

筑波大学大学院博士課程
理工情報生命学院
システム情報工学研究群修士論文

残差に基づく有限要素法の数値解補正法の検討

YU HUAIZHI

修士（工学）

（構造エネルギー工学学位プログラム）

指導教員 山本 亨輔

2022年 3月

概要

分散コンピューティングに適用することができる有限要素法陽解法数値解の数値修正検討を行う。有限要素法の中で汎用性が高く一般的な手法である重み付き残差法では、任意の重み関数に対して、重み付き残差の平均が常にゼロ、という条件を満たす数値解が算出されており、必ずしも残差を最小化しているわけではない。そこで、次元移流方程式を例として、重み付き残差法により求めた数値解の残差を評価する。次に、残差値を目的関数として、遺伝的アルゴリズムに基づいて数値解を更新し高精度化の可能性を検討した。残差に基づく数値解の更新はルンゲ=クッタ法数値解にある程度の精度改善が確認できた。但し、計算コストが大きく、効率の面では改善の余地がある。

目次

第 1 章	はじめに	1
第 2 章	ガラーキン重み付き残差法の概要	2
2.1	重み付き残差法の概要	2
2.2	ガラーキン法に基づく計算手法	2
2.3	数値シミュレーションの概要	5
2.4	数値シミュレーション結果と考察	6
2.4.1	前進オイラー法を適用した結果	6
2.4.2	ルンゲ=クッタ法を適用した結果	7
2.4.3	後退オイラー法を適用した結果	9
2.5	残差分析	10
2.6	まとめ	13
第 3 章	遺伝的アルゴリズムによる数値解の修正	14
3.1	遺伝的アルゴリズムの概要	14
3.2	本研究への適用	18
第 4 章	結果と考察	22
4.1	40 分割における各世代目の修正結果	23
4.2	100 分割における各世代目の修正結果	24
4.3	修正結果に関する考察	24
第 5 章	まとめ	27
5.1	確認した結果と得られた知見	27
5.2	今後の予定	27
	謝辞	28
	参考文献	29
付録 A	選点法と最小二乗法の概要	29

目次

2.1	理論解	5
2.2	前進オイラー法数値結果	6
2.3	ルンゲ=クッタ法数値結果 (5 ステップ)	7
2.4	ルンゲ=クッタ法数値結果 (1000 ステップ)	8
2.5	後退オイラー法数値結果	9
2.6	理論解の残差状況	11
2.7	$t = 0.01$ ($STEP : 1$) の時における残差および数値解	11
2.8	$t = 0.10$ ($STEP : 10$) の時における残差および数値解	12
2.9	$t = 1.00$ ($STEP : 100$) の時における残差および数値解	12
3.1	一点交叉	16
3.2	二点交叉	17
3.3	一様交叉	17
3.4	修正フローチャート	19
4.1	修正される数値解と理論解の比較	22
4.2	40 分割修正における $1e1, 1e2$ 世代目の修正結果	23
4.3	40 分割修正における $1e3, 1e4$ 世代目の修正結果	23
4.4	40 分割修正における $1e5, 1e6$ 世代目の修正結果	24
4.5	100 分割修正における $1e1, 1e2$ 世代目の修正結果	24
4.6	100 分割修正における $1e3, 1e4$ 世代目の修正結果	25
4.7	100 分割修正における $1e5, 1e6$ 世代目の修正結果	25
4.8	40 分割における残差と Norm2 誤差の関係	26
4.9	100 分割における残差と Norm2 誤差の関係	26

第1章 はじめに

有限要素法の基本的な概念は、航空機の構造計算に対する切実な需要から生まれた。弾性問題を数値的に解くために A.Hrennikoff (1941) [1] は有限要素法の原型となるフレームワーク法を初めて提唱した。R.Courant (1943) [2] は偏微分方程式と変分問題を論じ、応用として棒のねじり問題に三角メッシュを適用した。M.J.Turner ら (1956) [3] はトラス、平板、ボックスビームの剛度マトリックスを算出し、有限要素法を大きく発展させた。「有限要素」という名前は R.W.Clough (1960) [4] が正式に提案した。その後、計算機の性能向上により、有限要素法は最小ポテンシャルエネルギーの原理に基づいて、自動車、航空機、建築物、橋梁など構造物の変形や応力解析によく使用されるようになった。一方、重み付き残差法 (WRM) の提案により、有限要素法は、最小ポテンシャルエネルギーの原理に依存せずに、熱、流体、電磁場、浸透流など非構造問題の数値計算にも適用できるようになった。今日では、有限要素法は幅広い工学的問題を数値的にとくことができる強力なツールである。

一方、有限要素法を用いて、地球全体の気候変化や降水のような大規模シミュレーションを行うには、単一の計算機では難しいである。よって、今後は分散コンピューティングの利用が増えると考えられる。分散コンピューティングとはネットワークを介して、プログラムの個々の部分が同時並行的に複数のコンピュータ上で実行する計算方法である。

重み付き残差法を用いて、偏微分方程式を解くときに、じかんしんてんについては差分法を用いるのが一般的である。時間部分項を離散的に解く差分法を大きく陽解法と陰解法に分けられる。陽解法は、安定性が低く、発散しやすいが、並列計算が可能という特徴がある。一方、陰解法では、同条件の陽解法より安定性が高いが、分散コンピューティングの実行が困難である。陽解法の安定性改善できれば、分散コンピューティングへの適用性が高まり、有限要素法で解ける問題も増えると期待できる。

以上から、モンテカルロ法により精度改善したいため、本研究では、1次元の移流方程式の重み付き残差法かつ前進差分法（陽解法）により求めた数値解に対して遺伝的アルゴリズムを使用し、残差に基づいた修正可能性を検証した。第2章では、重み付き残差法による数値解の計算概要と数値シミュレーションの結果および数値解の残差について述べる。第3章では、遺伝的アルゴリズムの概要と最適化プロセスを説明する。第4章では、遺伝的アルゴリズムの最適化結果を考察し、第5章に、まとめと今後の予定を示す。

第2章 ガラーキン重み付き残差法の概要

2.1 重み付き残差法の概要

重み付き残差法は、偏微分方程式の近似解を求めるための工学的道具である。重み付き残差法は三つのステップから成る。まず、式 (2.1) に示すように、基底関数 ψ_i と係数 a_i により、境界条件を満たす試行関数 \hat{u} を構築する。

$$\hat{u} = \psi_0 + \sum_{i=1}^n a_i \psi_i \quad (2.1)$$

次に、試行関数を支配方程式に代入する。試行関数 \hat{u} が厳密解ではないため、支配方程式には残差 R が生じる。最後に、残差 R と重み関数 ϕ の内積を領域 Ω 内で積分する。積分の結果を 0 と仮定すると、試行関数の係数 a_i に関する線形な連立方程式を得る。

重み関数の選択には選点法、最小二乗法、ガラーキン法などの方法がある。選点法と最小二乗法の概要は付録 A に説明する。

ガラーキン法

ガラーキン法とは、基底関数 ψ_i と同様な重み関数 ϕ_i を使用して、計算を行う方法である。

$$\phi_i = \psi_i \quad (2.2)$$

本手法は Boris Galerkin により提唱された試行関数の選択法である。殆どの場合で、ガラーキン法は精度が高く、多くの商用数値計算ソフトウェアに採用されている。本研究も重み付き残差法としてガラーキン法を使う。

2.2 ガラーキン法に基づく計算手法

移流方程式を例として、重み付き残差法の基本的な計算方法を示す。まず、未知なスカラー関数 $u(x)$ に関する一次元移流方程式は式 (2.3) となる。

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (2.3)$$

まず、試行関数 $\hat{u}(x)$ を離散化し、離散点の値（係数） a_i と基底関数 $\psi_i(x)$ の内積になる。アインシュタインの縮約記法により、以下のように整理できる。

$$\hat{u}(x) = \sum a_i \psi_i(x) = a_i \psi_i(x) \quad (2.4)$$

また、式(2.3)に式(2.4)を代入して、その時に生じる残差を R と表す。

$$\frac{\partial a_i \psi_i(x)}{\partial t} + c \frac{\partial a_i \psi_i(x)}{\partial x} = R \quad (2.5)$$

式(2.5)により、 $R = 0$ の時を直接算出ができないため、重み付き残差法を用いて、すべての残差値に重み関数 ϕ_i を与え、領域 $0 \leq x \leq L$ 内で積分した時の値を0と仮定するという条件で計算を行う。

$$\int_0^L \phi \left(\frac{\partial a_i \psi_i(x)}{\partial t} + c \frac{\partial a_i \psi_i(x)}{\partial x} \right) dx = 0 \quad (2.6)$$

また、ガラーキン法を用いたので、重み関数 ϕ_i は基底関数 ψ_i と同様である式(2.6)。式(2.6)は

$$\left[\int_0^L \psi_i \psi_j dx \right] \frac{\partial \mathbf{a}}{\partial t} + \left[c \int_0^L \psi_i \frac{\partial \psi_j}{\partial x} dx \right] \mathbf{a} = 0 \quad (2.7)$$

に整理することができる。さらに、

$$\mathbf{M} = \int_0^L M_{ij} dx = \int_0^L \psi_i \psi_j dx \quad (2.8)$$

$$\mathbf{C} = \int_0^L C_{ij} dx = c \int_0^L \psi_i \frac{\partial \psi_j}{\partial x} dx \quad (2.9)$$

と定義すると式(2.7)は

$$\mathbf{M}\dot{\mathbf{a}} + \mathbf{C}\mathbf{a} = \mathbf{0} \quad (2.10)$$

と整理できる、式(2.10)は有限要素式と呼ぶ。基底関数の選択により、 \mathbf{M} 、 \mathbf{C} マトリックスの中身が異なりが、マトリックスの各成分は既知となる。ただし、 $\dot{\mathbf{a}}$ の時間離散化法によって、結果の精度や計算負荷は大きく影響を受ける。時間離散化には、主に陽解法と陰解法がある。陽解法には前進オイラー法やルンゲクッタ法などの方法が、陰解法には後退オイラー法などの方法がある。それぞれの計算手法を以下に示す。

陽解法

前進オイラー法 (Forward Euler Method) 前進オイラー法を採用し、計算を行う場合には基本的には以下の漸化式を利用する。初期条件を用いて各時間ステップの解が算出できる。

$$x(t + \Delta t) = x(t) + \dot{x}\Delta t \quad (2.11)$$

また、有限要素式(2.10)の $\dot{\mathbf{a}}$ を漸化式(2.11)に代入すると

$$\mathbf{a}(t + \Delta t) = (\mathbf{E} - \Delta t \mathbf{M}^{-1} \mathbf{C}) \mathbf{a}(t) \quad (2.12)$$

が得られる。これにより前の時間ステップの情報を用いて、次の時間ステップの近似解を算出できる。しかし、前進オイラー法を実際に使用すると不安定性が強く、著し精度低減を生じる。

ルンゲ=クッタ法 (Runge-Kutta methods) 実際の数値シミュレーションでは、前進オイラー法の精度低下を補うために、高次ルンゲ=クッタ法がよく使われる。ここで、次の初期値を考える。

$$y' = f(t, y) \quad , \quad y(t_0) = (y_0) \quad (2.13)$$

y' は $f(t, y)$ の勾配を示す。ガラーキン法の近似解 \mathbf{a} の1時刻ステップの近似をさらに複数のステップに分割し、それぞれの勾配を計算する。最後に勾配の重み付けこの時刻ステップの勾配を算出する。

$$\mathbf{a}(t_{n+1}) = \mathbf{a}(t_n) + \frac{6}{h}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

$$t_{n+1} = t_n + \Delta t$$

ここで、 $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4$ はそれぞれ以下のように計算する。

$$\mathbf{k}_1 = f(t_n, \mathbf{a}),$$

$$\mathbf{k}_2 = f\left(t_n + \frac{h}{2}, \mathbf{a} + \frac{h}{2}\mathbf{k}_1\right)$$

$$\mathbf{k}_3 = f\left(t_n + \frac{h}{2}, \mathbf{a} + \frac{h}{2}\mathbf{k}_2\right)$$

$$\mathbf{k}_4 = f(t_n + h, \mathbf{a} + h\mathbf{k}_3)$$

式 (2.10) にある \mathbf{M}, \mathbf{C} が既知、さらに、 $y' = f(t, \mathbf{a})$ が各空間位置における勾配から成るマトリックスであるため、 $\dot{\mathbf{a}}$ で近似される。そこで、以下の式を利用して、ルンゲ=クッタ法による時間離散化する。

$$f(t, \mathbf{a}) = \dot{\mathbf{a}} = -\mathbf{M}^{-1}\mathbf{C}\mathbf{a} \quad (2.14)$$

陰解法

後退オイラー法 (Back Euler Method) 後退オイラー法を採用し、計算を行う場合には前進オイラー法と類似な漸化式を利用する。時刻 $t - \Delta t$ の値から時刻 t の値を算出する。

$$\mathbf{a}(t) = \mathbf{a}(t - \Delta t) + \dot{\mathbf{a}}\Delta t \quad (2.15)$$

また、有限要素式 (2.10) の $\dot{\mathbf{a}}$ を漸化式 (2.15) に代入すると

$$\mathbf{a}(t) = (\mathbf{M} + \Delta t\mathbf{C})^{-1}\mathbf{M}\mathbf{a}(t - \Delta t) \quad (2.16)$$

前進オイラー法では \mathbf{M} の逆行列を計算すれば次の時刻の値を算出ができるに対して、後退オイラー法の場合では、 $(\mathbf{M} + \Delta t\mathbf{C})$ の逆行列を計算する必要がある。そのため、後退オイラー法では空間並列性を持たないことを確認できた。

2.3 数値シミュレーションの概要

本節では，矩形波の 1 次元移流問題に連続ガラーキン法と陰解法または陽解法を組み合わせさせて適用し，数値解を求めて，各手法の精度を検証する．

表 2.1: 計算条件

境界条件	周期境界条件
基底関数 ψ	一次ラグランジュ型
移流速度 c	1.00
計算領域 L	$0 \leq x \leq 1$
空間刻み幅 Δx	0.0001
時間刻み幅 Δt	0.0001

CFL (Courant-Friedrichs-Lewy) 条件 表 2.1 の計算条件をクーランの安定条件に代入すると

$$C = \frac{c\Delta x}{\Delta t} = 1 \quad (2.17)$$

となる． $C \leq 1$ であれば，安定条件を満たす．

理論解 理論解を図 2.1 に示す．初期値 $t = 0$ を黒線に表す．移流速度 $c = 1$ により， $t = 500$ の時， x 軸正方向に 0.5 移動する．周期条件のため，1001 時間ステップ毎に初期値に戻る．

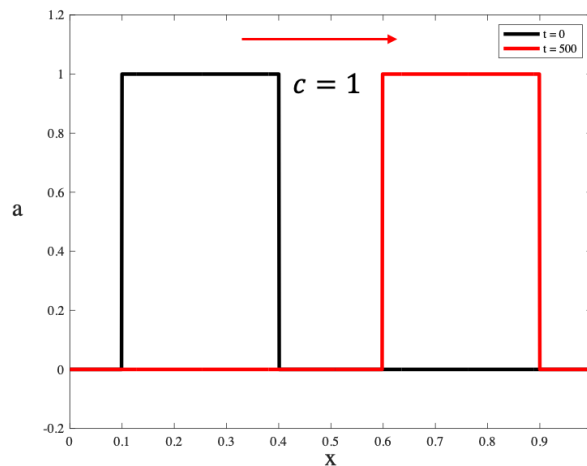


図 2.1: 理論解

2.4 数値シミュレーション結果と考察

連続ガラーキン重み付き残差法を用いて、前進オイラー法、ルンゲ=クッタ法、後退オイラー法それぞれ9000時刻ステップ、10周期までにおいて数値シミュレーションを行った。また、次式により、数値解と理論解の誤差を評価する。本指標が小さい程、数値シミュレーションの精度が良いことを表す。

$$\text{Norm2} = \sum_{i=1}^K \sqrt{(u_i - a_i)^2} \quad (2.18)$$

2.4.1 前進オイラー法による時間離散

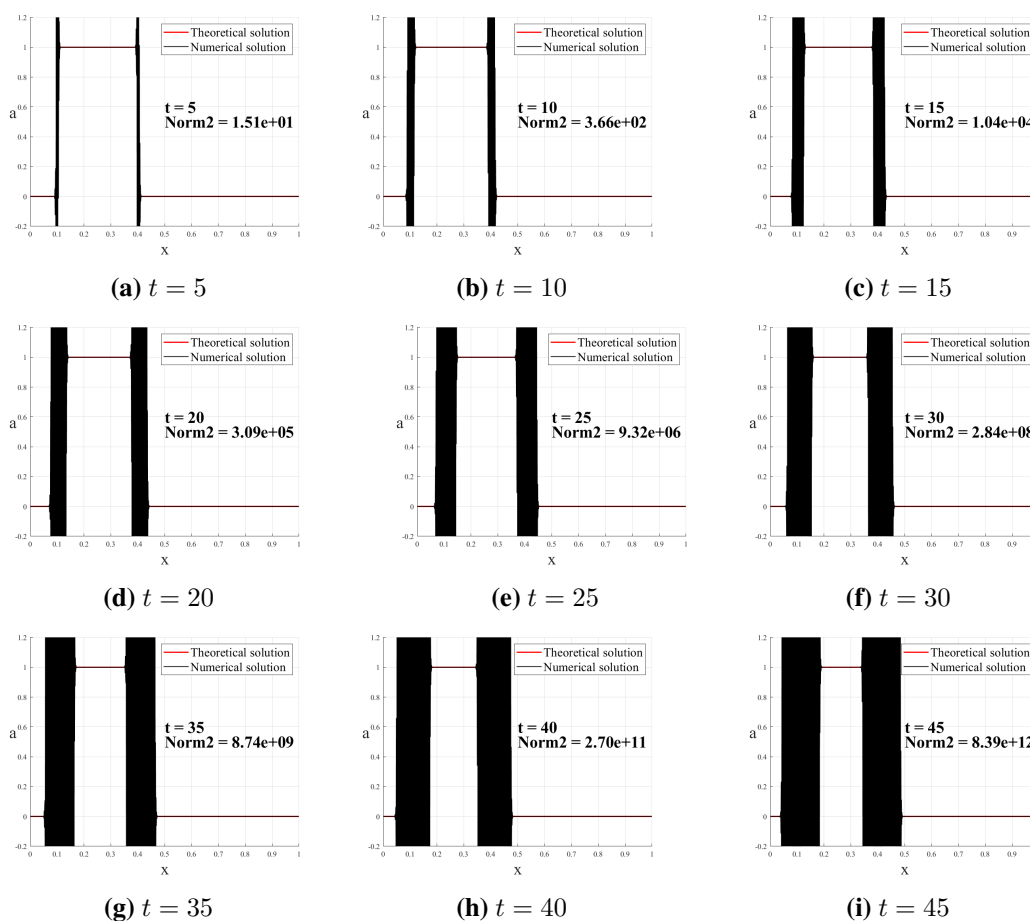


図 2.2: 前進オイラー法を適用した数値シミュレーション ($\Delta t = \Delta x = 0.0001$)

矩形波の移流では矩形を保ちつつ伝播していくべきである。しかし、図 2.2 に、5 時刻ステップずつの数値解を示す。前進オイラー法を採用した場合には、クーラン条件に満たすことにかかわらず、非常に激しい発散現象が発生する。Norm2 の値が時刻ステップが進む程急速に増加する。その理由として、前進オイラー法を適用した時、不連続点における、次の時刻の計算で誤差が蓄積しやすいことを確認した。この計算プロセスでは算出される数値解の正負が理論解と逆転することも多く、そのため、計算の初期段階、非常に激しい数値振動が発生する。

2.4.2 ルンゲ=クッタ法による時間離散

前進オイラー法と同じ計算条件でルンゲ=クッタ法に基づく数値シミュレーションを実施する。その結果を図 2.3 に示す。ここでは、数値振動の発生が確認できるが、数値振動はある空間位置に限定的に生じており、前進オイラー法のように、振幅が増幅されることがないこ

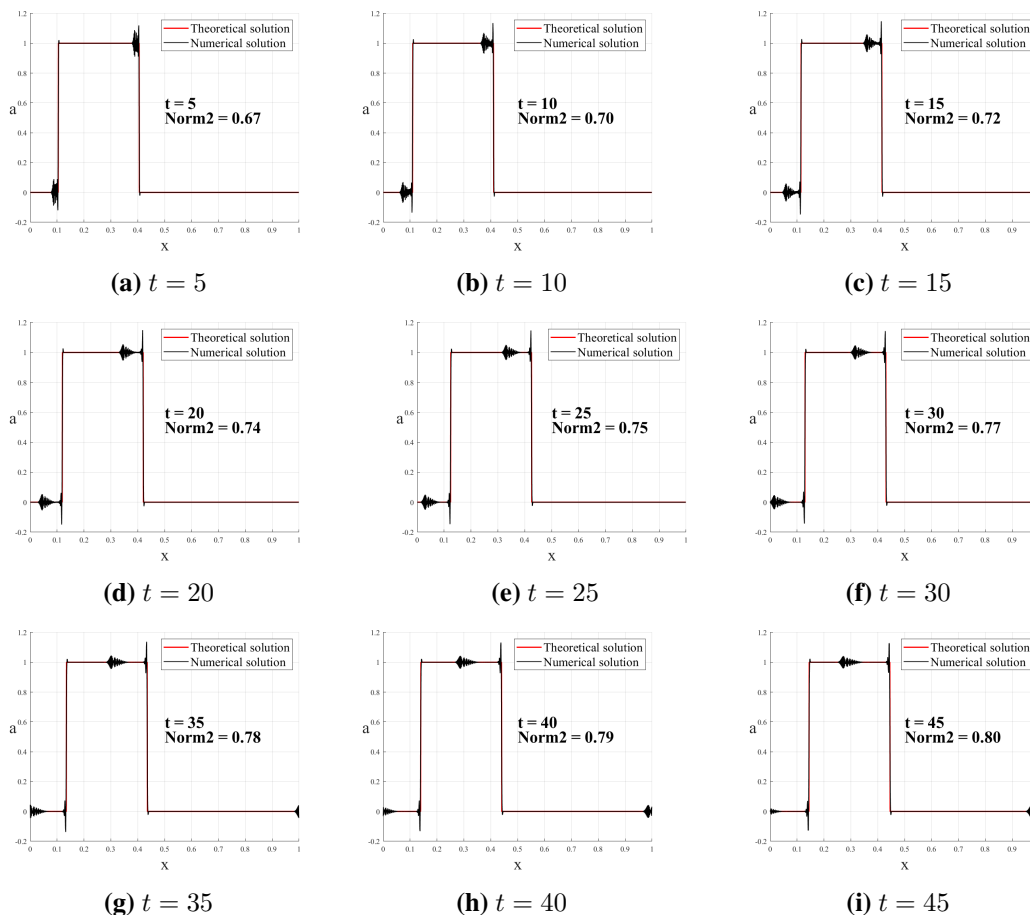


図 2.3: ルンゲ=クッタ法を適用した数値シミュレーション結果 (5 ステップ出力)

とが確認できる。一方、2-ノルム誤差が時間進行に伴い増加する傾向があることも確認できた。前進オイラー法と比較するとルンゲ=クッタ法の誤差は小さく、収束性も高い。

ルンゲ=クッタ法が陽的でありながら高い安定性を示す原因は、ルンゲ=クッタ法が4次項まで考慮したテイラー式を用いることで、4次精度になっているためであると考えられる。今回の数値シミュレーションモデルでは、不連続点周辺において、理論解が階段関数となるため、近似解の精度は近似関数の次数に比例する。そのため、4次精度であるルンゲ=クッタ法が一次精度の前進オイラー法より精度が高くなったと説明できる。

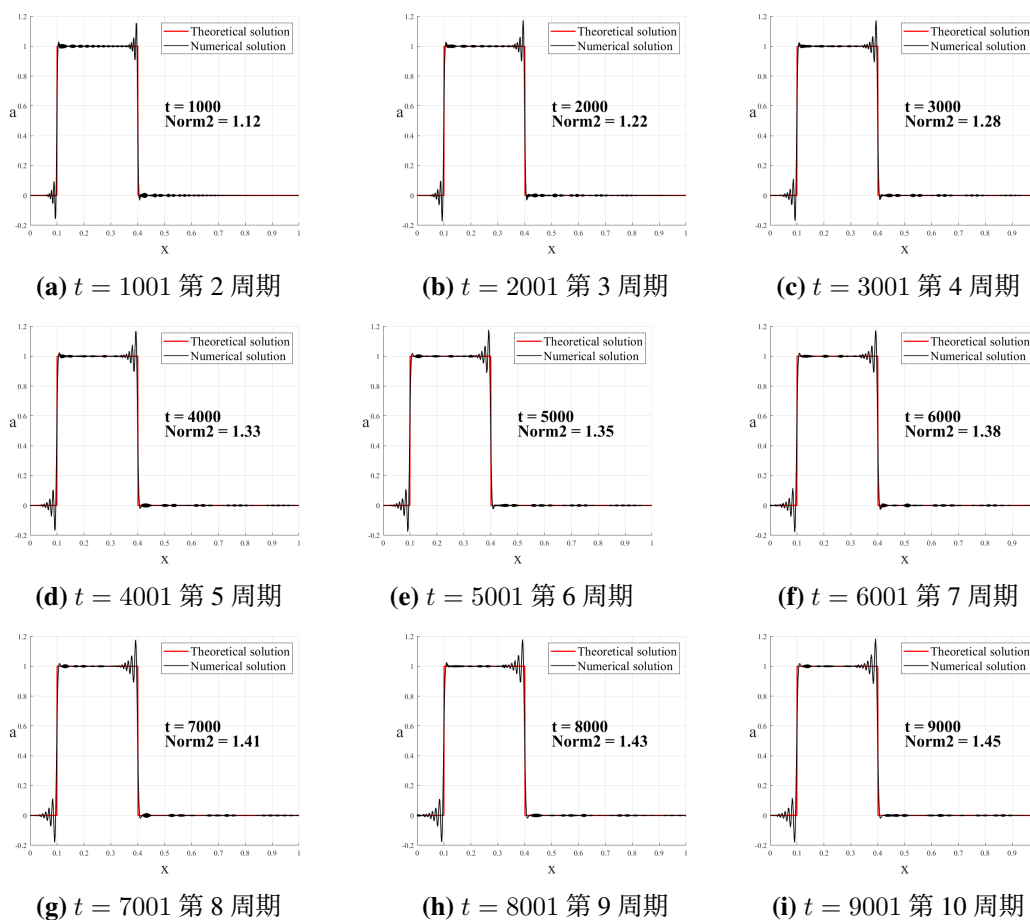


図 2.4: ルンゲ=クッタ法を適用した数値シミュレーション (1000 ステップ出力)

さらに、ルンゲ=クッタ法を用いて、一周期 1000 時間ステップごとの数値解を出力した結果を図 2.4 に示す。時間が経つにつれて、2-ノルム誤差が増加する傾向はあるものの、9000 ステップにおける Norm2 誤差は前進オイラー法の 5 ステップよりも高い精度であることが確認できた。つまり、ルンゲ=クッタ法を用いれば少なくとも、9000 ステップ目まで数値解の収束性が高く、不安定化しないことが分かった。

2.4.3 後退オイラー法による時間離散

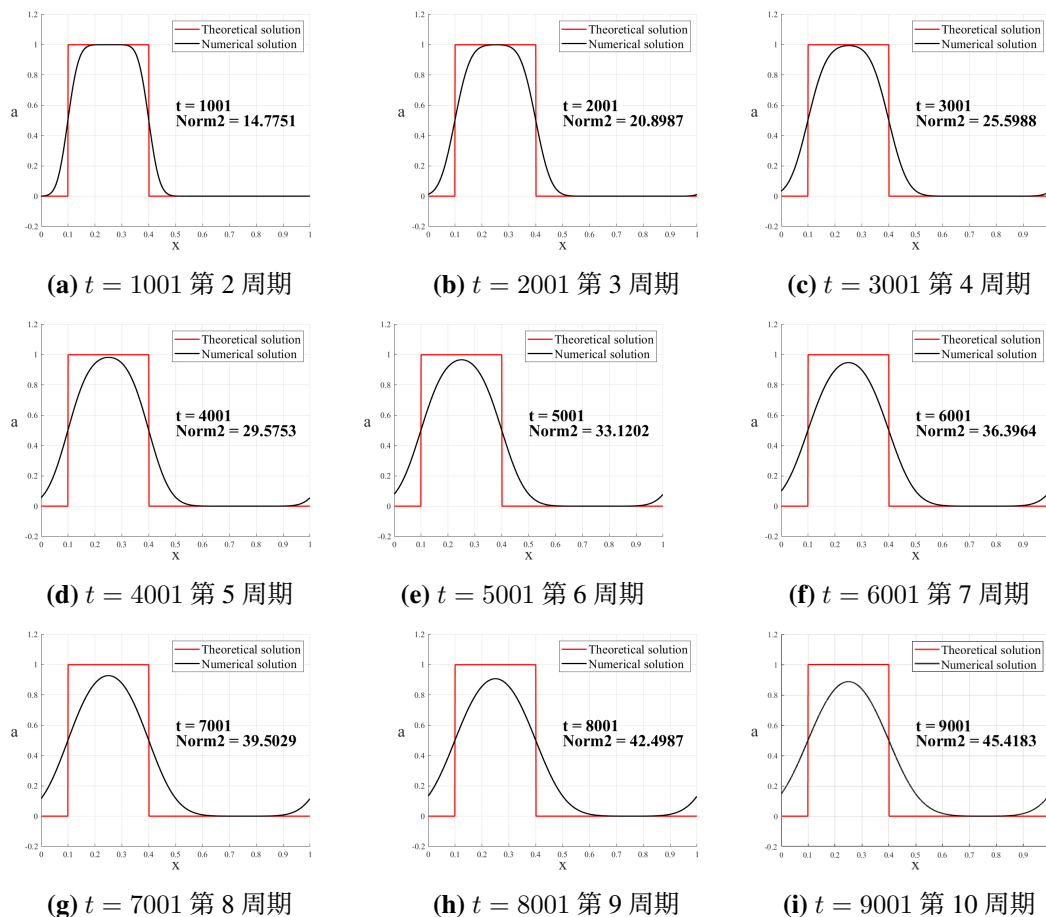


図 2.5: 後退オイラー法を適用した数値シミュレーション ($\Delta t = \Delta x = 0.0001$)

同様、後退オイラー法の場合のは、一周期 1000 時間ステップごとの数値解を図 2.5 に示す。前進オイラー法のような数値振動は発生しないことが確認できる。原因として、後退オイラー法においては、クーラン条件を満たす限り、次の時間ステップの値が必ず現時刻ステップ値の符号と一致するため、数値振動が発生しないことが挙げられる。しかし、数値振動が発生しない代わりに、矩形波の形が時間進行とともになまることも確認された。

どんな時間離散化を用いても、最初の第 1 時間ステップの数値解を計算する段階で既に誤差が生じており、第 1 時間ステップの情報を利用して、後の時間ステップの数値解を算出する限り、誤差の発生は避けられないと考えられる。

2.5 残差分析

陽解法の前進オイラー法、ルンゲ=クッタ法、陰解法である後退オイラー法とどれにしても誤差が発生している。誤差の主な原因は、重み付き残差法的前提が残差の最小化ではなく残差の平均の最小化にあるためだと考えられる。つまり、重み付き残差法は、残差の平均値がゼロと仮定したが、全定義域において、残差が必ずしも最小化されていないということである。全定義域の残差がすべてゼロの時、近似解は理論解と一致する。そのため、本節では、各方法における数値解を残差式に代入し、数値解の修正余地を調査する。

数値解と採用される基底関数 ψ_i を式 (2.5) に代入すると、その数値解の残差が得られる。また、理論解についてもラグランジュ基底関数を採用する場合には、時間微分項に前進差分法を、空間微分項に後退差分法を適用することで、残差がゼロとなることがかくにんできる。残差式が次のようになる。

$$\frac{\partial a_i \psi_i(x)}{\partial t} = \frac{a_{(t+\Delta t, x)} - a_{(t, x)}}{\Delta t} \psi_i(x) \quad (2.19)$$

$$c \frac{\partial a_i \psi_i(x)}{\partial x} = c \frac{a_{(t, x)} - a_{(t, x-\Delta x)}}{\Delta x} \frac{\psi_i(x)}{\partial x} \quad (2.20)$$

ここで、式 (2.5) は

$$\frac{a_{(t+\Delta t, x)} - a_{(t, x)}}{\Delta t} \psi_i(x) + c \frac{a_{(t, x)} - a_{(t, x-\Delta x)}}{\Delta x} \frac{\psi_i(x)}{\partial x} = R \quad (2.21)$$

のように整理できる。式 (2.21) に数値解を代入することで、各時刻各位置での残差が計算可能となる。

また、*Residual* はその時刻に領域内全ての残差絶対値の合計を表す。*Residual* の値が高ければ、修正余地が大きいことを表す。

$$Residual = \sum_{x=1}^{1/\Delta x} |R_x| \quad (2.22)$$

理論解を式 (2.21) に代入し、 $0 \leq t \leq 2 \times 10^4$ における理論解の残差を図 2.6 に示す。図 2.6 より理論解の残差はゼロであることがわかる。残差と数値解の関係を考察するために、前進オイラー法は明らかに精度が低いため、ここでは省略する。後退オイラー法とルンゲ=クッタ法を式 (2.21) に代入し、残差を求める。図 2.7 図 2.8 図 2.9 に示すように残差があることを確認できた。

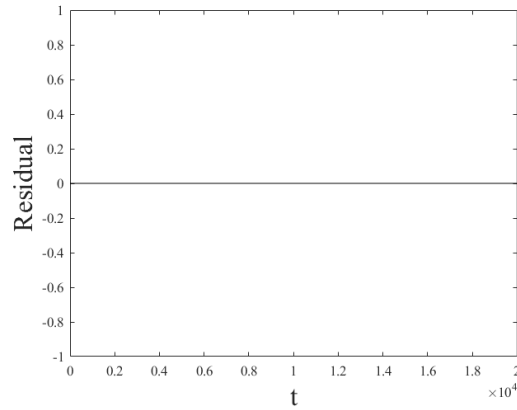


図 2.6: $0 \leq t \leq 2e4$ の時における理論解の残差値状況

$t = 1$ 時の残差と数値解を図 2.7 に示す. オイラー法とルンゲ=クッタ法は共に残差が不連続面の周辺に集中する. 式 (2.21) により, $t = 1$ の時はどの計算方法でもまだ発散していない. 初期値と一時間ステップ後の数値解の差が不連続面の周辺のみ存在するからである. 後退オイラー法の合計残差がルンゲ=クッタの合計残差値より高いことが確認できた.

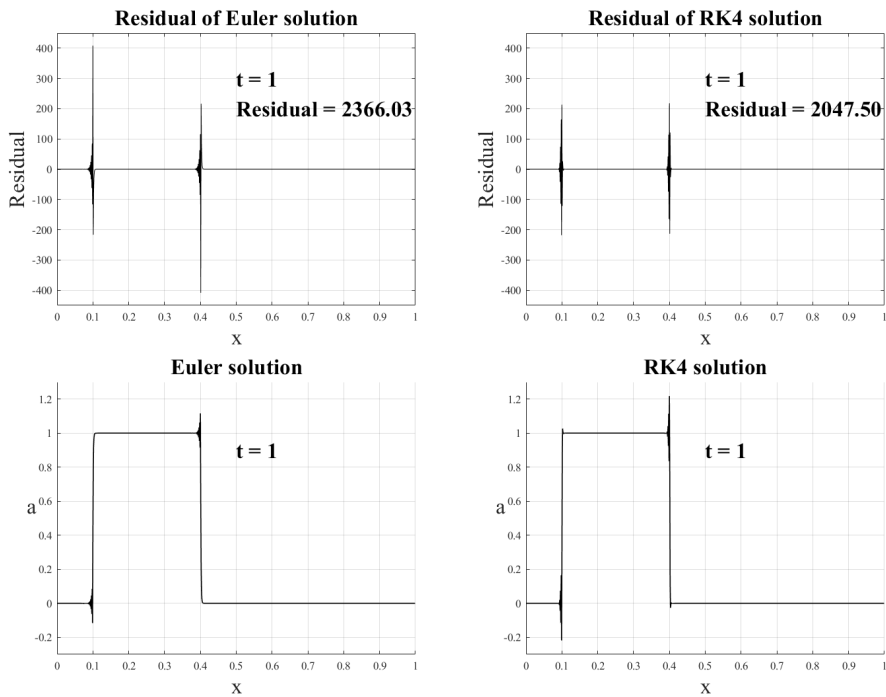


図 2.7: $t = 001(STEP : 1)$ の時における残差および数値解

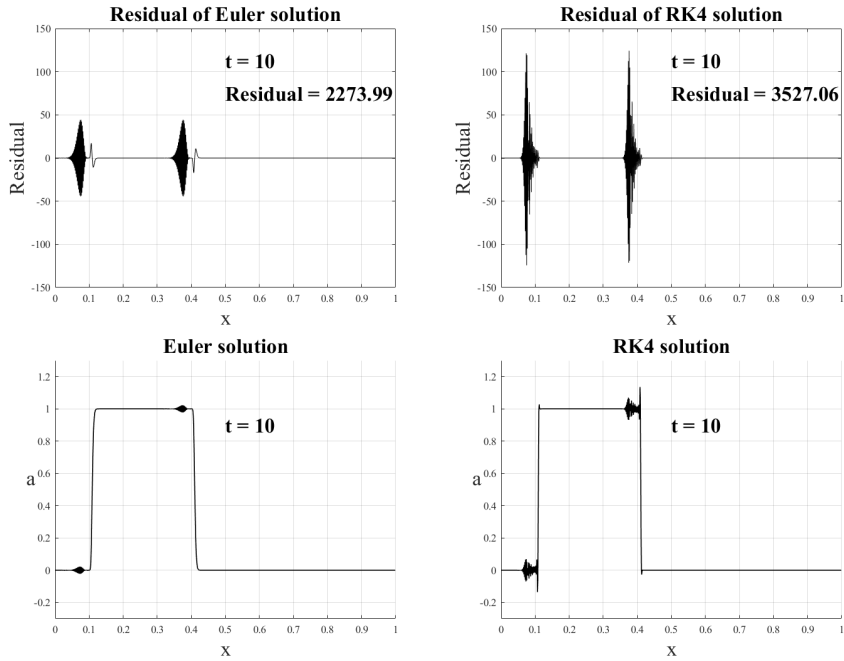


図 2.8: $t = 010$ (STEP : 10) の時における残差および数値解

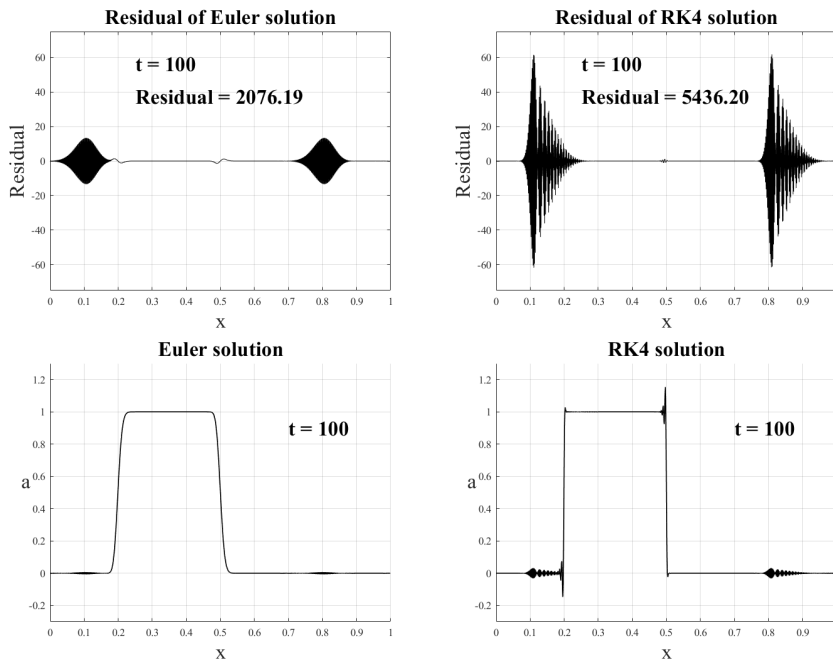


図 2.9: $t = 100$ (STEP : 100) の時における残差および数値解

$t = 10$ ($STEP : 10$) 時の残差と数値解状況を図 2.8 に示す。オイラー法数値解の残差と $t = 1$ の時の残差と比較すると、残差の極大値がかなり下がり、合計残差の減少、不連続面の周辺残差が激しく振動する。その原因はオイラー法を採用したことで、安定性と収束性が低く、時間ステップの進行とともに、徐々に発散し、矩形波の不連続性が失われ、次の時間ステップとの差が小さくなるからである。また、後退オイラー数値解が不連続面周辺において滑らかになり、式 (2.21) の空間微分項も小さくなる。以上により、時間微分項と空間微分項両方とも値が小さくなることにより、合計残差値が下がる。

ルンゲ=クッタ数値解の場合では、 $t = 1$ の時の残差と比較すると、残差の極値は減少したが、残差合計が増加した。原因は、多段階な計算プロセスが矩形波の不連続情報を次の時間ステップにより多く伝播してしまい、不連続関数に近似解する際に数値振動が起こると考えられる。

図 2.9 に $t = 100$ 時の残差数値解状況を示す。後退オイラー数値解残差の極値がさらに、下がり、数値解に数値振動がある程度収まってきたことがわかった。合計残差値がまだ下がった。一方、ルンゲ=クッタ法の場合では、残差の極値が抑えていても、合計残差値が増加したことが確認できた。

2.6 まとめ

本章ではガラーキン法において、矩形波の一次元移流の数値シミュレーションを比較した。また、数値シミュレーションの結果を残差式に代入することにより、各数値解の残差を計算した。その残差について比較分析を行ったところ、理論解を残差式に代入すると、全時刻、全領域において残差値がゼロ。ガラーキン法の数値解を代入すると、残差が生じることを確認した。

また、残差を比較する際に、オイラー法とルンゲ=クッタ法は共に時間経過に伴い、残差の振動やピークが変わっていくことを確認した。

そのため、数値解の残差を目的関数として、数値シミュレーションにより得られた数値解を最適化することで数値振動が抑えられる可能性がある。数値解更新方法には遺伝的アルゴリズム (GA) が考えられる、次節より重み付き残差法により求めた数値解を最適化することにより残差を最小化できる可能性について検討する。

第3章 遺伝的アルゴリズムによる数値解の修正

3.1 遺伝的アルゴリズムの概要

重み付き残差法を適用し、ある区域内の重み付き残差の平均値をゼロとする数値解が必ずしも定義域内の残差を最小化しているわけではないことをかくにんした。全ての残差がゼロであれば、近似解は理論解に一致するため、残差式を目的関数として最適化を行うことで、より高精度な数値解が得られる可能性がある。ある時刻における近似解全体を一つの個体と考えて、遺伝的アルゴリズムを適用して、解の修正を試みる。

遺伝的アルゴリズム (GA: Genetic Algorithms) は、生物進化 (選択淘汰, 突然変異) の原理に着想を得たアルゴリズムである、確率的探索手法の一つである [5]。JH Holland (1975) [6] において初めて導入された。基本的な構成は初期集団の生成と終了するまでの最適化ループである。最適化ループは生成選択, 交叉, 突然変異の三つの主要操作がある。

選択 (Selection)

まず、全ての遺伝子 (ここでは数値解の修正ベクトルの要素) が乱数の個体をいくつか生成する。なお各修正ベクトルは個体また染色体という。個体に含まれる要素は遺伝子という。最初に生成した個体をまとめて初期集団という。初期集団に含まれるすべての個体の適応度 (fitness) は評価関数に遺伝子を代入して求める。この評価値に基づいて、各個体の適応度を判断する遺伝的アルゴリズムでは、適応度の高い個体の遺伝子を優先的に残すため、いくつかの個体選択法が開発されてきた。また、適応度にスケール操作をすることにより、個体の特徴に合わせた選択も可能となる。スケール法には以下の式に示すように“線形スケールリング”, “シグマ切断スケールリング”, “べき乗スケールリング”といったスケールリング手法が存在する。

線形スケールリング:

$$f' = af + b \quad (3.1)$$

シグマ切断スケールリング:

$$f' = f - (\bar{f} - c\sigma) \quad (3.1)$$

べき乗スケールリング:

$$f' = f^k \quad (3.1)$$

ここで、 f は適応度、 f' はスケーリング後の適応度を表す。 a, b, c, k はそれぞれ、ハイパーパラメータである。 σ は当世代における個体集団の標準偏差を表す。

ルーレットホイール選択 (Roulette Wheel Selection)

ルーレット選択では、ある個体が次世代に選択される確率は、その個体の適応度に比例する。 個体選択時にルーレットのように、今世代の個体数のポケットを用意する。 適応度が高いほど選択を確率が高くなるように、優秀な個体のポケットを大きく設定する。 個体 i が選択される確率を式 (3.1) に表す。

$$P_i = \frac{f_i}{\sum_{j=1}^n f_j} \quad (3.1)$$

ここで f_i は i の適応度、 n は、集団内の個体数を表す。 直感的な選択方法であるため、汎用性が高い。 ただし、本方法のデメリットは、適応度の高い個体が複数回選択されることがある。 また、負の適応度を式 (3.1) に代入すると確率も負になるため、負の適応度が適用できないことである。

ランキング選択 (Rank Selection)

ランキング選択とは、適応度値の大きさにより個体を順番付ける。 各順番位置の選択率をあらかじめ決めて、その確率により、次世代に情報を伝播する手法である。

表 3.1: ランキング選択の簡単例 (6 個体)

ランキング	適応度	選択率
1	6.3	0.3
2	5.0	0.25
3	3.4	0.2
4	1.7	0.15
5	0.6	0.07
6	-0.4	0.03

表 3.1 に、ランキング選択の適応度および選択率の関係を示した。 個体の選択率が適応度に直接関係付けられるわけではなく、ランキングに依存する。 ゆえに、負の適応度でも機能させやすい。 また、他方法では、選択を実行の後半段階に、各個体の選択率はほぼ均等になる。 そのため、選ばれる確率はほぼ同じになる。 これにより、より適合した個体への選択圧力が失われ、GA は個体の選択に失敗することになる。 その場合では、このように集団内の個体の適応値が非常に近い場合にランキング選択がよく使用される。

トーナメント選択 (Tournament selection)

トーナメント選択とは、競争的に個体を選択する手法である。まず、トーナメントのサイズを決める。集団にトーナメントの大きさに応じて個体をランダムで取り出す。取り出された個体の適応度を比較し、適応度が最も高い個体を次世代に残す。そのため、トーナメントサイズの変更により、個体の淘汰圧を調節できる。トーナメントのサイズを大きく設定すると、次世代の個体はより適応度がたかくなるが多様性は失われる。トーナメントサイズを小さく設定すると、次世代多様性が高く保たれる。過大なトーナメントを設定すると、局所最適解に陥る可能性がある。

交叉 (Crossover)

交叉は生物の交配遺伝を模倣した情報交換システムである。遺伝的アルゴリズムが求める最適化の精度や探索効率は交叉のアルゴリズムに強く依存しているため、遺伝的アルゴリズムの非常に重要な部分である。染色体の交叉操作とは、前述のメソッドで選ばれた2つの親を交叉させ、何らかの方法で染色体を交換して2つの次世代個体を形成する操作である。主に三つの方法がある。

一点交叉 (One-point crossover)

一点交叉とは今世代個体に一つの交叉ポイントを任意に設定し、次世代個体が今世代個体から交叉的に、情報を伝承する交叉手法である。図3.1のように、今世代個体の交叉ポイントにおける遺伝子情報を交換して次世代個体を作る。

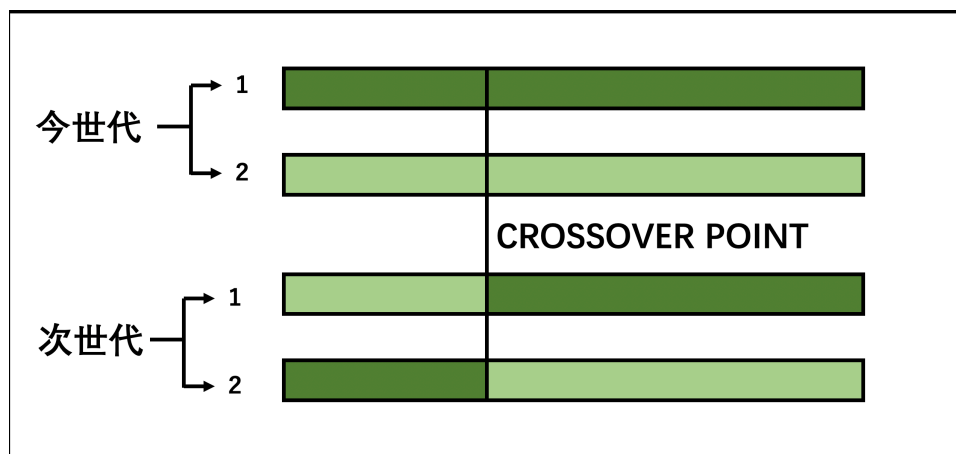


図 3.1: 一点交叉

二点交叉 (Two-Point Crossover)

二点交叉は一点交叉と同様に、今世代個体から交叉的に、次世代個体へ遺伝子情報を伝播する交叉手法である。ただし、一つの交叉ポイントではなく、もう一つの交叉ポイントを設定する。図(3.2)に、二点交叉を示す。また、同様に、交叉ポイントを増やすことで多点交叉も可能である。

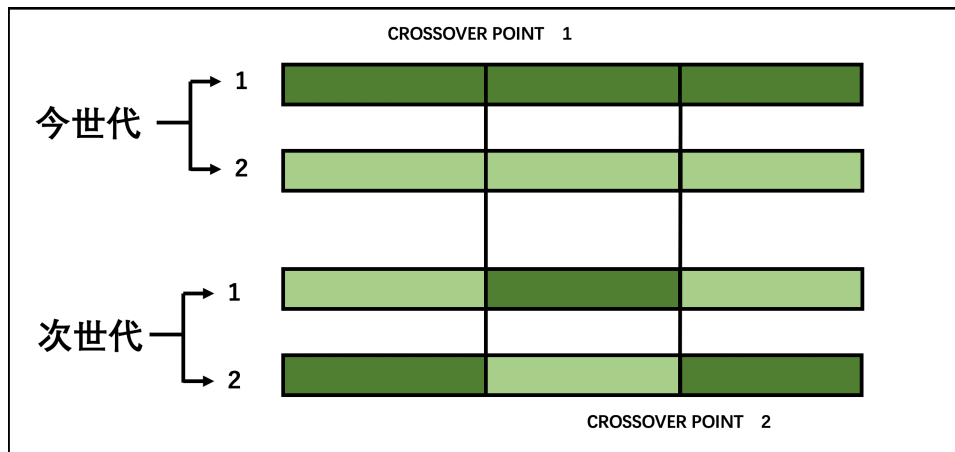


図 3.2: 二点交叉

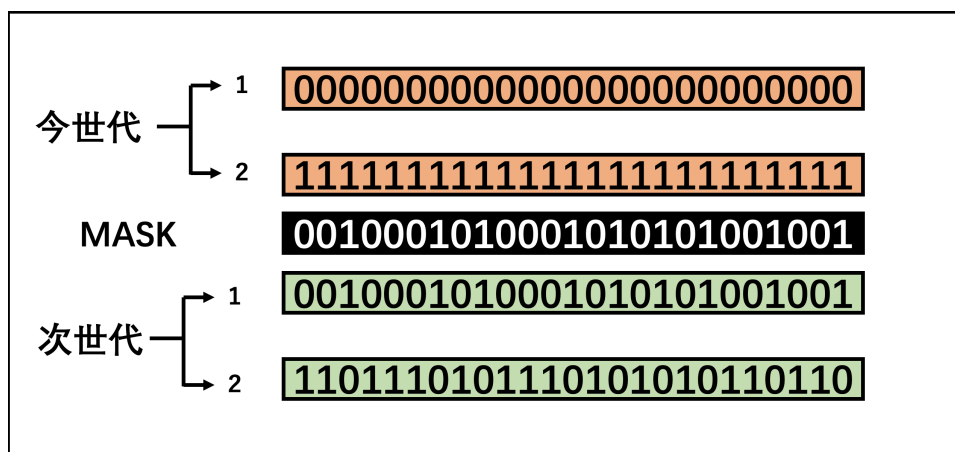


図 3.3: 一様交叉

一様交叉 (Uniform Crossover)

一様交叉は一点交叉や二点交叉をかくちょうしたもので、交叉ポイントを設定するメソッドと異なり、マスクを生成し、そのマスクにより、どちらの今世代個体から遺伝子を引き継ぐかを定める手法である。図 3.3 のように、一番目の次世代個体がマスクの値が 0 の場合では、一番目の今世代個体から遺伝子を継いで、マスク値が 1 の場合は、二番目の今世代個体から遺伝子を継ぐ。二番目の次世代個体は逆に、マスク値が 1 の場合一番目の今世代個体から遺伝子を継ぐ。

以上三つの手法の優劣性は WM.Spear(1991)[7] で発表された。以上の三種類メソッド以外では様々な手法が提案されているが、本研究では、主に一様交叉を採用した。

突然変異

突然変異とは、解の空間を広げる操作である。選択、交叉の操作だけを繰り返すことにより、集団内の最適解を見つけることができる。ただし、集団中の最適解は局部最適解と考えられるため、精度は不十分である。そのため、できるだけ、広い解の空間を探索し、より良いグローバルな、最適解を見つけることが望ましい。変異率による突然変異や適応変異などの変異手法がある。

3.2 本研究への適用

修正最適化のプロセス

数値解の修正が 1 時間ステップごとに行うため、一時間ステップ前の数値解情報も使用される。初期条件を 0 時間ステップの情報として、次々とその後の時間ステップの修正を行う。また、終了条件を満たすとこの時間ステップの最適化が終わる。そして、次の時間ステップの最適化に使う。

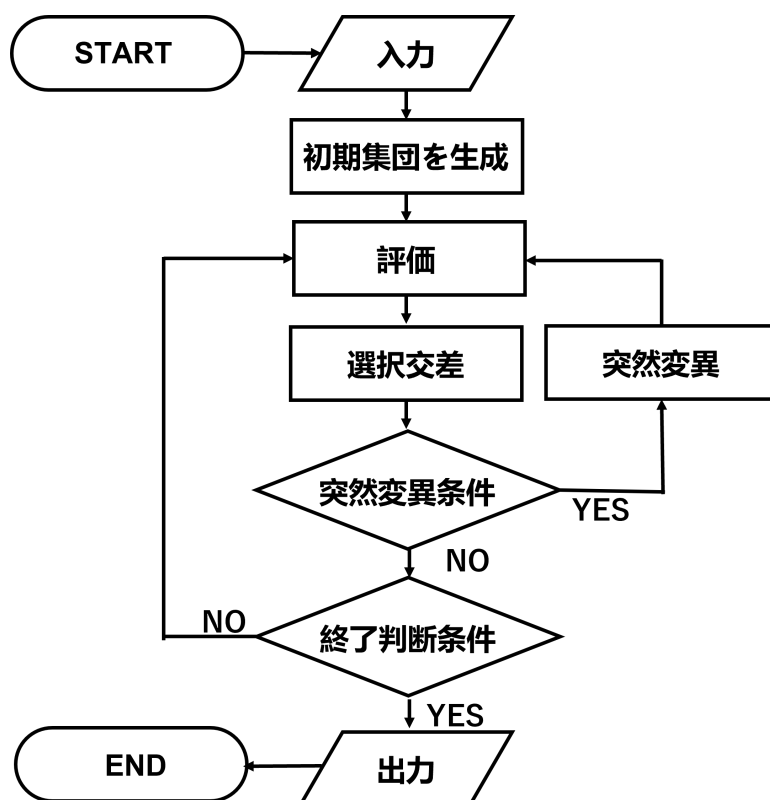


図 3.4: 1 時刻ステップにおける最適化のフローチャート

入力

最初に、第二章の計算方法を用い、設定した計算条件における数値解の算出を行う。また、前の時間ステップの情報も使用されるため、全ての時間ステップにおける数値解の結果をマトリックスに保存することになる。各時刻における数値解を一行として、マトリックスに保存する。マトリックスのサイズは時間ステップの数と要素数による。そのマトリックスを入力する。

初期集団の生成

1時間ステップの数値解と同様なサイズの修正ベクトルを設定した個体数により生成する。その修正ベクトルは個体また染色体という。個体に含まれる要素は遺伝子という。初めて生成した個体の集合は初期集団という。すべての遺伝子は乱数であるため、最適化するためのプログラムの計算負荷が高い。その解決策は、数値解の残差値を利用し、遺伝子の乱数生成範囲の重み関数とする。最適化する必要がない要素の残差がゼロに近い極めて小さい数値である。最適化する必要を判断する閾値を設定して、閾値より小さい残差であれば、遺伝子がゼロになる。この操作で、最適化が不必要な部分の計算資源を節約ができる。

評価

1時間ステップにおける数値解の各値と個体の各遺伝子を加算し、残差式に代入すると、この個体による修正後の残差ベクトルが得る。残差ベクトルの各要素を目的関数に代入すると、この個体の適応度が計算できる。

目的関数について、第二章に紹介したように、 $\Delta t = \Delta x$ の場合には、クーラン条件を満たす。かつ、理論解を残差式に代入して残差が常にゼロである。そのため、数値解を修正してから、結果を残差式に代入し、得られた残差（式2.5）の2-ノルムをとり、目的関数とする。理論解の残差がゼロであることから、目的関数を最小化することにより、数値解の最適化を行う。

$$O = \|\mathbf{R}\|_2 = \sum_{i=1}^n \sqrt{R_i^2} = \sqrt{R_1^2 + R_2^2 + R_3^2 + \dots + R_n^2} \quad (3.1)$$

しかし、遺伝子がランダムに生成したので、適応度が高い個体にも劣等遺伝子が存在する。適応度が低い個体に著しく優秀な遺伝子が存在する可能性もある。そのため、すべての個体を m 要素ごとに分割し、分割された部分の適応度も評価する。さらに、分割された m 個の遺伝子がすべてゼロの場合では、計算負担も減らせる。この後の選択にも役に立つ。また、各個体は他個体と独立であるため、適応度を計算する際に、並列計算を行うと効率的によい。最適化を行う時に、最も計算負荷が高いのは評価の部分である。

$$O = \sum_{j=1}^{L/m} O_j$$

$$O_j = \sum_{k=mj+1}^{m(j+1)} \sqrt{R_k^2} = \sqrt{R_{mj+1}^2} + \sqrt{R_{mj+2}^2} + \sqrt{R_{mj+3}^2} + \dots + \sqrt{R_{m(j+1)}^2}$$

選択交叉

今回、アルゴリズムではルーレットホイール選択を採用する。各個体の適応度の二乗をとる。“べき乗スケールリング”の“二乗スケールリング”が採用された。そのことにより、良い個体を選択される確率を拡大し、良くない個体を選択される確率を減少する。また、個体を分割して、個体の各部分の適応度を利用することにより、個体を丸ごとを選択するのではなく、個体の各部分の適応度により選択を行うになる。

交叉操作については今回の最適化においては個体にある遺伝子の数が多いため、一点交叉や二点交叉を採用せずに、一様交叉を採用される。

突然変異

突然変異を行う時に、事前に突然変異量率を設定する必要がある。それは突然変異を行う時に、新しく生成した個体が集団に占める割合を示すパラメータである。また、前節で選択交叉した個体を改めて評価を行う、その中に適応度が低い個体を削除し、突然変異による生成した個体に入れ替える。

突然変異を発生する頻度が突然変異率による、本研究では突然変異率の設定が二種類がある。一つ目は定変異率、定変異率とは、設定した突然変異率により、突然変異を行う。しかし、最適化を行う途中に局部最適解に陥る場合では、突然変異が期待されるが、なかなか突然変異を発生しない現象をよく観察された。そのため、定率の代わりに、二つ目も適応変異率手法も採用した。適応変異率手法とは、過去世代に最も良い個体の適応度を保存し、近隣世代の適応度の標準偏差を取る。標準偏差が小さければ突然変異率が高くなるという突然変異手法である。この手法の特徴は局部最適解に陥ると突然変異率が上がり、変異の発生に促進する。しかし、過去のデータ保存する必要があるため、メモリーと計算速度に影響を及ぼす。

終了判断条件

本研究の最適化を行う際に、モンテカルロ法を利用し、乱数による数値的な最適化であるため、計算負荷が高い。そのため、終了判断条件としては2パターン設定した。一つ目は最適化後の *Residual* 値と数値解の *Residual* 値の比例による判断する。二つ目は最適化ループの最大回数である。

第4章 結果と考察

第二章最後に、数値解誤差の原因は重み付き残差法による残差値の最小化を行えずという仮説を提出した。前進オイラー法を採用して、数値シミュレーションを行う場合には、数値振動がかなり激しいので、修正するため、計算負荷が著しく高い。本研究の修正アルゴリズムに対して、コストパフォーマンス的には良くない。また、ルンゲ=クッタ法が陽解法的一种であるため、空間並列性も持ち、分散コンピューティングに適用する可能性が高いと考えられる。そのため、本章では残差値に基づく目的関数を用いて、ルンゲ=クッタ法を用いた数値解の修正を行うことにより、仮説を検証する。

表 4.1: 最適化数値解の計算条件

境界条件	周期境界条件
基底関数 ψ	ラグランジュ型
移流速度 c	1.00
計算領域 L	$0 \leq x \leq 1$
空間刻み幅 Δx	0.005
時間刻み幅 Δt	0.005

以上の計算条件を用いた数値解の第一時刻ステップにおける結果図 4.1 で表す。

図 4.1 によると、第一時刻ステップの数値解に既に数値振動が発生することを確認できた。

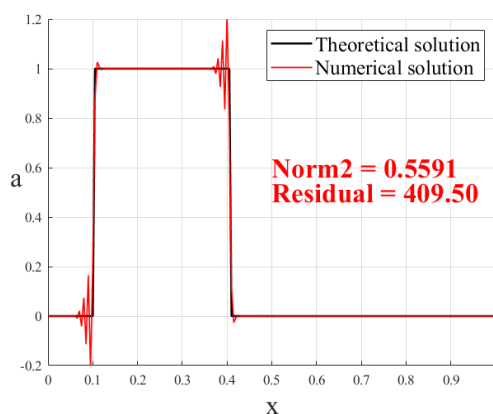


図 4.1: 修正される数値解と理論解の比較

また、理論解と比較による $Norm2$ 誤差値が 0.5591 である。また、この時刻に、全ての残差の絶対値の合計が 409.5 である。

第三章第二節の評価部分に個体を分割する評価方法に沿って、40 分割、100 分割をそれぞれ $1e6$ 世代で行う。

4.1 40 分割における各世代目の修正結果

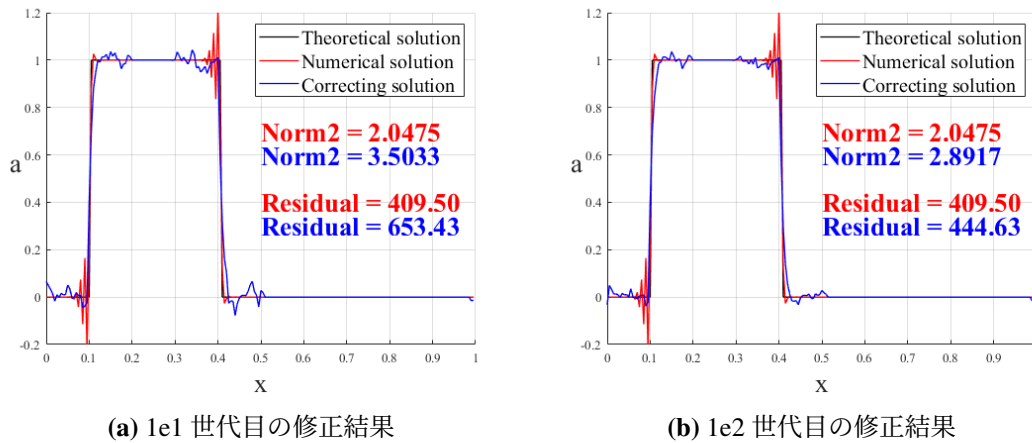


図 4.2: 40 分割修正における 1e1,1e2 世代目の修正結果

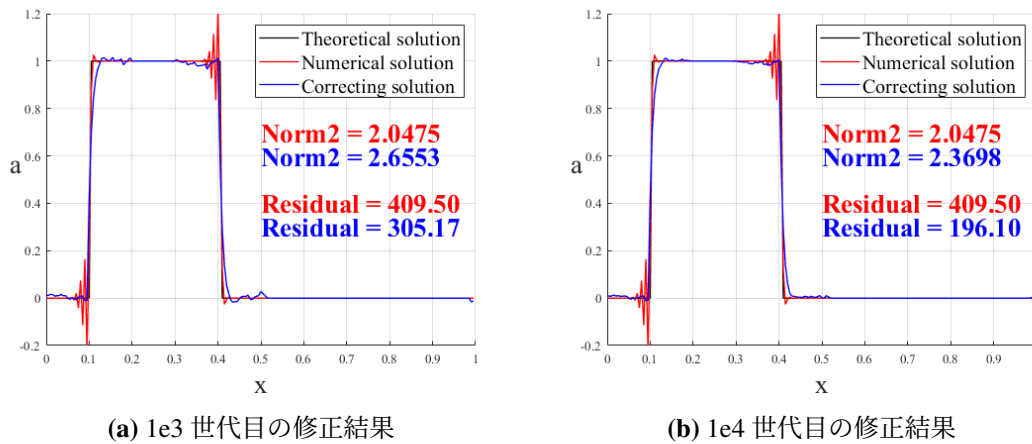


図 4.3: 40 分割修正における 1e3,1e4 世代目の修正結果

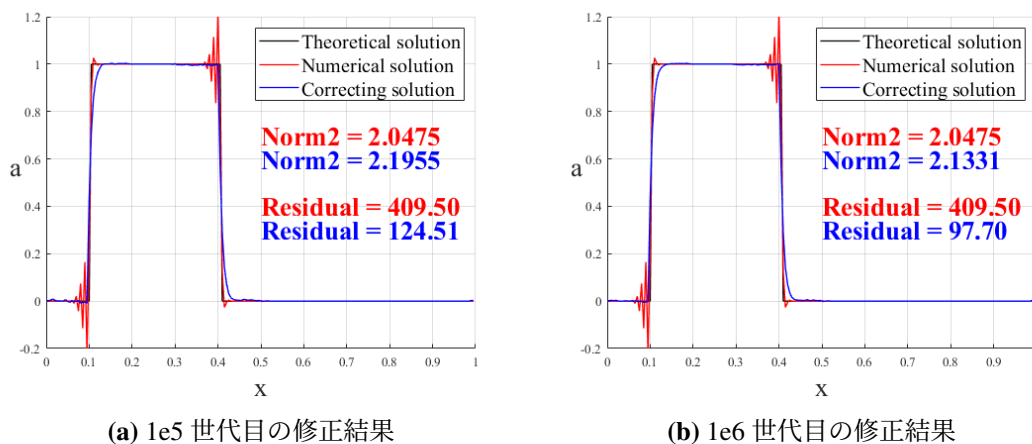


図 4.4: 40 分割修正における 1e5,1e6 世代目の修正結果

4.2 100 分割における各世代目の修正結果

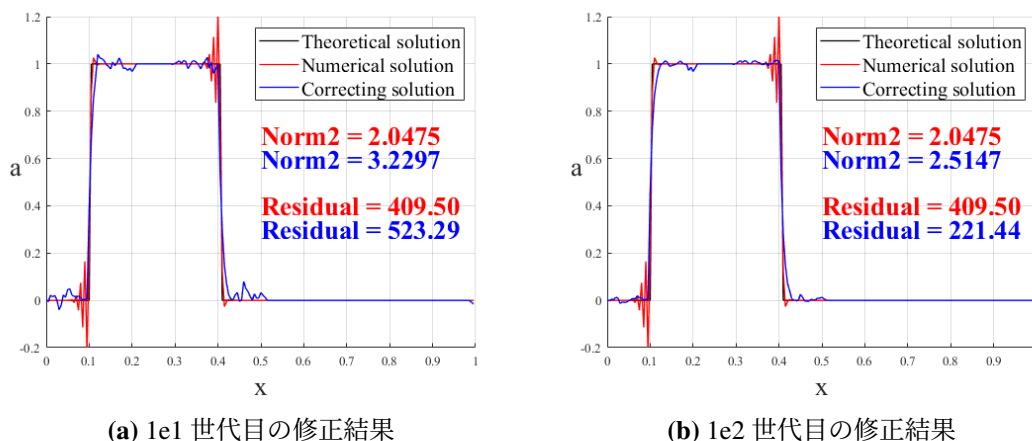


図 4.5: 100 分割修正における 1e1,1e2 世代目の修正結果

4.3 修正結果に関する考察

40 分割と 100 分割の修正結果を比較すると、100 分割された時に、修正アルゴリズムの効率性のほうが高いが、分割された個体の遺伝子数が少ないため、速く修正できる。また、遺伝子数が少ないことにより極端な遺伝子による修正影響が抑えることも原因の一つと考えられる。

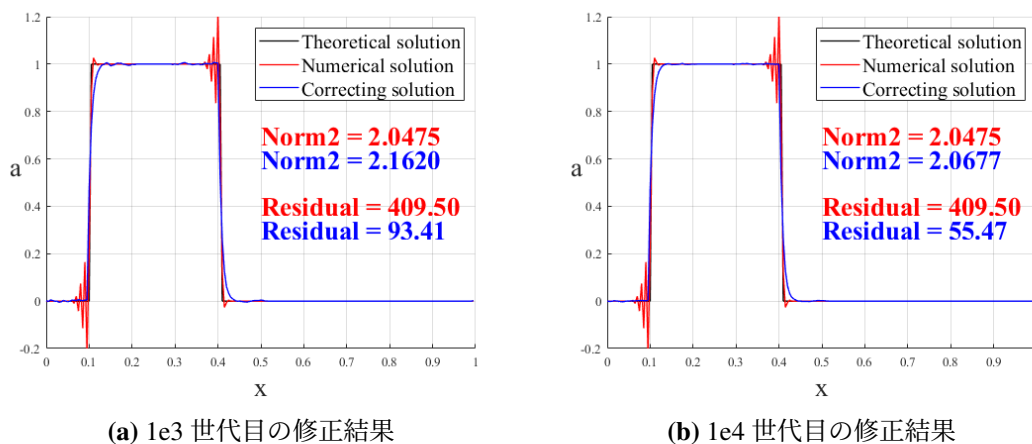


図 4.6: 100 分割修正における 1e3,1e4 世代目の修正結果

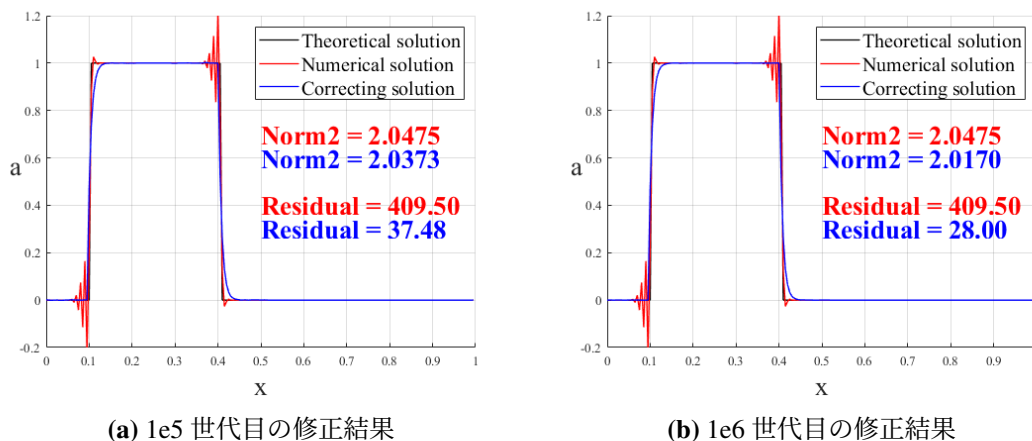


図 4.7: 100 分割修正における 1e5,1e6 世代目の修正結果

図 4.7 により, 1e5,1e6 世代における 100 分割された数値解の修正を行った. その時の修正解の 2 - ノルム誤差が重み付き残差法の数値解より小さいことを確認できた. また, 図 4.8 と図 4.9 により, 40 分割と 100 分割において, 残差値と 2 - ノルム誤差とある比例関係を持つと考えられる.

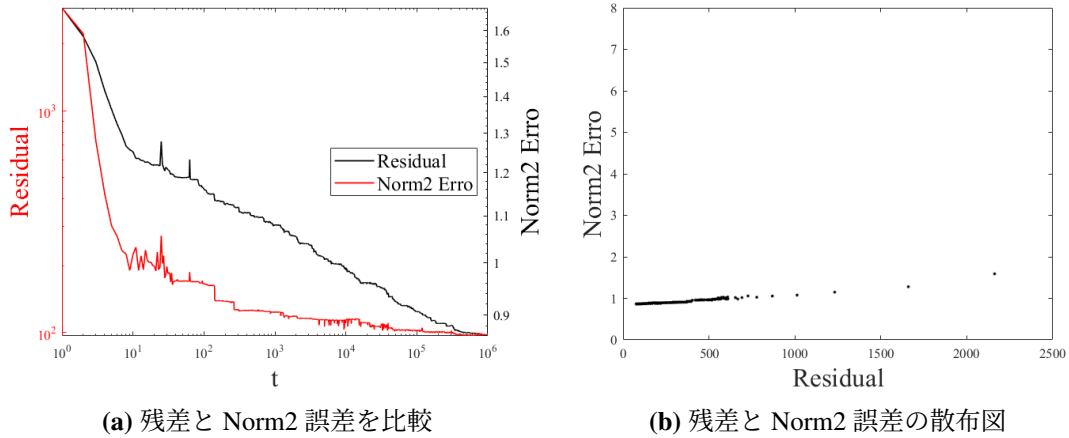


図 4.8: 40 分割における残差と Norm2 誤差の関係

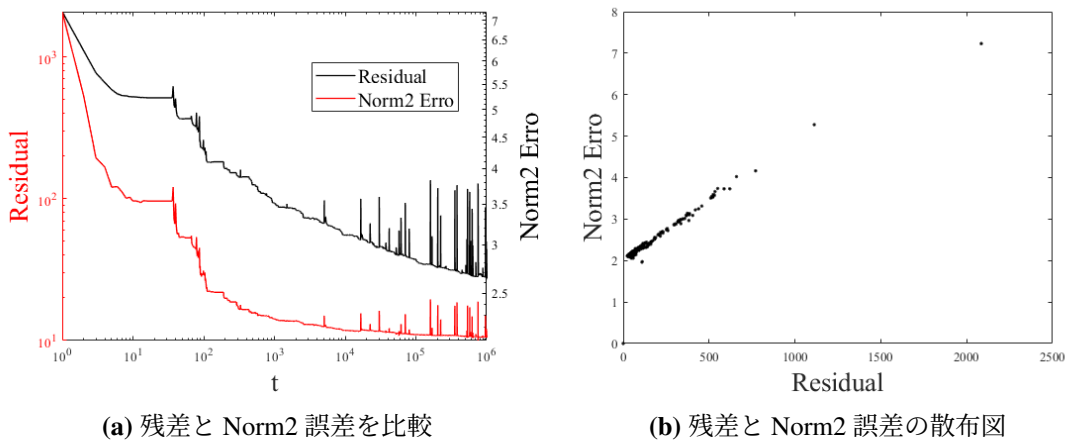


図 4.9: 100 分割における残差と Norm2 誤差の関係

第5章 まとめ

遺伝的アルゴリズムにより、有限要素法の数値解を残差に基づき修正する方法を検討した。

5.1 確認した結果と得られた知見

- 1) 重み付き残差法は、定義域の残差を最小化せず、数値解には誤差が残ることを確かめた。
- 2) 分散コンピューティングへの適用性を考慮して、前進オイラー法やルンゲ=クッタ法の精度を比較した。ルンゲ=クッタ法は並列計算への適用性を備える上に、前進オイラー法より高い精度を持つ。
- 3) 遺伝的アルゴリズムにより、重み付き残差法数値解の残差に基づく修正が可能であることを明らかにした。
- 4) 遺伝的アルゴリズム内部では、個体を細かく分割すると修正速度が速くなる。
- 5) 修正結果によると、数値解の2-ノルム誤差と残差には比例関係がある。

5.2 今後の予定

- 1) 本研究では40分割と100分割を行った。分割数が変わることで、2-ノルム誤差と残差の比例関係が変化する。他の分割数でも同様に検討を行い、その関係を明らかにする必要がある。
- 2) 200要素の数値解修正を行ったが、要素数を増やすことにより、メモリーの使用やハードディスクの負担などの原因により、計算負荷が増加する。計算負荷の小さな更新アルゴリズムを開発することで、より細かい要素の数値修正が期待できる。
- 3) 遺伝的アルゴリズムは乱数による最適化アルゴリズムであるため、今回のような連続性を持つ問題に対する限界がある。そのため、もう一つの遺伝的アルゴリズムである差分進化法 (Differential evolution) の修正検討を行う。

謝辞

本研究の課題設定から、論文執筆まで、また就職活動におけるアドバイスまで非常に親切にご指導を下さいました山本亨輔先生に心から感謝を申し上げます。二年間、ゼミで丁寧にご指導をいただきました松島亘志先生、非常に重要なアドバイスを頂きました ARANHA CLAUS DE CASTRO 先生に深く感謝いたします。

大学院入学試験から、修論を書くまで、常に優しく励ましてくれた Jiang Xiaoyu さんにも感謝します。留学生活における悩みに丁寧に乘ってくれた Jiang Haoran さん、同級生の橘幸宏さん、Xu Dan さんにも感謝します。

最後に、日本への留学を支えてくれた家族に心から御礼を申し上げます。

付録A 選点法と最小二乗法の概要

選点法

選点法とはデルタ関数を重み関数として選ぶ方法である $\phi_i = \delta(x - x_i)$. デルタ関数は引数が0の時に無限大, 他の場合は常に0という性質を備える.

また, 全領域において, 積分すると1となる. そのため,

$$\int_{\Omega} R\delta(x - x_i)d\Omega = R|_{x=x_i} = 0 \quad (\text{A.0})$$

を得ることができ, 係数 a_i 算出ができる.

最小二乗法

最小二乗法では, 定義領域 Ω 内重み関数は残差 R がある未知関数 α_i の偏微分と定義される.

$$\phi_i = \frac{\partial R}{\partial \alpha_i} \quad (\text{A.0})$$

残差 R の二乗と α_i の関係は

$$I(\alpha_i) = \int_{\Omega} [R(\alpha_i, x)]^2 d\Omega \quad (\text{A.0})$$

である. I を α_i について偏微分すると.

$$\frac{\partial I}{\partial \alpha_i} = 2 \int_{\Omega} R(\alpha_i, x) \frac{\partial R(\alpha_i, x)}{\partial \alpha_i} d\Omega = 0 \quad (\text{A.0})$$

を得る. さらに,

$$\int_{\Omega} R(\alpha_i, x) \frac{\partial R(\alpha_i, x)}{\partial \alpha_i} d\Omega = 0 \quad (\text{A.0})$$

である. 連立方程式を計算すると, α_i が算出できる.

参考文献

- [1] Alexander Hrennikoff. “Solution of problems of elasticity by the framework method”. In: (1941).
- [2] Richard Courant. “Variational methods for the solution of problems of equilibrium and vibrations”. In: *Bull. Amer. Math. Soc.* 49 (1943), 1-23 (1943).
- [3] M Jon Turner et al. “Stiffness and deflection analysis of complex structures”. In: *journal of the Aeronautical Sciences* 23.9 (1956), pp. 805–823.
- [4] Ray W Clough. “The finite element method in plane stress analysis”. In: *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960.* 1960.
- [5] 北野宏明. “遺伝的アルゴリズム”. In: *人工知能* 7.1 (1992), pp. 26–37.
- [6] JH Holland. “Adaptation in Natural and Artificial System”. In: (1975).
- [7] William M Spears and Kenneth A De Jong. “An analysis of multi-point crossover”. In: *Foundations of genetic algorithms.* Vol. 1. Elsevier, 1991, pp. 301–315.