps
$\delta T$ for coarse solver	$2.0 \times 10^{-6}$ ( $R_{fc} = 20$ ) $4.0 \times 10^{-6}$ ( $R_{fc} = 40$ ) $1.0 \times 10^{-5}$ ( $R_{fc} = 100$ ) $2.0 \times 10^{-5}$ ( $R_{fc} = 200$ ) $4.0 \times 10^{-5}$ ( $R_{fc} = 400$ ) $10.0 \times 10^{-5}$ ( $R_{fc} = 1,000$ )
Convergence test (SOR)	$res^{k^{SOR}} \leq 10^{-6}$
Convergence test (Newton-Raphson)	$res^{NR} \leq 10^{-5}$

は  $\delta = (2D/k)^{0.5} = 0.011$  程度である．計算の初期値は，この  $D/k$  値を使った式 (5) の定常状態の解を使った．その  $\phi$  の初期値分布を Fig.(4) に示す．

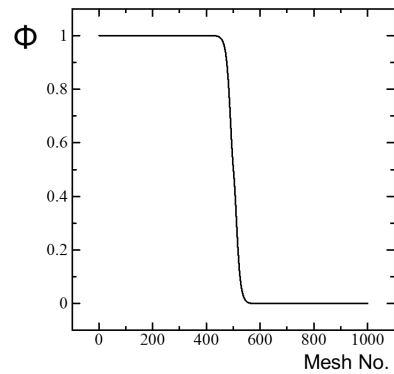


Fig. 4: Initial value of phase field variable  $\phi$  for parareal calculation.

### 5.1 時間粗視化率の収束性への影響

Time slice 数を  $N_{ts} = 100$  に固定し，時間粗視化率を  $R_{fc} = 20 \sim 1,000$  と変化させた．その結果を Fig.(5) に示す．収束判定の残差を  $res^{(K^{par})} = 1.0^{-6}$  のオーダーとすれば，Crank-Nicolson 法を使えば  $R_{fc}=400$  まで ( $K^{par}$ ) は数回で収束する．Backward-difference 4th を使う場合は  $R_{fc}=400$  まで ( $K^{par}$ ) は数回で収束し，収束が飽和することもない．これは  $res^{(K^{par})}$  のオーダーをより減じるためには Backward-difference 4th が有効であることを示している．

### 5.2 Time slice 数の収束性への影響

時間粗視化率を  $R_{fc} = 100$  に固定し，Time slice 数を  $N_{ts} = 20, 50, 100$  と変化させた．その結果を Fig.(5) に示す．前記と同様に収束判定の残差を  $res^{(K^{par})} = 1.0^{-6}$  のオーダーとすれば，Crank-Nicolson 法と Backward-difference 4th は共に  $K^{par}$  は数回で収束する．但し Crank-Nicolson 法の場合は Time slice 数を減じても収束性の改善は見られない．一方，Backward-difference 4th を使う場合は，ime slice 数を減じることは収束性の改善になる．また， $res^{(K^{par})}$  のオーダーをより減じるために

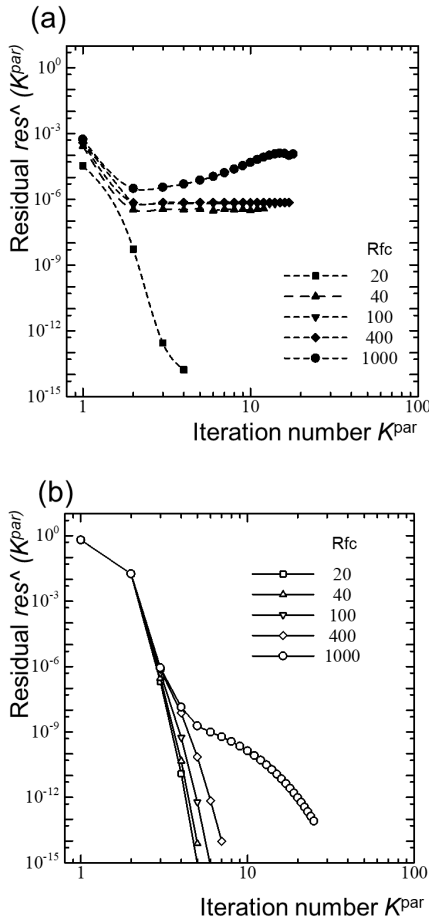


Fig. 5: Residual during the parareal method iteration using (a) the Crank Nicolson method and (b) the Backward-difference 4th for the case in which the number of time slices  $N_{ts}=100$  is constant and the time coarsening factor  $R_{fc}$  increases.

は Backward-difference 4th が必要となることは前記と同様である。

### 5.3 Coarse solver の時間積分法の収束性への影響

上記の結果から、Coarse solver の時間積分法の収束性への影響は次のようである。Parareal 法の反復初期では後退差分法 4 次比 Crank-Nicolson 法の方が残差  $res^{(K^{par})}$  は小さい。特に収束判定の残差を  $res^{(K^{par})} = 1.0^{-6}$  のオーダーとすれば Crank-Nicolson 法と Backward-difference 4th の場合で共に  $K^{par}$  は数回で収束してしまう。しかし、 $res^{(K^{par})} = 1.0^{-6}$  のオーダーよりも残差を下げるためには Backward-difference 4th が必要となる。一方、Backward-difference 4th の場合は Parareal 法の反復初期の残差  $res^{(K^{par})}$  は大きく、収束が進むまでには数回の反復数を必要とする。

### 6. まとめと今後

Parareal 法は、拡散項をベースとした非線形偏微分方程式によるフェーズフィールド法に対して有効と考える。Time slice 数  $N_{ts}$  (並列数) が 100 で Parareal 法の反復数は収束判定の残差を  $res^{(K^{par})} = 1.0^{-6}$  のオーダーで数回程度であり、並列という意味では 10 倍の加速の可能性が推測される。一方現在、高速化調整無しの状態 Coarse solver の Parareal 法 1 反復計算時間は、Fine solver の 1 ステップの計算時間の 20 倍～30 倍 (SOR, Newton-Raphson 法の判定基準  $10^{-6}$ ) であり高速化調整、ソルバーの改良が必要である。

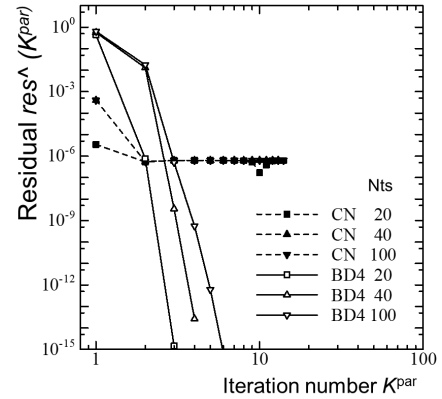


Fig. 6: Residual during parareal method iteration using the Crank Nicolson method and Backward-difference 4th for the case in which the time coarsening factor  $R_{fc} = 100$  is constant and the number of time slices  $N_{ts}$  increases.

今後は、Allen-Cahn 方程式の 1 次元コードを 3 次元に拡張し、保存則である Cahn-Hilliard 方程式を導入し、高速化調整を行い詳細な処理時間等の性能データを取得・分析し課題を洗い出し検討を進める予定である。

### 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」の一環として実施したものです。本論文の結果 (の一部) 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)。

### 参考文献

- (1) J.-L. LIONS, Y. MADAY, AND G. TURINICI, “A parareal in time discretization of PDEs”, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp.661-668.
- (2) M. J. Gander and S. Vandewalle, “Analysis of the parareal time parallel time integration method”, SIAM Journal on Scientific Computing, vol.29, no.2 (2007), pp.556-578.
- (3) 今村成吾, 飯塚幹夫, 小野謙二, 横川三津夫, “大規模拡散問題に対する Parareal 法による時間並列計算の性能挙動”, 日本流体力学学会 第 30 回数値流体力学シンポジウム, タワーホール船堀 (東京都江戸川区船堀), (2016) .
- (4) 今村成吾, 小野謙二, 飯塚幹夫, 横川三津夫, “Parareal 法と領域分割法による拡散問題での時空間並列性能評価”, 研究報告ハイパフォーマンスコンピューティング (HPC), (2016) .
- (5) 今村成吾, 小野謙二, 飯塚幹夫, 横川三津夫, “EVALUATION OF THE PARAREAL METHOD FOR LARGE-SCALE DIFFUSION PROBLEMS”, ELSEVIER, 出版予定.
- (6) 今村成吾, 飯塚幹夫, 小野謙二, 横川三津夫, “大規模時空間並列計算での Parareal 法の性能評価”, 計算工学会論文集, (2017).
- (7) M. L. Minion, “A hybrid parareal spectral deferred corrections method.” Communications in Applied Mathematics and Computational Science, 5.2 (2011), pp.265-301.