

【2011 年度学士卒業論文】
有限要素法による熱の移流拡散問題の解析
- 安定化有限要素法の適用と定常問題の解析 -

大阪工業大学 工学部 環境工学科 知能設計工学研究室
F08-037 妹尾 賢
Department of Environmental engineering, Faculty of Engineering
Osaka Institute of Technology
Ken SENOO

2012 年 3 月 31 日

目次

第 1 章	緒言	3
1.1	はじめに	3
1.2	ヒートパイプについて	3
1.3	研究の手段・方法	3
1.4	研究の意義	3
1.5	論文の構成	4
1.6	伝熱現象について	4
第 2 章	熱の移流拡散問題の定式化	5
2.1	移流拡散現象の概要	5
2.2	拡散方程式 (Diffusion equation) の導出	5
2.3	移流方程式 (Advection equation) の導出	7
2.4	移流拡散方程式 (Advection Diffusion equation) の導出	9
2.5	本研究で扱う問題	11
第 3 章	有限要素法の適用	12
3.1	有限要素法とは	12
3.2	二次元定常移流拡散方程式への有限要素法の適用	12
3.2.1	場の方程式	12
3.2.2	重み付き残差法 (Weighted Residual Method, WRM) の適用	12
3.2.3	弱形式 (weak form) の導出	14
3.2.4	Galerkin 法の適用	15
3.2.5	有限要素方程式 (finite element equation) の導出—四角形要素の適用—	16
3.2.6	境界条件の組み込み	17
3.2.7	多元連立方程式の解法—ガウスの消去法 (Gauss elimination)—	18
3.3	座標変換・内挿関数の設定・積分の計算	20
3.3.1	局所座標系の設定と要素の次数	20
3.3.2	座標変換とアイソパラメトリック要素	21
3.3.3	ガウスの数値積分	22
第 4 章	安定化有限要素法の適用	23
4.1	解の不安定さと Peclet 数—一次元定常移流拡散問題の厳密解の例—	23
4.2	Galerkin 法と Upwind Galerkin 法	24
4.3	Upwind Petrov-Galerkin 法	28
4.4	SUPG(Streamline Upwind/Petrov-Galerkin) 法	29
4.5	要素サイズの検討	30
4.6	二次元定常移流拡散方程式への SUPG 法の適用	31
第 5 章	三角形要素での解析	34
5.1	三角形要素での近似関数	34
5.2	安定化を施した有限要素式への適用	36
第 6 章	熱の移流拡散方程式の解析	40
6.1	問題の設定	40
6.1.1	場の方程式と有限要素方程式	40
6.1.2	解析対象の物性値	41
6.2	解析システムの構築	42
6.3	プログラムでの解析と結果	42
6.3.1	4 要素での解析	43
6.3.2	一次元定常移流拡散問題での検証	44
6.3.3	二次元定常移流拡散問題での検証	46

第 7 章	結言	54
7.1	結果	54
7.2	今後の課題	54
7.2.1	移流項の計算と非定常問題への拡大の必要性	54
7.2.2	境界積分項の計算での問題点	55
7.2.3	Upwind Galerkin 法での安定化の実装	55
7.2.4	三角形要素と四角形要素の計算について	55
7.2.5	熱伝導率の非等方性の取り扱い	55
7.2.6	その他—三次元問題への拡大、発熱率や流速及び物性値の取り扱い—	56
参考文献		57
謝辞		58
付 録 A	付録	59
A.1	プログラム fem2dh_steady_AD の構造	59
A.2	プログラムのルーチンの説明	59
A.3	解析プログラム fem2dh_steady_AD のソースコードの掲載	60
A.4	入力データ作成支援プログラム melement4 及び melement3 のソースコードの掲載	69
A.5	fem2dh_steady_AD におけるルーチンでの変数の役割	74
A.6	式に登場する記号	77
A.7	用語集	79

第1章 緒言

1.1 はじめに

世の中の現象は数式で表すことが可能なことがある。特に自然現象はその典型である。力や熱、電気などは数式で表現することが可能で、今日まで学問として研究される対象となってきた。これらの現象は相互に作用して複雑になると、現象の解明が困難になる。

環境工学としては、エネルギーの観点から熱がどのように伝わるのか、どのように利用すれば、高効率であるかを解明することは、持続可能な社会を形成す社会を構築・推進する上で重要な問題である。なぜならば、高効率な熱利用により火力発電や自動車など化石燃料の燃焼によって排出される温室効果のある二酸化炭素（CO₂）の低減につながるからである。

このような考えのもと、本研究では熱がどのように伝わるかに着目した。そして、熱がどのように伝わり分布するかを解明する目的で研究を行った。また、熱の伝わり方を利用した高効率な熱利用がされている装置としてヒートパイプがある。将来的にはこの装置における熱の現象の解明を行い、この装置の改良や応用に適用することを念頭に置いている。本論文ではヒートパイプについては取り扱っていないが、次節でヒートパイプの説明およびなぜヒートパイプに注目するのかについて触れる。

1.2 ヒートパイプについて

ヒートパイプと呼ばれる装置が存在する。この装置は高効率な熱輸送装置である。ヒートパイプは構造として、作動流体が密閉された管状の容器となっている。一端が加熱されると、内部の作動流体が蒸発し、周囲から潜熱として熱を奪うことで冷却機能が働く。もう一方で、蒸発した流体が冷やされて凝縮を起こし、潜熱として管壁に熱を放出する。

このようにして、物体の潜熱がはたらき状態の遷移が伴わない伝熱（顕熱輸送）と異なり、伝熱が素早く高効率の熱輸送が可能である。この装置の興味深い点はこの装置では、熱、物質、運動量がかかわる複雑な現象が作用して、装置の機能をなしている点である。これら三つの要因については、それぞれフーリエの法則、フィックの法則、ニュートンの法則という法則が存在しており、式で表現することができる。

このような複雑な現象を表現し現象の解明ができれば、その他の現象についても適用することができ有用である。それゆえに、ヒートパイプは工学の対象として興味深い装置であり、将来の解析を意識している。

1.3 研究の手段・方法

熱の伝わり方を解明する方法としては、測定器具を用いた実験による測定と計算機を用いた数値解析とが考えられる。前者の実験においては個別の事象について、詳細な現象の解明が期待できる。しかし、人が実際に対象に対して測定を行う必要があるため、大きな規模の対象領域に対しては全体の解明が困難であり、また実際に対象が存在し測定できなければ実験できない。

一方で、後者の数値解析は物性値や対象とする領域を見積もり設定すれば、想像しうるどのような対象に対しても解明が可能である。そして、実際に実験を行うよりも少ない労力で済む。しかし、物性値や対象とする領域のデータなどをどのように取得するか、解析の精度の問題が生じる。

これらの二つの解明方法のうち、本研究では後者の数値解析を解明の手段として用いる。その理由として、次のことを考慮した。現在は過去の実験により、物性値が判明している物質が多く、また将来的には数値解析の方が適用範囲が広く有用であると考えた。

数値解析には、有限要素法という方法を用いた。その過程において、コンピューターを用いた計算によって解明を試みる。領域全体の値の分布をみることで現象の解明とする。

1.4 研究の意義

熱の伝わり方自体を知るには市販の解析プログラムを使えばこと足りる。しかし、将来的により複雑な現象を扱い、その解析の途中での値を取り出したり、可視化しやすいうように出力の形式を変更したり、他の解析プログラムとの複合的な利用をしたいと考えると不都合である。

これらの不都合を解消し対象領域での熱の分布を解明するには、まず解析社が解析の理論を理解することである。そして、それを自ら解析プログラムに落とし込むことで、どのプロセスでこういった処理を行っているかを明確になる。こうすることで、将来改良および他のプログラムと連携させることが容易となる。

このようなことを念頭においており、将来の解析プログラムの構築の礎となるところに解析プログラム自作するという本研究の意義があると考ええる。

1.5 論文の構成

本論文ではまず、解析を試みる領域をどのような数式で表現できるか考え、定式化を行い、領域を表す場の方程式を導出する。そして、求めた場の方程式をモデルに、有限要素法の原理の説明・適用を行う。

有限要素法の適用された場の方程式に対して、コンピューターを用いた数値計算を行い、解析領域の解明を試みる。そして最後に計算によって得られた結果が妥当であるか検証を行い、開発した解析システムに欠陥がないかどうか確認を行う。このような構成で本論文は構成される。

1.6 伝熱現象について

自然界において、熱の伝わり方として伝導（拡散）、対流（伝達）、輻射（放射）の3形態がある。伝導（拡散）とは、物質内部の原子や分子の運動によって生じる熱の伝わり方である。対流（伝達）とは、液体や気体が移動することでそれらを介して伝わることである。輻射（放射）とは、太陽光をはじめとする電磁波によって伝わることである。

本研究では、これらの3つの伝熱の形態のうち、伝導と対流について着目し、熱の移流拡散問題を解くこととした。

第2章 熱の移流拡散問題の定式化

本章では熱の伝わり方を定量的に解析するために、伝熱現象の定式化を行う。定式化のプロセスとして、まずある領域に対しての熱の収支を伝導と移流とで考えてそれぞれについて式をたてる。そして、最後に両方が混在する式を考えることである領域での移流拡散による伝熱を式で表現する。

2.1 移流拡散現象の概要

まず、熱の移流拡散現象について説明する。伝熱における拡散とは、温度の高いところから低いところへ熱が散らばる現象である。熱伝導は物質の内部でおこる現象であるが、熱拡散と同様に温度の高いところから低いところへ熱が移動することから、熱伝導と熱拡散は本質的に同じ現象である。したがって、本論文での拡散という言葉は伝導と置いて解釈してもよい。

移流とは、分布を保ったまま流れによって熱が運ばれる現象である。移流拡散は移流と拡散が同時に起こる現象である。これら拡散、移流、移流拡散の3つの現象はそれぞれ以下の図のようなイメージとなる。

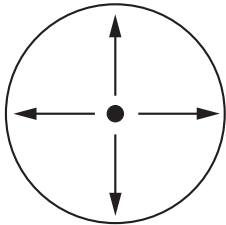


図 2.1: 拡散のイメージ

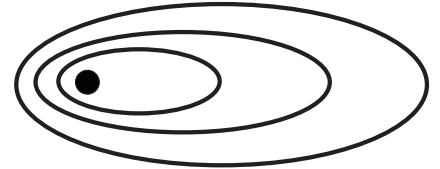


図 2.3: 移流拡散のイメージ

2.2 拡散方程式 (Diffusion equation) の導出

初めに熱の拡散（伝導）方程式 (Diffusion equation) を導出する。ここでいう拡散とは、温度・熱・物質（濃度）・圧力・運動量などが自発的に散らばり広がる現象を表す。

熱の拡散現象はフーリエの法則により記述される。フーリエの法則を言葉で説明すると、次のようになる。

「熱は温度の高い側から低い側へ移動する。そして、その移動する熱は温度差に比例する。」

また、フーリエの法則を式で表すと次のようになる。

$$q_x = -\kappa \frac{\partial T}{\partial x}, \quad q_y = -\kappa \frac{\partial T}{\partial y}, \quad q_z = -\kappa \frac{\partial T}{\partial z} \quad (2.2.1)$$

ここで q_x, q_y, q_z はそれぞれ x, y, z 方向への単位時間・単位断面積あたりの熱流量であり、このことを熱流束 (Heat Flux) という。単位は $[\text{J}/(\text{s} \cdot \text{m}^2) = \text{W}/\text{m}^2]$ である。 κ はカッパと読み熱伝導率 $[\text{J}/(\text{s} \cdot \text{m} \cdot \text{K}) = \text{W}/(\text{m} \cdot \text{K})]$ を表す。（書物によっては λ や k を熱伝導率の記号に用いている。）熱伝導率は単位時間・単位長さ・単位温度あたりで伝わる熱量を表す。

熱伝導率 κ は一般的な材質では x, y, z 方向において全て同じ値で等方性であるが、例えば異なる熱伝導率の材質を貼り合わせて作った板材などは x, y, z 方向への拡散による温度分布はそれぞれ異なる。この場合の熱伝導率を直交異方性熱伝導係数という。また、楕円を回転させたような温度分布をもつ場合を非等方性熱伝導率という。こういった等方性以外の熱伝導率を持つ場合は計算が複雑になる。

そこで、本研究では熱伝導率が等方性の場合のみを取り扱った。非等方性の場合の定式化については文献 [6, pp189–193] などを参考にされたい。また、熱伝導率は温度に依存して変化すると考えられる。しかし、その場合複雑になり解くことが困難である。そのため、本研究では熱伝導率は一定値であるとして取り扱った。

ある位置での温度の変化を考えるために、横幅、奥行き、高さが微小でそれぞれ $dx, dy, dz[\text{m}]$ の長さの検査体積を考える。この検査体積を図 2.4 に示す。

検査体積内の温度を $T[\text{K}]$ 、熱量を $Q[\text{J}]$ とする。微小時間 $dt[\text{s}]$ 経過したとき、この検査体積に x, y, z の各方向から伝導によって入ってくる熱をそれぞれ $Q_{x,\text{in,dif}}, Q_{y,\text{in,dif}}, Q_{z,\text{in,dif}}[\text{J}]$ とする。ここで、dif は拡散の英語 diffusion の意味である。

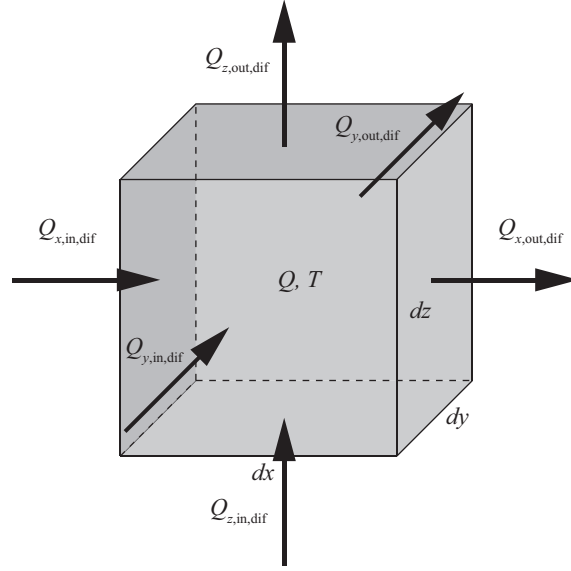


図 2.4: 三次元拡散問題での検査体積

検査体積は直方体であるので、 x 方向に出入りする熱は $dydz$ の面積の分である。同様に、 y 方向は $dxdz$ 、 z 方向へは $dxdy$ の面積分の熱が出入りする。これを考慮して、 x 方向・ y 方向・ z 方向に流入する熱はフーリエの法則を用いて以下のように示される。

$$\begin{cases} Q_{x,\text{in,dif}} = q_x dydzdt = -\kappa \frac{\partial T}{\partial x} dydzdt \\ Q_{y,\text{in,dif}} = q_y dxdzdt = -\kappa \frac{\partial T}{\partial y} dxdzdt \\ Q_{z,\text{in,dif}} = q_z dxdydt = -\kappa \frac{\partial T}{\partial z} dxdydt \end{cases} \quad (2.2.2)$$

同様に、この検査体積の各方向 x, y, z から微小時間 dt の間に流出する熱をそれぞれ $Q_{x,\text{out,dif}}$, $Q_{y,\text{out,dif}}$, $Q_{z,\text{out,dif}}$ [J] とすると、流入してからそれぞれ dx, dy, dz 移動する間に熱が変化することを考慮して流出する熱を求める。流入時は $Q_{x,\text{in,dif}}$, $Q_{y,\text{in,dif}}$, $Q_{z,\text{in,dif}}$ [J] の熱がある。そこからの変化は $Q_{x,\text{in,dif}}$, $Q_{y,\text{in,dif}}$, $Q_{z,\text{in,dif}}$ のそれぞれ x, y, z 方向への傾きに幅 dx, dy, dz を掛けることで $Q_{x,\text{in,dif}}$, $Q_{y,\text{in,dif}}$, $Q_{z,\text{in,dif}}$ からの変化分を算出できる。それらを足すことで $Q_{x,\text{out,dif}}$, $Q_{y,\text{out,dif}}$, $Q_{z,\text{out,dif}}$ は求まる。したがって、以下のように流出する熱は表される。

$$\begin{cases} Q_{x,\text{out,dif}} = Q_{x,\text{in,dif}} + \frac{\partial Q_{x,\text{in,dif}}}{\partial x} dxdt = Q_{x,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial x^2} dxdydzdt \\ Q_{y,\text{out,dif}} = Q_{y,\text{in,dif}} + \frac{\partial Q_{y,\text{in,dif}}}{\partial y} dydt = Q_{y,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial y^2} dxdydzdt \\ Q_{z,\text{out,dif}} = Q_{z,\text{in,dif}} + \frac{\partial Q_{z,\text{in,dif}}}{\partial z} dzdt = Q_{z,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial z^2} dxdydzdt \end{cases} \quad (2.2.3)$$

各方向の流出入により検査体積に蓄積される熱 Q_x, Q_y, Q_z は流入から流出を引くことで求まるので、これら $Q_{x,\text{in,dif}}$, $Q_{y,\text{in,dif}}$, $Q_{z,\text{in,dif}}$ と $Q_{x,\text{out,dif}}$, $Q_{y,\text{out,dif}}$, $Q_{z,\text{out,dif}}$ の差で表される。

$$\begin{cases} Q_x = Q_{x,\text{in,dif}} - Q_{x,\text{out,dif}} = Q_{x,\text{in,dif}} - \left(Q_{x,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial x^2} dxdydzdt \right) = \kappa \frac{\partial^2 T}{\partial x^2} dxdydzdt \\ Q_y = Q_{y,\text{in,dif}} - Q_{y,\text{out,dif}} = Q_{y,\text{in,dif}} - \left(Q_{y,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial y^2} dxdydzdt \right) = \kappa \frac{\partial^2 T}{\partial y^2} dxdydzdt \\ Q_z = Q_{z,\text{in,dif}} - Q_{z,\text{out,dif}} = Q_{z,\text{in,dif}} - \left(Q_{z,\text{in,dif}} - \kappa \frac{\partial^2 T}{\partial z^2} dxdydzdt \right) = \kappa \frac{\partial^2 T}{\partial z^2} dxdydzdt \end{cases} \quad (2.2.4)$$

検査体積全体での熱の蓄積はこれら Q_x, Q_y, Q_z の和であるので以下のようになる。

$$\begin{aligned} Q &= Q_x + Q_y + Q_z = \kappa \frac{\partial^2 T}{\partial x^2} dxdydzdt + \kappa \frac{\partial^2 T}{\partial y^2} dxdydzdt + \kappa \frac{\partial^2 T}{\partial z^2} dxdydzdt \\ &= \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) dxdydzdt \end{aligned} \quad (2.2.5)$$

ある系の温度を 1K 上昇させるのに必要な熱量 $\frac{dQ}{dT}$ を熱容量 C [J/K] という。熱容量は定圧条件下の場合と定積条件下の場合とで値が異なる。これらを区別するために、それぞれ C_p, C_v という記号が用いられる。また、熱容量は物質によって異なる値を示し、ある物質の単位質量あたりの熱容量である比熱容量 c [J/(kg·K)] とその物質の質量 m [kg] の積として表される。比熱容量もまた定圧条件下か定積条件下によって値がことなるため区別しそれぞれ c_p, c_v という記号が用いられる。

ここで、以下で示すマイヤーの関係式より定圧比熱と定積比熱は結び付けられる。

$$c_p - c_v = R \quad (2.2.6)$$

R は気体定数である。したがって、比熱の値としては定積比熱でも定圧比熱であっても換算できるためどちらを使っても問題ない。本研究では定圧比熱容量 c_p を使う。

検査体積の質量を m [kg]、定圧比熱容量を c_p [J/(kg·K)]、温度変化 dT [K] とすると、検査体積全体での熱は熱容量の定義より

$$Q = mc_p dT \quad (2.2.7)$$

に等しい。

検査体積の体積が $dxdydz$ [m³] であるので質量は $m = \rho dxdydz$ [kg] となる。なお、ここで ρ は検査体積の密度 [kg/m³] である。一般的には密度は圧力や温度に依存する。しかし、そう考えると計算が複雑になるため密度は一定とする。同様に、定圧比熱容量も一般的には圧力や温度の関数として表されるが、一定として考える。これらを式 (2.2.7) に当てはめると次のように表すことができる。

$$Q = \rho c_p dxdydz dT \quad (2.2.8)$$

これら式 (2.2.5) および式 (2.2.8) は保存則より等しいので、

$$\kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) dxdydz dt = \rho c_p dxdydz dT \quad (2.2.9)$$

が成り立つ。これを整理する。

$$\frac{dT}{dt} = \frac{\kappa}{\rho c_p} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (2.2.10)$$

ここで

$$k = \frac{\kappa}{\rho c_p} \quad (2.2.11)$$

とする。 k は温度伝導率や熱拡散率、温度拡散率といい、単位は [m²/s] である。

左辺は T が多変数関数なので、

$$\frac{dT}{dt} \rightarrow \frac{\partial T}{\partial t} \quad (2.2.12)$$

となる。

これらを当てはめることで、以下の式で示される三次元非定常熱伝導方程式が導出される。

$$\frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (2.2.13)$$

表記の簡略のため、ラプラシアン： $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ を用いて式 (2.2.13) を表すと以下ようになる。

$$\frac{\partial T}{\partial t} = k \Delta T \quad (2.2.14)$$

ここまでで、拡散方程式の導出ができた。

2.3 移流方程式 (Advection equation) の導出

移流とは流れによって引き起こされる現象である。物質や圧力・温度・エネルギーなどの物理量が、分布を保ったまま流れによって運ばれることを移流と呼ぶ。

拡散方程式の導出と同様に幅、奥行、高さがそれぞれ dx, dy, dz の検査体積を考え、移流方程式 (Advection equation) を導出する。この検査体積を図 2.5 に示した。

検査体積内の温度を T [K]、熱量を Q [J] とする。微小時間 dt [s] 経過したとき、この検査体積に x, y, z の各方向から入ってくる熱をそれぞれ $Q_{x,\text{in},\text{ad}}, Q_{y,\text{in},\text{ad}}, Q_{z,\text{in},\text{ad}}$ [J] とする。ここで ad は移流の英語 advection を表す。

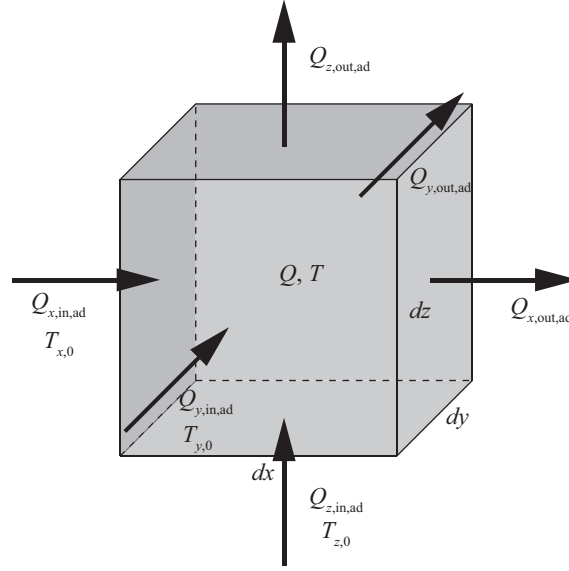


図 2.5: 三次元移流問題での検査体積

x, y, z 方向からの流入による検査体積の内部と外部との温度変化をそれぞれ dT_x, dT_y, dT_z と表し、温度の基準点を流入する流体の直前の温度 $T_{x,0}, T_{y,0}, T_{z,0}$ (これらは定数) とする。このように考えると、 $T = T(x, y, z, t)$ は基準点とそこからの変化分の和で表すことができ以下ようになる。

$$\begin{cases} T = T_{x,0} + dT_x \\ T = T_{y,0} + dT_y \\ T = T_{z,0} + dT_z \end{cases} \quad (2.3.1)$$

各方向からの流速を v_x, v_y, v_z [m/s] とすると、移流によって流入する熱はそれぞれ熱容量の関係式 (2.2.7) において、移流によって運ばれる質量分である。質量は密度と体積の積として表され、密度は決まっているので流れによって運ばれる体積分の熱が流入する熱である。流れによって運ばれる体積とは、例えば x 方向については断面積が $dydz$ であるため、 dt の時間で流入する体積は $v_x dydzdt$ [m³] である。 y, z 方向についても同様に考える。したがって、移流によって流入する熱 $Q_{x,in,ad}, Q_{y,in,ad}, Q_{z,in,ad}$ は式で書くと以下ようになる。

$$\begin{cases} Q_{x,in,ad} = \rho c_p dT_x v_x dydzdt \\ Q_{y,in,ad} = \rho c_p dT_y v_y dx dzdt \\ Q_{z,in,ad} = \rho c_p dT_z v_z dx dydt \end{cases} \quad (2.3.2)$$

本来ならば流速も位置や時間によって変化する関数であると考えられる。しかし、速度が時間や位置で変化する関数であると複雑になり計算が困難になるので、本研究では速度は既知の定数とした。

出ていく熱 $Q_{x,out,ad}, Q_{y,out,ad}, Q_{z,out,ad}$ は、検査体積内を移動する間に変化することを考慮して以下ようになる。

$$\begin{cases} Q_{x,out,ad} = Q_{x,in,ad} + \frac{\partial Q_{x,in,ad}}{\partial x} dx = Q_{x,in,ad} + \rho c_p \frac{\partial(dT_x)}{\partial x} dx dy dz dt \\ Q_{y,out,ad} = Q_{y,in,ad} + \frac{\partial Q_{y,in,ad}}{\partial y} dy = Q_{y,in,ad} + \rho c_p \frac{\partial(dT_y)}{\partial y} dx dy dz dt \\ Q_{z,out,ad} = Q_{z,in,ad} + \frac{\partial Q_{z,in,ad}}{\partial z} dz = Q_{z,in,ad} + \rho c_p \frac{\partial(dT_z)}{\partial z} dx dy dz dt \end{cases} \quad (2.3.3)$$

検査体積で各方向からの流入出で蓄積される熱 Q_x, Q_y, Q_z はこれらの差である。よって、次のようになる。

$$\begin{cases} Q_x = Q_{x,in,ad} - Q_{x,out,ad} = Q_{x,in,ad} - \left(Q_{x,in,ad} + \rho c_p v_x \frac{\partial(dT_x)}{\partial x} dx dy dz dt \right) = -\rho c_p v_x \frac{\partial(dT_x)}{\partial x} dx dy dz dt \\ Q_y = Q_{y,in,ad} - Q_{y,out,ad} = Q_{y,in,ad} - \left(Q_{y,in,ad} + \rho c_p v_y \frac{\partial(dT_y)}{\partial y} dx dy dz dt \right) = -\rho c_p v_y \frac{\partial(dT_y)}{\partial y} dx dy dz dt \\ Q_z = Q_{z,in,ad} - Q_{z,out,ad} = Q_{z,in,ad} - \left(Q_{z,in,ad} + \rho c_p v_z \frac{\partial(dT_z)}{\partial z} dx dy dz dt \right) = -\rho c_p v_z \frac{\partial(dT_z)}{\partial z} dx dy dz dt \end{cases} \quad (2.3.4)$$

検査体積に実際に蓄積される熱 Q はこれらの Q_x, Q_y, Q_z の和である。したがって、次のようになる。

$$\begin{aligned} Q &= -\rho c_p v_x \frac{\partial(dT_x)}{\partial x} dx dy dz dt - \rho c_p v_y \frac{\partial(dT_y)}{\partial y} dx dy dz dt - \rho c_p v_z \frac{\partial(dT_z)}{\partial z} dx dy dz dt \\ &= -\rho c_p \left(v_x \frac{\partial(dT_x)}{\partial x} + v_y \frac{\partial(dT_y)}{\partial y} + v_z \frac{\partial(dT_z)}{\partial z} \right) dx dy dz dt \end{aligned} \quad (2.3.5)$$

ここで式 (2.3.1) より dT_x, dT_y, dT_z を置き直す。

$$\begin{aligned} Q &= -\rho c_p \left(v_x \frac{\partial(T - T_{x,0})}{\partial x} + v_y \frac{\partial(T - T_{y,0})}{\partial y} + v_z \frac{\partial(T - T_{z,0})}{\partial z} \right) dx dy dz dt \\ &= -\rho c_p \left(v_x \frac{\partial T}{\partial x} - v_x \frac{\partial T_{x,0}}{\partial x} + v_y \frac{\partial T}{\partial y} - v_y \frac{\partial T_{y,0}}{\partial y} + v_z \frac{\partial T}{\partial z} - v_z \frac{\partial T_{z,0}}{\partial z} \right) dx dy dz dt \\ &= -\rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) dx dy dz dt \end{aligned} \quad (2.3.6)$$

この式は、検査体積全体で熱の流入の前後で比熱容量から定義される熱量と等しいと考えることができる。(この場合の dT は、検査体の流入前の温度 T_f と流入後の温度 T_l の差、つまり流入の前後での温度差と考えることができる。) したがって、次のようになる。

$$Q = -\rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) dx dy dz dt = \rho c_p dT dx dy dz \quad (2.3.7)$$

この式を整理する。

$$\frac{dT}{dt} + \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) = 0 \quad (2.3.8)$$

ここで、 T は多変数関数なので $\frac{dT}{dt} \rightarrow \frac{\partial T}{\partial t}$ となり、また $\mathbf{v} = (v_x, v_y, v_z)$ と演算子 $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ を用いて以下のように表すことができる。

$$\frac{\partial T}{\partial t} + \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) = \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = 0 \quad (2.3.9)$$

このようにして三次元非定常移流方程式が求まった。

2.4 移流拡散方程式 (Advection Diffusion equation) の導出

ここから、移流と拡散の両方が存在する移流拡散方程式 (Advection Diffusion equation) を導出していく。移流や拡散方程式を導出したときと同様に、検査体積を考えてそれに流入する熱と流出する熱の収支を考えることで式を導出していく。まず、これまでと同様に幅、奥行、高さがそれぞれ dx, dy, dz の検査体積を考える。この検査体積を図 2.6 に示した。

移流と拡散の両方が存在するため、流入する熱は拡散 (伝導) によるものと移流によって流れ込んでくるものの2種類が考えられる。、これらを考慮して収支の式を立てる。

dt 時間でこの検査体に入ってくる熱を方向ごとに $Q_{x,in}, Q_{y,in}, Q_{z,in}$ とする。流入する熱は伝導による分 $Q_{in,dif}$ と、移流による分 $Q_{in,ad}$ とが考えられる。

伝導により x, y, z の各方向から流入する熱 $Q_{x,in,dif}, Q_{y,in,dif}, Q_{z,in,dif}$ はフーリエの法則により、以下のように表される。

$$\begin{cases} Q_{x,in,dif} = q_x dy dz dt = -\kappa \frac{\partial T}{\partial x} dy dz dt \\ Q_{y,in,dif} = q_y dx dz dt = -\kappa \frac{\partial T}{\partial y} dx dz dt \\ Q_{z,in,dif} = q_z dx dy dt = -\kappa \frac{\partial T}{\partial z} dx dy dt \end{cases} \quad (2.4.1)$$

各方向からの流速を v_x, v_y, v_z とする。 x, y, z 方向からの流入による検査体積の内部と外部との温度変化をそれぞれ dT_x, dT_y, dT_z と表し、温度の基準点を流入する流体の直前の温度 $T_{x,0}, T_{y,0}, T_{z,0}$ (これらは定数) とする。このように考えると、 $T = T(x, y, z, t)$ は基準点とそこからの変化分の和で表すことができ以下のようなになる。

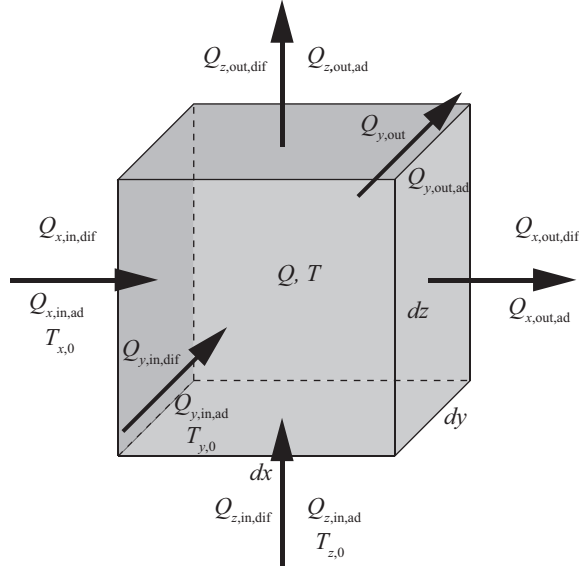


図 2.6: 三次元移流拡散問題での検査体積

$$\begin{cases} T = T_{x,0} + dT_x \\ T = T_{y,0} + dT_y \\ T = T_{z,0} + dT_z \end{cases} \quad (2.4.2)$$

移流によって x, y, z の各方向から流入する熱 $Q_{x,in,ad}, Q_{y,in,ad}, Q_{z,in,ad}$ は、熱容量の定義より次のように表される。

$$\begin{cases} Q_{x,in,ad} = \rho c_p dT_x v_x dy dz dt \\ Q_{y,in,ad} = \rho c_p dT_y v_y dx dz dt \\ Q_{z,in,ad} = \rho c_p dT_z v_z dx dy dt \end{cases} \quad (2.4.3)$$

各方向から流入する熱は伝導と移流によるものとの和である。したがって以下のようになる。

$$\begin{cases} Q_{x,in} = Q_{x,in,dif} + Q_{x,in,ad} = -\kappa \frac{\partial T}{\partial x} dy dz dt + \rho c_p dT_x v_x dy dz dt \\ Q_{y,in} = Q_{y,in,dif} + Q_{y,in,ad} = -\kappa \frac{\partial T}{\partial y} dx dz dt + \rho c_p dT_y v_y dx dz dt \\ Q_{z,in} = Q_{z,in,dif} + Q_{z,in,ad} = -\kappa \frac{\partial T}{\partial z} dx dy dt + \rho c_p dT_z v_z dx dy dt \end{cases} \quad (2.4.4)$$

一方で、流出する熱 $Q_{x,out}, Q_{y,out}, Q_{z,out}$ は、検査体内を移動する間に変化することを考慮して、次の式となる。

$$\begin{cases} Q_{x,out} = Q_{x,in} + \frac{\partial Q_{x,in}}{\partial x} dx \\ Q_{y,out} = Q_{y,in} + \frac{\partial Q_{y,in}}{\partial y} dy \\ Q_{z,out} = Q_{z,in} + \frac{\partial Q_{z,in}}{\partial z} dz \end{cases} \quad (2.4.5)$$

検査体で各方向からの流入出で蓄積される熱 Q_x, Q_y, Q_z はこれらの差である。よって、次のようになる。

$$\begin{cases} Q_x = Q_{x,in} - Q_{x,out} = -\frac{\partial Q_{x,in}}{\partial x} dx = \kappa \frac{\partial^2 T}{\partial x^2} dx dy dz dt - \rho c_p v_x \frac{\partial(dT_x)}{\partial x} dx dy dz dt \\ Q_y = Q_{y,in} - Q_{y,out} = -\frac{\partial Q_{y,in}}{\partial y} dy = \kappa \frac{\partial^2 T}{\partial y^2} dx dy dz dt - \rho c_p v_y \frac{\partial(dT_y)}{\partial y} dx dy dz dt \\ Q_z = Q_{z,in} - Q_{z,out} = -\frac{\partial Q_{z,in}}{\partial z} dz = \kappa \frac{\partial^2 T}{\partial z^2} dx dy dz dt - \rho c_p v_z \frac{\partial(dT_z)}{\partial z} dx dy dz dt \end{cases} \quad (2.4.6)$$

検査体に実際に蓄積される熱 Q はこれらの Q_x, Q_y, Q_z の和である。したがって、これらを足すと次のようになる。

$$\begin{aligned} Q &= \kappa \frac{\partial^2 T}{\partial x^2} dx dy dz dt - \rho c_p v_x \frac{\partial(dT_{x,in})}{\partial x} dx dy dz dt + \kappa \frac{\partial^2 T}{\partial y^2} dx dy dz dt - \rho c_p v_y \frac{\partial(dT_{y,in})}{\partial y} dx dy dz dt \\ &\quad + \kappa \frac{\partial^2 T}{\partial z^2} dx dy dz dt - \rho c_p v_z \frac{\partial(dT_{z,in})}{\partial z} dx dy dz dt \\ &= \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial(dT_{x,in})}{\partial x} + v_y \frac{\partial(dT_{y,in})}{\partial y} + v_z \frac{\partial(dT_{z,in})}{\partial z} \right) \right\} dx dy dz dt \end{aligned} \quad (2.4.7)$$

ここで式 (2.4.2) より dT_x, dT_y, dT_z を置きなおすと、以下ようになる。

$$Q = \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial(T - T_{x,0})}{\partial x} + v_y \frac{\partial(T - T_{y,0})}{\partial y} + v_z \frac{\partial(T - T_{z,0})}{\partial z} \right) \right\} dx dy dz dt$$

$$Q = \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \left(\frac{\partial T}{\partial x} - \frac{\partial T_{x,0}}{\partial x} \right) + v_y \left(\frac{\partial T}{\partial y} - \frac{\partial T_{y,0}}{\partial y} \right) + v_z \left(\frac{\partial T}{\partial z} - \frac{\partial T_{z,0}}{\partial z} \right) \right) \right\} dx dy dz dt$$

$$Q = \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) \right\} dx dy dz dt \quad (2.4.8)$$

ここで、検査体積内で化学反応によって生じる発熱や吸熱、また放射性崩壊による発熱や吸熱、さらには太陽光などによる輻射によって発熱する現象をひっくるめて発熱率 $f = f(x, y, z, t)$ という記号を定義する。発熱率は単位体積、単位時間で生じる熱を表し単位は $[J/(m^3 \cdot s)]$ である。

この発熱率を考慮すると、検査体積内の熱 Q は式 (2.4.8) で表される移流と拡散によって蓄積される熱にさらに検査体積での体積での発熱率を足すことで表現できる。よって、発熱率を考慮した式を次のように示す。

$$Q = \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) \right\} dx dy dz dt + f dx dy dz dt \quad (2.4.9)$$

$$= \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) + f \right\} dx dy dz dt \quad (2.4.10)$$

この式は検査体積全体で熱の流入の前後で比熱容量の定義による熱量と等しいので、以下ようになる。

$$Q = \left\{ \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) + f \right\} dx dy dz dt = \rho c_p dT dx dy dz \quad (2.4.11)$$

$k = \kappa/(\rho c_p)$ などを利用してこの式を整理すると、以下ようになる。

$$\frac{dT}{dt} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) + \frac{f}{\rho c_p} \quad (2.4.12)$$

$$\frac{dT}{dt} + \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) = \frac{f}{\rho c_p} \quad (2.4.13)$$

ここで、 T は多変数関数なので $\frac{dT}{dt} \rightarrow \frac{\partial T}{\partial t}$ とし、 $\mathbf{v} = (v_x, v_y, v_z)$ と演算子 $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$ と $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ を用いて以下のように表せる。

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T - k \Delta T = \frac{f}{\rho c_p} \quad (2.4.14)$$

このようにして、三次元非定常移流拡散方程式が求まった。

2.5 本研究で扱う問題

ここまでで、移流拡散方程式の一般式である三次元非定常移流拡散方程式が式 (2.4.13) のように求まった。しかし、この問題は非定常であるため定常の問題が解析できていないのにいきなり取り組むことは無謀である。また、三次元問題であるため扱うデータが多く、計算の確認を行うことが困難となることが予想される。しかし、一次元問題であれば理論解が比較的簡単に求まるため解析のプログラムを作る必要性が小さい。

また、発熱率 f は移流拡散の本質ではないため $f = 0$ とした。基本となる移流拡散問題の解析ができたのちに発熱率の考慮、次元の拡大、非定常問題の処理を行っていくべきだと考えた。

したがって、これらのことを考慮して本研究では以下の式で示される二次元定常移流拡散問題を取り扱うこととした。

$$v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 \quad (2.5.1)$$

第3章 有限要素法の適用

3.1 有限要素法とは

有限要素法 (Finite Element Method, FEM) は微分方程式を近似的に解くための数値解析の手法である。有限要素法のおおまかな流れを説明する。領域を要素とよばれる小領域に分割を行う。そして要素ごとに式を立てて計算を行う。領域全体は全ての要素について足し合わせることで表現する。このような流れで計算を行う手法であり、領域を要素に分割するところが特徴的である。

有限要素法の大元となる考え方として重み付残差法がある。さらに、重み付残差法にもいくつかの種類が存在し、有限要素法では Galerkin 法を用いる。したがって、有限要素法とは重み付残差法のうち Galerkin 法を用いるものであるというように数学的に定義することができる。

重み付残差法とは、次のような考えで問題を解く手法である。

対象とする場の方程式に対して、近似的な解を代入する。近似的な解を代入するため当然誤差が生じる。この誤差がなくなるように重みをかけて、さらに領域全体で足し合わせて0になるようにする。未知の関数は形のわかっている関数がある定数倍して足し合わせて合成することで作ることができるという考えに則って、初めに定めた近似解を形のわかっている関数がある定数倍したものとの和とする。また、同様に重みについても形のわかっている関数がある定数倍したものとの和の合成とする。こうすることで、問題がこれらの係数の値を求めるものに帰結される。

Galerkin 法は近似関数と重み関数に同じ関数を用いる方法である。これらの形のわかっている関数はもともとの場の方程式を表わす関数が満たす要件である、境界条件を満足するために決める必要がある。なぜならば、境界条件を満たさないと近似したとは言えないからである。そのためには境界ではその境界値を持つように近似解を構成する関数を定めなくてはならない。単純な要素でならば、関数を定めるのは簡単である。しかし、複雑な領域の場合は関数を定めるのは困難である。そこで、領域を比較的単純な要素に分解して、ここの要素について近似解の関数を設定する。それらを足し合わせることで、領域全体の解が値の分布として求まる。

この要素に分割する段階で、場の方程式が要素の節点ごとに適用される。そのため、領域全体の節点数分の方程式が形成される。この多次元連立方程式を解き、近似関数を構成する係数を求めることで場の方程式を要素ごとに解いたこととなる。このような考えに則って、有限要素法は構成される。

この流れを念頭において、実際に本研究で取り扱う二次元定常移流拡散方程式に有限要素法を適用させる。

3.2 二次元定常移流拡散方程式への有限要素法の適用

3.2.1 場の方程式

移流拡散方程式を有限要素法で解析するために、有限要素式を導出する。本研究で取り扱う二次元定常移流拡散方程式は以下で示される。

$$\begin{cases} v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 & (\Omega \text{内}) \\ T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial n} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (3.2.1)$$

$$(3.2.2)$$

Ω は領域全体を表す。領域の全境界を Γ として、 Γ_1, Γ_2 はそれぞれ境界のある部分を表し、それぞれ重複はなく $\Gamma = \Gamma_1 + \Gamma_2$ を満たす。 g_1, g_2 は各境界 Γ_1, Γ_2 内で成立する関数である。また、式 (3.2.2) の第一項は第一種境界条件 (Dirichlet 境界条件、ディリクレ境界条件)、第二項は第二種境界条件 (Neumann 境界条件、ノイマン境界条件) と一般的に呼ばれている。 n は外向き法線方向を表す。ここで、各種の文字および記号は表 3.1 に示した。以下では、この式と条件に基づいて有限要素法を適用していく。

3.2.2 重み付き残差法 (Weighted Residual Method, WRM) の適用

この問題で求めたい値は温度 $T[K]$ である。しかし、 T は位置の関数で場の方程式が二階偏微分方程式であるため導出が困難である。そこで、 T に領域内でのある座標点 $i(x_i, y_i)$ での近似解を $\hat{T}$ として考える。そうすると、 $\hat{T}$ は近似であるため $T \neq \hat{T}$ となり、誤差が生じ左辺 $\neq 0$ となってしまう。ここで、この近似解の右辺を残差 ε_i とすると、次のようになる。

表 3.1: 記号の意味

記号	名称	単位	定義や条件
T	温度	K	位置に依存する未知関数 $T(x, y)$
x, y	基準軸	m	対象領域の位置を示す
v_x, v_y	x, y 方向への流速	m/s	既知の定数とする
$\mathbf{n}$	単位法線ベクトル	m	$\mathbf{n} = (n_x, n_y)$
k	温度伝導率 (熱拡散率、温度拡散率)	m^2/s	$k = \kappa/(\rho c_p)$
κ	熱伝導率	$\text{W}/(\text{m} \cdot \text{K})$	材質内で一定値で等方性とする
ρ	密度	kg/m^3	材質内で一定値とする
c_p	定圧比熱容量	$\text{J}/(\text{kg} \cdot \text{K})$	材質内で一定値とする
g_1	第一種境界条件値	K	一定値とする
g_2	第二種境界条件値	K/m	一定値とする

$$\left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) - k \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) = \varepsilon_i \quad (3.2.3)$$

この残差 (Residual) ε をどうにかして 0 にしたいので、点 i での重み関数 w_i をかけて重み付き残差の和が領域全体で 0 になるように考える。そうすると以下ようになる。

$$\sum_{i=1}^n w_i \varepsilon_i = 0 \quad (3.2.4)$$

ここで n は考える領域全体での地点数である。この式 (3.2.4) は式 (3.2.1) と近似 $\approx$ である。シグマ記号は式 (3.2.1) が関数なので領域での積分になる。したがって、これを式で表すと以下ようになる。

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w d\Omega - k \int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega = 0 \quad (3.2.5)$$

このようにして、関数に近似解を代入してさらに重みをかけた和の誤差を 0 にするように式を立てることを**重み付き残差法 (Weighted Residual Method, WRM)**と呼び、式 (3.2.5) のように重みをかけて積分した式を**重み付き残差方程式**や**重み付き残差式**と呼んでいる。

求めたい T の近似として $\hat{T}$ を導入した。この近似の仕方として、予め形のわかっている関数を任意定数倍して足し合わせることで求めたい T の近似とする方法がある。ここで任意のある (既知) 関数を ϕ_i 、その関数の係数を a_i とすると T の近似関数 $\hat{T}$ は以下のような式になる。

$$\hat{T} = \phi_0 + \sum_{i=1}^m a_i \phi_i \quad (3.2.6)$$

ここで、 ϕ_0 は式 (3.2.2) の第一項である Dirichlet 条件を満たすように関数として設定され、同様に ϕ_i も Dirichlet 条件を満たすように設定される。これを式で示すと以下になる。

$$\Gamma_1 \text{ 上で } \phi_0 \approx g_1, \quad \phi_i = 0 \quad (1 \leq i \leq m) \quad (3.2.7)$$

m は近似に用いる関数 ϕ_i の数を表す。なお、 ϕ_i は互いに定数倍しても同じにならない条件 (このことを一次独立という) を満たす必要がある。なお、この ϕ_i を基底関数、 $\hat{T}$ を近似関数や試行関数、試験関数と呼んでいる。

一方、重み関数 w のみたす条件として以下の 3 つがあげられる。

$$\begin{cases} \cdot \text{重み関数は領域内で連続 (正則) で微分可能である} & (3.2.8) \\ \cdot \text{重み関数は境界上で 0 となる} & (3.2.9) \\ \cdot \text{重み関数は無次元である} & (3.2.10) \end{cases}$$

1 つ目の条件は領域全体で積分を実行するために必要な条件である。

2 つ目の条件は、文献 [5, p.16] に次のように記されている。

“一般には重み付き残差法のうちで、境界条件は満足するが場の方程式を満足しない試行関数を用いる方法を内部法 (interior method)、場の方程式を満足するが境界条件を満足しない試行関数を用いる方法を境界法 (boundary method)、場の方程式も境界条件も満足しない試行関数を用いる方法を混合法 (mixed method) とよぶ。”

境界条件を課したにも関わらず、境界で誤差が生じるのは困るので、境界で誤差が 0 となる内部法を採用する。そのため、2 つ目の条件が必要である。3 つ目の条件は、重み関数には物理的な意味がなく、誤差の調整以外に与式に影響を与えては困るので次元数はない。

内部法を採用することで、境界では値が固定される。また、このあとの式を展開していく上での基本となることから、この場合の第一種境界条件 (Dirichlet 境界条件) を特に**基本境界条件**や**固定境界条件**と呼んでいる。

3.2.3 弱形式 (weak form) の導出

いま考える式は、重み付き残差方程式である

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w d\Omega - k \int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega = 0 \quad (3.2.11)$$

に置き換わった。

この式は二階微分の項が入っており、解を求めるのが困難になっている。未知関数の二階微分項が存在すると、あとで未知関数を近似式で表す時に近似するための式の条件がきつくなってしまう問題となる。そこで、多変数関数の部分積分の公式である **Green** の公式またはガウスの発散定理

$$\int_{\Omega} \frac{\partial u}{\partial x_i} w d\Omega = \int_{\Gamma} n_i u w d\gamma - \int_{\Omega} u \frac{\partial w}{\partial x_i} d\Omega \quad (3.2.12)$$

を用いて微分の階数を落とした弱形式を導出する。いま扱う問題はこの公式における $i = 1 \sim 2$ に対応しており、 u は $\hat{T}$ に、 x_1, x_2 はそれぞれ x, y に対応し、また、 n_1, n_2 は単位法線ベクトルの各成分 n_x, n_y に対応する。

Green の公式において、 $u = \partial T / \partial x$ とすると次のような式となる。

$$\int_{\Omega} \frac{\partial}{\partial x} \left(\frac{\partial T}{\partial x} \right) w dx = \int_{\Gamma} n_x \frac{\partial T}{\partial x} w d\gamma - \int_{\Omega} \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} d\Omega \quad (3.2.13)$$

この公式を使って、二階微分の項 $\int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega$ を変形していく。まず、この項を公式に則って展開する。

$$\int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega = \int_{\Gamma} n_x \frac{\partial \hat{T}}{\partial x} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega + \int_{\Gamma} n_y \frac{\partial \hat{T}}{\partial y} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \quad (3.2.14)$$

この式の左辺 $\int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega$ がくどいので（二階微分の項）と以下の変形の間は表記する。

ここで、 $\Gamma = \Gamma_1 + \Gamma_2$ より、

$$\begin{aligned} (\text{二階微分の項}) &= \int_{\Gamma_1 + \Gamma_2} n_x \frac{\partial \hat{T}}{\partial x} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega + \int_{\Gamma_1 + \Gamma_2} n_y \frac{\partial \hat{T}}{\partial y} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \\ &= \int_{\Gamma_1} n_x \frac{\partial \hat{T}}{\partial x} w d\gamma + \int_{\Gamma_2} n_x \frac{\partial \hat{T}}{\partial x} w d\gamma + \int_{\Gamma_1} n_y \frac{\partial \hat{T}}{\partial y} w d\gamma + \int_{\Gamma_2} n_y \frac{\partial \hat{T}}{\partial y} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \end{aligned} \quad (3.2.15)$$

となる。本研究では w は内部法を採用するので境界である Γ_1 上で 0 であるから

$$(\text{二階微分の項}) = \int_{\Gamma_2} n_x \frac{\partial \hat{T}}{\partial x} w d\gamma + \int_{\Gamma_2} n_y \frac{\partial \hat{T}}{\partial y} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \quad (3.2.16)$$

となる。また、

$$\frac{\partial T}{\partial \mathbf{n}} = n_x \frac{\partial T}{\partial x} + n_y \frac{\partial T}{\partial y} \quad (3.2.17)$$

より

$$(\text{二階微分の項}) = \int_{\Gamma_2} \frac{\partial \hat{T}}{\partial \mathbf{n}} w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \quad (3.2.18)$$

である。ここで、式 (3.2.2) の第二項の自然境界条件

$$\frac{\partial \hat{T}}{\partial \mathbf{n}} = g_2 \quad (\Gamma_2 \text{上})$$

より以下のようになる。

$$(\text{二階微分の項}) = \int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) w d\Omega = \int_{\Gamma_2} g_2 w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \quad (3.2.19)$$

これをもとの式 (3.2.11) に代入することで次の式となる。

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w d\Omega - k \left(\int_{\Gamma_2} g_2 w d\gamma - \int_{\Omega} \frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} d\Omega - \int_{\Omega} \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} d\Omega \right) = 0 \quad (3.2.20)$$

この式を整理すると

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w d\Omega + k \int_{\Omega} \left(\frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} + \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} \right) d\Omega = k \int_{\Gamma_2} g_2 w d\gamma \quad (3.2.21)$$

となる。

このように、二階偏微分の式を一階微分の式に変形することができた。この式は、微分の階数が弱められていることから**弱形式 (weak form)**と呼ばれている。今回は元の式に二階微分の項があったため、弱形式を求める中で Neumann 境界条件が式 (3.2.21) の右辺に組み込まれた。このように、自然と Neumann 境界条件が組み込まれていることからこの場合の Neumann 境界条件を**自然境界条件**と呼んでいる。このように、弱形式を求める段階で自然境界条件が組み込まれていることで、後に Neumann 境界条件を組み込むことが容易になっている。

もし、元の式において一階部分の項しか存在しない場合は、重み付き残差法を適用する際に、式 (3.2.5) の後にさらに次の処理を施す。式 (3.2.2) の第二項である Neumann 境界条件の両辺に重み関数 w をかけて、さらに Γ_2 で積分をしたものを式 (3.2.5) に足す。こうすることで二階微分の項が元の式に無くても Neumann 境界条件を組み込むことができる。

3.2.4 Galerkin 法の適用

重み付き残差法の中で、重み関数 w をどう設定するかでいくつかの方式がある。その中で、有限要素法では一般的に Galerkin 法が使われている。

Galerkin 法とは、式 (3.2.6) で表される近似関数 $\hat{T}$ と同じ基底関数 ϕ_i を重み関数 w にも導入し、それを任意定数倍して足し合わせることで重み関数 w を近似する方法である。

w の近似関数を $\hat{w}$ として、係数 b_i とすると以下の式で表される。

$$\hat{w} = \sum_{i=1}^m b_i \phi_i \quad (1 \leq i \leq m) \quad (3.2.22)$$

なお、 $\hat{w}$ は $\hat{T}$ と同じ基底関数を用いているので、

$$\phi_i(\Gamma_1) = 0 \quad (1 \leq i \leq m) \quad (3.2.23)$$

となり、(3.2.9) より内部法を用いているので重み関数の近似形 $\hat{w}$ もまた、

$$\hat{w} = 0 \quad (\Gamma_1 \text{ 上}) \quad (3.2.24)$$

を満たす。これを、式 (3.2.21) にあてはめる。

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) \hat{w} d\Omega + k \int_{\Omega} \left(\frac{\partial \hat{T}}{\partial x} \frac{\partial \hat{w}}{\partial x} + \frac{\partial \hat{T}}{\partial y} \frac{\partial \hat{w}}{\partial y} \right) d\Omega = k \int_{\Gamma_2} g_2 \hat{w} d\gamma \quad (3.2.25)$$

この式のように、重み付き残差法と Galerkin 法によって、未知の関数 T はその近似関数 $\hat{T}$ 、 $\hat{w}$ に置き換わり、基底関数 ϕ は境界条件を満たすように決めればよいので、これらの係数 a_i, b_i を求める問題へと帰着した。

しかし、基底関数 ϕ をどのように決めるかという問題が生じる。単純な領域であれば、近似関数が境界条件を満たすように、基底関数を決めることは比較的容易である。しかし、複雑な領域を扱う場合にそのまま基底関数を決定するのは困難である。この問題を解決するために、領域を**要素**と呼ばれる小領域に分割してそれぞれの要素について近似関数を考える。こうすることで、基底関数をもとの領域に比べて容易に設定することができる。領域全体については、要素ごとに求めた結果を全要素について足し合わせることで求めることができる。

要素分割のしかたはさまざまである。二次元問題では三角形や四角形に要素分割が行われる。今回は領域が四角形要素だとして、図 3.1 に示すような四角形要素 16 個で要素分割を行うものとして進める。

要素の端点を**節点 (nodal point または node)**といい、節点には節点値として節点 i には未知数が T_i として与えられている。要素と節点にはそれぞれ順番に番号をつける。これらを**要素番号 (element number)** 及び**節点番号 (node number)** とよぶ。また、各要素ごとに節点番号をつけ、これを**要素節点番号 (node number of element)** とよぶ。図 3.1 においては、 $[]$ でくくられた数字は要素番号を表し、単なる数字は節点番号を表す。 $*$ のついた数字は要素節点番号を表し、 e のついた文字 $T_i^e, i = 1 \sim 4$ はある要素番号 e である要素の要素節点番号 $i^*, i^* = 1 \sim 4$ の節点値を表す。なお、要素節点番号は反時計回りに番号付けを行わないといけないという決まりがある。

節点番号は通し番号がつけられており、この番号は任意である。そのため、節点番号の対応を考えるのは表記の点からも問題となる。そこで、ある e 番目の要素における節点番号と節点値は要素節点番号を使って表現する。例えば、図 3.1 では要素番号 10 の要素の要素節点番号 $1^* \sim 4^*$ はそれぞれ節点番号 12, 17, 18, 13 と対応する。同様にして、節点値 $T_1^e \sim T_4^e$ はそれぞれ $T_{12}, T_{17}, T_{18}, T_{13}$ と対応する。

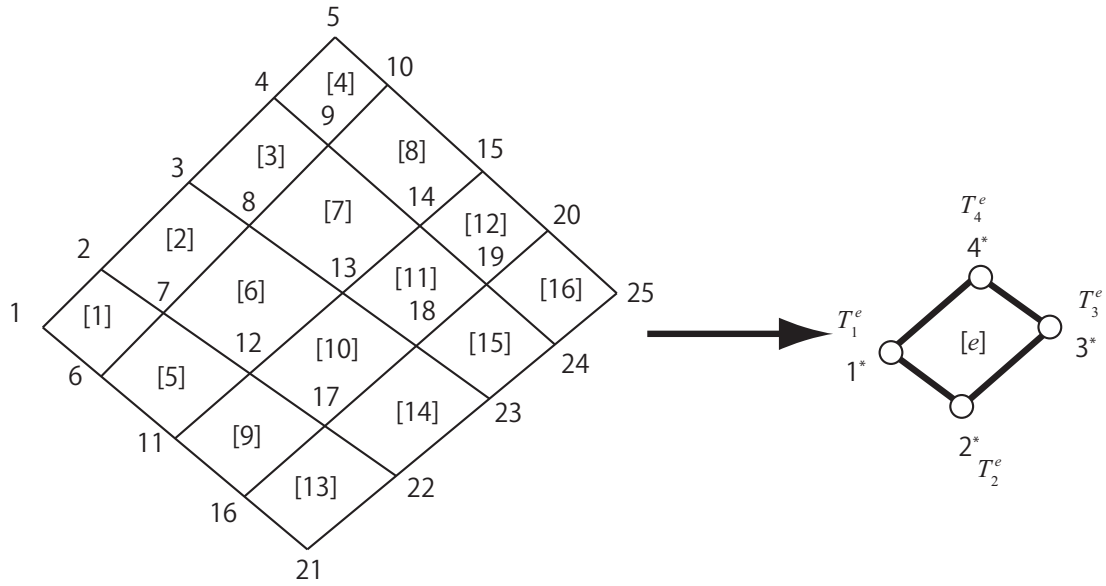


図 3.1: 領域の要素分割図

3.2.5 有限要素方程式 (finite element equation) の導出—四角形要素の適用—

ある要素内での基底関数を内挿関数といい $N_i^e(x, y) = N_i^e$ の記号で表す。内挿関数の性質として一般に対応する節点で 1 の値となり、それ以外での節点では 0 となる。また内挿関数の和は 1 となる。

四角形要素を用いる場合は内挿関数は $N_1^e, N_2^e, N_3^e, N_4^e$ の 4 つである。どの四角形要素のどの辺が領域の境界にきても、境界条件を満たすようにするために、近似関数を以下のように決めることができる。

$$\begin{cases} \hat{T} = \sum_{i=1}^4 N_i^e T_i^e = N_1^e T_1^e + N_2^e T_2^e + N_3^e T_3^e + N_4^e T_4^e \\ \hat{w} = \sum_{i=1}^4 N_i^e w_i^e = N_1^e w_1^e + N_2^e w_2^e + N_3^e w_3^e + N_4^e w_4^e \end{cases} \quad (3.2.26)$$

ここで T_i^e, w_i^e はそれぞれ要素の要素節点番号 $i^*, i^* = 1 \sim 4$ での温度と重みであり定数である。

式 (3.2.26) を式 (3.2.25) に代入すると、以下の式となる。

$$\sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(v_x \frac{\partial N_i^e}{\partial x} + v_y \frac{\partial N_i^e}{\partial y} \right) N_j^e \Omega + k \sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega = k g_2 \sum_{j=1}^4 w_j^e \int_{\Gamma_2} N_j^e d\gamma \quad (3.2.27)$$

この式を行列を使って表す。

$$\mathbf{w}_e^T (\mathbf{A}_e + \mathbf{D}_e) \mathbf{T}_e = \mathbf{w}_e^T \mathbf{F}_e \quad (3.2.28)$$

ここで、両辺に $\mathbf{w}_e^T$ が存在する。 $\mathbf{w}_e^T$ は重み係数のベクトルであり任意の定数である。そのため、これを除いて考えることができる。そうすると、次のような 4 元の連立方程式となることがわかる。

$$(\mathbf{A}_e + \mathbf{D}_e) \mathbf{T}_e = \mathbf{F}_e \quad (3.2.29)$$

式 (3.2.27) や式 (3.2.29) を (ある要素の) 有限要素方程式 (finite element equation) や有限要素式と呼んでいる。 $\mathbf{A}_e, \mathbf{D}_e, \mathbf{F}_e$ はそれぞれ移流項の行列、拡散項の行列、自然境界条件のベクトルである。自然境界条件はフーリエの法則である式 (2.2.1) と同じ形式であることから熱フラックスベクトルともいう。また、拡散項の行列のように二階微分項からなる領域積分項を剛性行列とよぶ慣習がある。それぞれの記号は英語 Advection, Diffusion, Flux のイニシャルを表し、成分は次のように表される。

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} N_1^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) \end{bmatrix} d\Omega \quad (3.2.30)$$

$$\mathbf{D}_e = k \int_{\Omega_e} \begin{bmatrix} \frac{\partial N_1^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_2^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_3^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_4^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_4^e}{\partial y} \end{bmatrix} d\Omega \quad (3.2.31)$$

$$\mathbf{T}_e = \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \\ T_4^e \end{Bmatrix}, \quad \mathbf{w}_e = \begin{Bmatrix} w_1^e \\ w_2^e \\ w_3^e \\ w_4^e \end{Bmatrix} \quad (3.2.32)$$

ここで、 $\mathbf{F}$ は隣接する要素間ではお互いに相殺されるため、要素境界辺での境界積分は、重ね合わせを行った段階で消去され、最終的に領域の境界 Γ のみの積分が残ることとなる。例えば、ある要素 Ω_e の境界 Γ_{12} で自然境界条件が与えられている場合以下ようになる。

$$\mathbf{F}_e = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ N_3^e \\ N_4^e \end{Bmatrix} d\gamma = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ 0 \\ 0 \end{Bmatrix} d\gamma \quad (3.2.33)$$

また、いま扱う問題は表 3.1 より、流速は既知の定数としている。将来的には流体の解析を行うことで得られる流速の値を適用することを想定する。それゆえに、本研究では流速 v_x, v_y は領域内で場所によって値が異なる場合も扱えるようにする。これを実現するために要素内では流速 v_x, v_y も内挿近似する。流速を内挿近似することで以下のような流速の近似式が示される。

$$v_x \approx \hat{v}_x = \sum_{i=1}^4 N_i^e v_{xi}^e = N_1^e v_{x1}^e + N_2^e v_{x2}^e + N_3^e v_{x3}^e + N_4^e v_{x4}^e \quad (3.2.34)$$

$$v_y \approx \hat{v}_y = \sum_{i=1}^4 N_i^e v_{yi}^e = N_1^e v_{y1}^e + N_2^e v_{y2}^e + N_3^e v_{y3}^e + N_4^e v_{y4}^e \quad (3.2.35)$$

ここで、 v_{xi}^e, v_{yi}^e はある要素 e の要素節点番号 i^* での流速である。

式 (3.2.29) を全要素について足し合わせることで、全体の有限要素方程式が以下のように求まる。

$$(\mathbf{A} + \mathbf{D})\mathbf{T} = \mathbf{F} \quad (3.2.36)$$

このように、要素ごとに求めた値の和を取ることで全体の近似方程式を組み立てる方法を直接剛性法という。この式は領域での全節点数を N とすると、 N 元の連立一次方程式となっている。したがって、このように微分方程式が多元の連立一次方程式を解くことに帰着された。

3.2.6 境界条件の組み込み

前節までで、有限要素方程式が導出され、連立方程式を解くことに問題が帰着された。しかし、境界条件をどう組み込むかという問題が生じる。そこで、ここでは基本境界条件と自然境界条件の二種類の境界条件を有限要素方程式に組み込む方法について記述する。

自然境界条件の組み込み

自然境界条件は式 (3.2.2) の第二項

$$\frac{\partial T}{\partial \mathbf{n}} = g_2 \quad (\Gamma_2 \text{ 上}) \quad (3.2.37)$$

で表される境界条件である。自然境界条件は未知関数の一階微分ということで、本来なら処理を行うのは困難である。しかし、有限要素法では弱形式を求めたところで式の中に式 (3.2.21) の右辺のように、この条件が組み込まれている。そのため、自然境界条件は、条件のかかっている節点に対応する右辺にその値を入れて足しこめばよい。

基本境界条件の組み込み

基本境界条件は式 (3.2.2) の第一項

$$T = g_1 \quad (\Gamma_1 \text{ 上}) \quad (3.2.38)$$

で表される境界条件である。この条件を組み込むには、基本境界条件のかかっている節点に対応する係数行列の行の対角成分のみを 1 にして、同じ行の他の列の成分は 0 にする。その後、右辺ベクトルの対応する行に基本境界条件値を足しこめばよい。

一次元要素を使った具体例

これらの二種類の境界条件の組み込みを図 3.2 の簡単な一次元要素を例に示す。

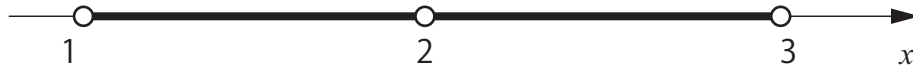


図 3.2: 一次元問題での境界条件

数字は節点番号を示す。この図 3.2 において、 Γ_1, Γ_2 がそれぞれ節点 1, 3 に対応しているとして、次のような境界条件を考える。

$$T = g_1 \quad (\text{節点 1 上}), \quad \frac{\partial T}{\partial x} = g_2 \quad (\text{節点 3 上}) \quad (3.2.39)$$

未知数と係数行列の成分を $T_i, a_{ij} (i, j = 1 \sim 3)$ とすると、境界条件を導入する前の連立方程式は次のようになる。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad (3.2.40)$$

これに、自然境界条件を組み込むと

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ kg_2 \end{Bmatrix} \quad (3.2.41)$$

となる。続いて基本境界条件を組み込む。

$$\begin{bmatrix} 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} g_1 \\ 0 \\ kg_2 \end{Bmatrix} \quad (3.2.42)$$

このようにして境界条件は組み込むことができる。

3.2.7 多元連立方程式の解法—ガウスの消去法 (Gauss elimination)—

3.2.5 項の有限要素方程式によって、多元の連立一次方程式を解くことに問題が帰着された。そこでいま説明を簡略にするために解くべき連立一次方程式を以下のように記す。

$$\mathbf{Ax} = \mathbf{b} \quad (3.2.43)$$

ここで、 $\mathbf{A}$ は $n \times n$ の行列で $\mathbf{x}, \mathbf{b}$ はベクトルで以下のように示される。

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, \quad \mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix} \quad (3.2.44)$$

解くべき式 (3.2.43) は二次元や三次元の場合なら手計算でも未知数 x_n を求めることは容易である。しかし、これが数十、数百次元ともなると手計算で解くのは困難である。そこで、連立一次方程式を組織的に解く方法として、ガウスの消去法 (Gauss elimination) がある。ここではガウスの消去法について説明する。

係数行列 $\mathbf{A}$ と右辺ベクトル $\mathbf{b}$ の成分を

$$a_{ij}^{(1)} = a_{ij}, \quad b_i^{(1)} = b_i \quad (3.2.45)$$

とすると方程式 (3.2.43) は次のようになる。

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)} & \textcircled{1} \\ a_{21}^{(1)}x_1 + a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} & \textcircled{2} \\ \vdots & \vdots \\ a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} & \textcircled{n} \end{cases} \quad (3.2.46)$$

上付添え字 (1) はこれから第 1 段目の消去が行われることを示す。消去の第 1 段では②以下の式から x_1 を含む項を消去する。

消去の前にまず、①の両辺を対角成分（ピボット）である $a_{11}^{(1)}$ で割る。すると次のようになる。

$$\begin{cases} x_1 + \frac{a_{12}^{(1)}}{a_{11}^{(1)}}x_2 + \cdots + \frac{a_{1n}^{(1)}}{a_{11}^{(1)}}x_n = \frac{b_1^{(1)}}{a_{11}^{(1)}} & \textcircled{1}' \\ a_{21}^{(1)}x_1 + a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} & \textcircled{2} \\ \vdots & \vdots \\ a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} & \textcircled{n} \end{cases} \quad (3.2.47)$$

続いて、①' に①' の対角成分の下項である②の $a_{21}^{(1)}$ 乗じて②から引く。

$$\begin{cases} x_1 + \frac{a_{12}^{(1)}}{a_{11}^{(1)}}x_2 + \cdots + \frac{a_{1n}^{(1)}}{a_{11}^{(1)}}x_n = \frac{b_1^{(1)}}{a_{11}^{(1)}} & \textcircled{1}' \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} & \textcircled{2}' \\ \vdots & \vdots \\ a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} & \textcircled{n} \end{cases} \quad (3.2.48)$$

同様に、①から下の成分をかけて引く操作を繰り返し、一番下の項である n 行目まで行くと次のようになる。

$$\begin{cases} x_1 + \frac{a_{12}^{(1)}}{a_{11}^{(1)}}x_2 + \cdots + \frac{a_{1n}^{(1)}}{a_{11}^{(1)}}x_n = \frac{b_1^{(1)}}{a_{11}^{(1)}} & \textcircled{1}' \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} & \textcircled{2}' \\ \vdots & \vdots \\ a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n = b_n^{(2)} & \textcircled{n}' \end{cases} \quad (3.2.49)$$

ここで、 $a_{ij}^{(2)}, b_i^{(2)}$ は次のように定義される。

$$\begin{cases} a_{ij}^{(2)} = a_{ij}^{(1)} - a_{i1}^{(1)} \frac{a_{1j}^{(1)}}{a_{11}^{(1)}}, \quad i, j = 2, 3, \dots, n \\ b_i^{(2)} = b_i^{(1)} - a_{i1}^{(1)} \frac{b_1^{(1)}}{a_{11}^{(1)}}, \quad i = 2, 3, \dots, n \end{cases} \quad (3.2.50)$$

次に消去の第 2 段では式 (3.2.49) の②' より下の式から x_2 の項を消去する。このための手順は第 1 段目の消去と全く同じである。最終的にこの操作を x_{n-1} の項が消えるまで行うことで、次のような左下の成分が 0 の行列の式が得られる。

$$\begin{bmatrix} 1 & \frac{a_{12}^{(1)}}{a_{11}^{(1)}} & \cdots & \frac{a_{1n}^{(1)}}{a_{11}^{(1)}} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ 0 & & & a_{nn}^{(n)} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} \frac{b_1^{(1)}}{a_{11}^{(1)}} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{Bmatrix} \quad (3.2.51)$$

ここまでの操作をガウスの消去法における**前進消去 (forward elimination)** という。前進消去を行うことで、 x_n は容易に求まり、

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}} \quad (3.2.52)$$

と求まる。 x_{n-1} は x_n の値を使って

$$x_{n-1} = \frac{b_{n-1}^{(n-1)} - a_{n-1n}^{(n-1)} x_n}{a_{n-1n-1}^{(n-1)}} \quad (3.2.53)$$

というように求まる。同様に、 $x_1 \sim x_n$ は求まる。 x_n から x_{n-1} を求めることからこの操作を**後退代入 (backward substitution)** という。

このように、ガウスの消去法を使って組織的に連立一次方程式を解くことができる。この項を書くにあたって、文献 [8, pp.52-55] を参考にした。

3.3 座標変換・内挿関数の設定・積分の計算

図 3.1 に示したように要素分割を行う際、すべての要素が同じ形状とは限らない。そのため、内挿関数を決める際には要素ごとに関数を設定する必要がある。しかし、要素ごとに内挿関数を設定するのは煩雑で計算機で計算を行う際でも非効率的である。そこで、内挿関数は正規化された局所座標系に対して設定し、その後座標変換によって個別の要素に対応する。

3.3.1 局所座標系の設定と要素の次数

正規化された局所座標系として、図 3.3 に示す四角形（正方形）一次要素を用いた。図の * のついた数字は要素節点番号を表す。

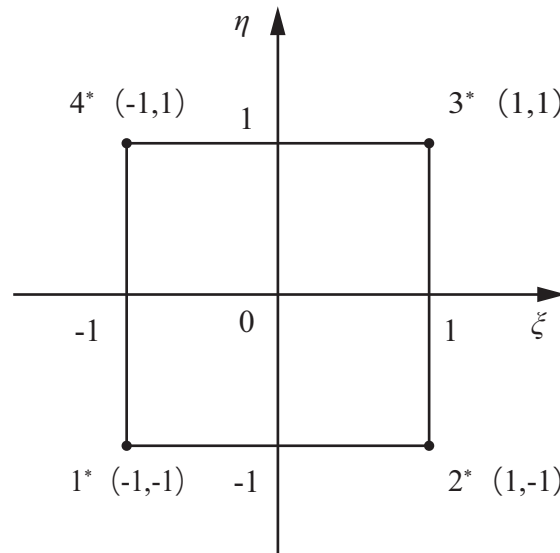


図 3.3: 想定する局所座標系

内挿関数はある要素について設定された基底関数である。基底関数は境界上では境界条件を満たすように設定されなければならない。また、どの要素のどの辺が境界になるかは任意である。したがって、正規化された要素のすべての辺が節点での座標では 1、それ以外では 0 となるように関数を設定しなくてはならない。なぜならば、境界条件によって定めた境界上の値はその値となる必要があるからである。

内挿関数は要素の境界上では連続性を保つために、境界上で任意の次数の ξ, η の関数である必要がある。

ここで、要素の節点間をどのように近似するかで要素の種類が異なる。要素を一次式で表す場合を一次要素という。この場合、要素内は節点間で直線近似することになる。同様にして、要素を二次式で表す場合を二次要素という。高次の式で表す場合は、式の係数も増えるため要素の節点も多くなり、解析の精度も向上する。一方で計算も複雑になり、計算時間もその分かかることが想定される。本研究では解析のプログラムを作成することに主眼を置いており、計算の簡単な一次要素を用いる。

これらのことを踏まえて、内挿関数には次に示す一次補間式を用いた。

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i), \quad i = 1 \sim 4 \quad (3.3.1)$$

ここで、 N_i^e は要素節点番号節点 i^* での内挿関数であり、 ξ_i, η_i は要素節点番号 i^* での座標値を表す。これを参考に各節点での内挿関数を求めると以下ようになる。

$$\begin{cases} N_1(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_1)(1 + \eta\eta_1) = \frac{1}{4}(1 - \xi)(1 - \eta) \\ N_2(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_2)(1 + \eta\eta_2) = \frac{1}{4}(1 + \xi)(1 - \eta) \\ N_3(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_3)(1 + \eta\eta_3) = \frac{1}{4}(1 + \xi)(1 + \eta) \\ N_4(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_4)(1 + \eta\eta_4) = \frac{1}{4}(1 - \xi)(1 + \eta) \end{cases} \quad (3.3.2)$$

3.3.2 座標変換とアイソパラメトリック要素

前項で考えた局所座標系 ξ, η を全体座標系 x, y に座標変換することを考える。ここで、本研究ではアイソパラメトリック要素の適用を考えた。

アイソパラメトリック要素 (isoparametric element) とは、近似関数を近似するときに使った内挿関数と同じ関数を座標変換に用いた要素のことである。アイソパラメトリック要素を用いることで、近似関数を決めた時のように、全体座標 x, y もある要素 e の要素節点番号 i^* での座標値 x_i^e, y_i^e と内挿関数をかけたもので表すことができる。そのため、全体座標系は局所座標系を用いて以下のように変換できる。

$$x = \sum_{i=1}^4 N_i(\xi, \eta) x_i^e, \quad y = \sum_{i=1}^4 N_i(\xi, \eta) y_i^e \quad (3.3.3)$$

実際の有限要素法の解析には全体座標系の内挿関数の微分が必要となる。これらを定義より容易に求められる局所座標系の微分を用いて求める。

局所座標系の内挿関数の微分は偏微分の公式 (合成関数の微分) より以下ようになる。

$$\frac{\partial N_i(\xi, \eta)}{\partial \xi} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi}, \quad \frac{\partial N_i(\xi, \eta)}{\partial \eta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta}$$

これをマトリックス表示すると以下ようになる。

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} \quad (3.3.4)$$

ここで $[J]$ をヤコビの行列 (Jacobian matrix) と呼ぶ。各成分はそれぞれ、式 (3.3.3), 式 (3.3.3) を用い計算を行うと以下のような 2 行 2 列の行列になる (x_i, y_i は座標値のため微分対象外)。

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1^e & y_1^e \\ x_2^e & y_2^e \\ x_3^e & y_3^e \\ x_4^e & y_4^e \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

式 (3.3.4) を形式的に解くと (全体座標系の微分を局所座標系の微分で表すと) 以下ようになる。

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \quad (3.3.5)$$

ここで、余因子定理を用いると $[J]^{-1}$ は以下ようになる。

$$[J]^{-1} = \frac{1}{\det[J]} \begin{bmatrix} c_{22} & -c_{12} \\ -c_{21} & c_{11} \end{bmatrix} = \frac{1}{c_{11}c_{22} - c_{12}c_{21}} \begin{bmatrix} c_{22} & -c_{12} \\ -c_{21} & c_{11} \end{bmatrix} \quad (3.3.6)$$

また、局所座標系における内挿関数の偏微分は以下のようにになる (座標系は図 3.3 に従う)。

$$\frac{\partial N_1}{\partial \xi} = \frac{1}{4}(-1 + \eta), \quad \frac{\partial N_2}{\partial \xi} = \frac{1}{4}(1 - \eta), \quad \frac{\partial N_3}{\partial \xi} = \frac{1}{4}(1 + \eta), \quad \frac{\partial N_4}{\partial \xi} = \frac{1}{4}(-1 - \eta) \quad (3.3.7)$$

$$\frac{\partial N_1}{\partial \eta} = \frac{1}{4}(-1 + \xi), \quad \frac{\partial N_2}{\partial \eta} = \frac{1}{4}(-1 - \xi), \quad \frac{\partial N_3}{\partial \eta} = \frac{1}{4}(1 + \xi), \quad \frac{\partial N_4}{\partial \eta} = \frac{1}{4}(1 - \xi) \quad (3.3.8)$$

3.3.3 ガウスの数値積分

重み付き残差法によって、領域全体で誤差の和を 0 にするという考えにしたがって、積分の計算が登場してきた。しかし、いま積分したい関数は要素ごとに異なる内挿関数で表される。そのため、ひとつひとつの要素について関数の形を確認して定義に則って積分した関数を求めていくのは効率が悪く、場合によっては積分した関数を求められないことも考えられる。そこで数値積分を行う。

数値積分は積分したい関数に対し、決められた座標での関数値に重みをかけることで積分を近似する手法である。有限要素法ではしばしばガウス・ルジャンドル法 (ガウスの数値積分法) が用いられている。本研究でもこの方法を採用する。

ガウス・ルジャンドル法だけでは $-1 \sim +1$ までの範囲の積分範囲しか計算が行えない。しかし、前項で述べた $-1 \sim +1$ の範囲の局所座標系を想定し、座標変換を行うため適用可能である。

二次元で四角形領域を各方向について二点で積分する場合は図 3.4 に示される座標点での関数値を用いる。またこの場合の重みはすべて 1.0 である。図では $0.577 \dots$ と書いたが実際には 0.577350269189626 の値を使っている。ス

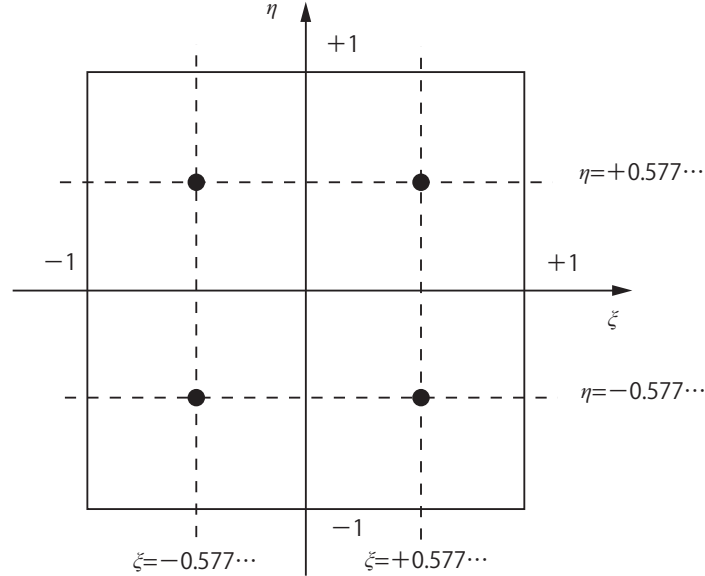


図 3.4: ガウス・ルジャンドル法での積分点での座標

ペースの都合上このような記号を使って省略した。この問題での積分式は関数 $f(\xi, \eta)$ と重み関数 w を使って、次のように表される。

$$\begin{aligned} \int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta) d\xi d\eta &= \sum_{i=1}^2 \sum_{j=1}^2 f(\xi_i, \eta_j) w_i w_j \\ &= f(-0.577 \dots, -0.577 \dots) \times 1.0 \times 1.0 + f(-0.577 \dots, +0.577 \dots) \times 1.0 \times 1.0 \\ &\quad + f(+0.577 \dots, -0.577 \dots) \times 1.0 \times 1.0 + f(+0.577 \dots, +0.577 \dots) \times 1.0 \times 1.0 \end{aligned} \quad (3.3.9)$$

これを正規化された局所座標系で適用することで、積分の計算を行う。

第4章 安定化有限要素法の適用

式 (4.0.1) に示される一般的な移流拡散問題を解く際に移流の影響が卓越する場合において、解の不安定性が問題となる。ここでは、そのことについて述べこの問題を解決する方法について記述する。

$$\begin{cases} \frac{\partial T}{\partial t} + \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) = f & (\Omega \text{上}) \\ T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial \mathbf{n}} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (4.0.1)$$

$$\begin{cases} T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial \mathbf{n}} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (4.0.2)$$

なお、本来であれば式 (4.0.1) の右辺は $f/(\rho c_p)$ であるが、この章に限り表記をきれいにするために右辺を f に置き直した。

4.1 解の不安定さと Peclet 数——一次元定常移流拡散問題の厳密解の例——

具体的に比較的簡単に理論解を導出できる、以下で示される境界条件を持つ一次元定常移流拡散問題について考える。

$$\begin{cases} v_x \frac{dT}{dx} - k \frac{d^2 T}{dx^2} = 0 & (0 < x < L) \\ T(0) = 1, (L) = 0 \end{cases} \quad (4.1.1)$$

$$\begin{cases} T(0) = 1, (L) = 0 \end{cases} \quad (4.1.2)$$

この問題の理論解を考える。この問題は二階線形常微分方程式である。よって、

$$T = e^{ax} a \text{ は未知の定数} \quad (4.1.3)$$

と置いて特性方程式を解く。

$$\begin{aligned} v_x a e^{ax} - a^2 k e^{ax} &= 0 \\ a e^{ax} (v_x - ak) &= 0 \end{aligned} \quad (4.1.4)$$

ここで、 $e^{ax} \neq 0$ より、この式を満たすのは

$$a = 0, = \frac{v_x}{k} \quad (4.1.5)$$

のときである。よって、これらの値を式 (4.1.3) に代入するとそれぞれ

$$T = 1, = e^{\frac{v_x}{k} x} \quad (4.1.6)$$

となる。解が複数求まったときは重ね合わせの原理より、それぞれ任意定数倍して足したものが求める式の解である。よって、この問題の一般解は

$$T = C_1 + C_2 e^{\frac{v_x}{k} x} \quad (C_1, C_2 \text{ は任意定数}) \quad (4.1.7)$$

となる。

この一般解に式 (4.1.2) の境界条件を当てはめることでこの問題の特殊解を求める。

$$\begin{cases} T(0) = C_1 + C_2 = 1 \\ T(L) = C_1 + C_2 e^{\frac{v_x}{k} L} \end{cases} \quad (4.1.8)$$

$$\begin{cases} T(L) = C_1 + C_2 e^{\frac{v_x}{k} L} \end{cases} \quad (4.1.9)$$

この連立方程式を解く。式 (4.1.8) 及び式 (4.1.9) より

$$C_2 \left(1 - e^{\frac{v_x}{k} L} \right) = 1, \quad C_2 = \frac{1}{1 - e^{\frac{v_x}{k} L}} \quad (4.1.10)$$

である。これらを式 (4.1.8) に代入して

$$C_1 = 1 - \frac{1}{1 - e^{\frac{v_x}{k}L}} = \frac{1 - e^{\frac{v_x}{k}L} - 1}{1 - e^{\frac{v_x}{k}L}} = \frac{-e^{\frac{v_x}{k}L}}{1 - e^{\frac{v_x}{k}L}} \quad (4.1.11)$$

となる。これらの C_1, C_2 の値を式 (4.1.7) に代入すると以下の厳密解が求まる。

$$\begin{aligned} T &= \frac{-e^{\frac{v_x}{k}L}}{1 - e^{\frac{v_x}{k}L}} + \frac{e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = \frac{1}{1 - e^{\frac{v_x}{k}L}} + \frac{-e^{\frac{v_x}{k}L}}{1 - e^{\frac{v_x}{k}L}} - \frac{1}{1 - e^{\frac{v_x}{k}L}} + \frac{e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} \\ &= \frac{-e^{\frac{v_x}{k}L}}{1 - e^{\frac{v_x}{k}L}} + \frac{e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = \frac{-e^{\frac{v_x}{k}L} + e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = \frac{1 - e^{\frac{v_x}{k}L} - 1 + e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = 1 + \frac{-1 + e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} \\ T &= 1 - \frac{1 - e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = 1 - \frac{e^{\frac{v_x}{k}x} - 1}{e^{\frac{v_x}{k}L} - 1} \end{aligned} \quad (4.1.12)$$

この厳密解を $v_x = 1, L = 10$ として、熱拡散率 k 変化させた分布を図 4.1 に示した。

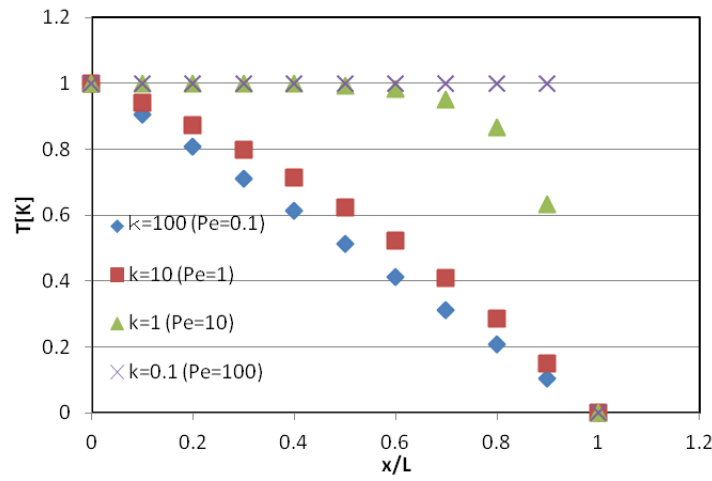


図 4.1: 熱拡散率 k に対する移流拡散方程式の厳密解の分布

この図 4.1 からわかるように、熱拡散率が小さいと境界付近で急激に値が変化する。このように、移流拡散現象は移流項と拡散項の比として表される無次元量である **Peclet 数** (Peclet number)

$$Pe = \frac{|v_x|L}{k} \quad (4.1.13)$$

に左右される。

Peclet 数が大きい場合は移流が支配的で、小さいときは拡散が支配的となることが図 4.1 からわかる。

厳密解は式 (4.1.12) からわかるように e^{Pe} という形の項が入っている。そのため、Peclet 数が大きくなると値が膨大になり計算が難しくなる。(図 4.1 での計算において表計算ソフトのエクセルでは $Pe=720$ 付近から計算エラーが発生した。)

このように、Peclet 数が大きくなると数値的な不安定性を引き起こし正しい計算結果が得られない。これを回避するために、有限要素法では流れ方向に重みを変化させることで、人工的な拡散を加えて (Peclet 数を下げて) 安定化を図る Upwind Galerkin 法が考案され、さらに流れの流線方向のみに適切な量の人工拡散を加える SUPG(Streamline Upwind/Petrov-Galerkin) 法に基づく安定化有限要素法へと発展していった。

安定化の手法については文献 [4] 『続・有限要素法による流れのシミュレーション』について詳しい記述がある。以下ではこの文献を参考に安定化の理論について述べていく。

4.2 Galerkin 法と Upwind Galerkin 法

前節で扱った一次元問題を使って Upwind Galerkin 法について確認する。そのために、まず通常の Galerkin 法を適用して、その後 Upwind Galerkin 法について取り上げ違いを見ていく。

重み付き残差法と弱形式 まず、式 (4.1.1) に対して重み付き残差法を適用し、さらに弱形式を求める。

重み関数を w として、これを両辺に掛けて領域全体で積分を行う。すると、重み付き残差方程式は次のようになる。

$$\int_0^L w v_x \frac{dT}{dx} dx - k \int_0^L w \frac{d^2 T}{dx^2} dx = 0 \quad (4.2.1)$$

部分積分を施すと拡散項は以下のようになる。

$$\int_0^L w \frac{d^2 T}{dx^2} dx = \left[w \frac{dT}{dx} \right]_0^L - \int_0^L \frac{dw}{dx} \frac{dT}{dx} dx \quad (4.2.2)$$

したがって、式は次のようになる。

$$\int_0^L w v_x \frac{dT}{dx} dx - k \left[w \frac{dT}{dx} \right]_0^L + k \int_0^L \frac{dw}{dx} \frac{dT}{dx} dx = 0 \quad (4.2.3)$$

ここで、境界条件を満たすように境界で誤差が生じない内部法を採用するので、重み関数 w は境界上では 0 となる。したがって、式は次のようになる。

$$\int_0^L w v_x \frac{dT}{dx} dx + k \int_0^L \frac{dw}{dx} \frac{dT}{dx} dx = 0 \quad (4.2.4)$$

このようにして、弱形式は求まった。

Galerkin 法の適用と内挿関数の設定及び有限要素方程式の導出 続いて、この弱形式に対して Galerkin 法を適用して有限要素方程式を導く。まず、領域を n 個の要素に分割する。Galerkin 法の特徴は近似関数と同じ基底関数を重み関数にも適用することである。

いま一次元問題であり、領域分割を行うと図 4.2 に示す線要素となる。

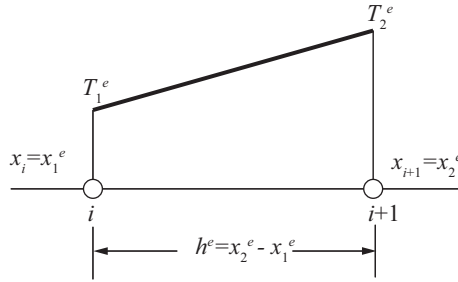


図 4.2: 2 節点一次線要素

要素内を一次近似する一次要素を採用するため、形状関数は 2 つとなる。よって、近似関数と重み関数は要素内で内挿関数 N_1^e, N_2^e を使って次のように表される。

$$\hat{T} = N_1^e T_1^e + N_2^e T_2^e = \sum_{i=1}^2 N_i^e T_i^e = N_i^e T_i^e, \quad \hat{w} = N_1^e w_1^e + N_2^e w_2^e = \sum_{i=1}^2 N_i^e w_i^e = N_i^e w_i^e \quad (4.2.5)$$

ここで、式の最後で $\sum$ が省略されているが、これは総和規約による表記法であり、同一項において 2 つの同じ添え字が繰り返して現れた場合は、総和規約の表記法を用いる。 T_1^e, T_2^e はある要素での節点での温度を示し、 w_1^e, w_2^e はある要素の節点での重み（任意定数）である。要素 e 内での内挿関数 N_1^e, N_2^e は要素が線要素であり一次近似するため、次のような形状関数となる。

$$N_1^e = \frac{x_2^e - x}{h_e}, \quad N_2^e = \frac{x - x_1^e}{h_e} \quad (4.2.6)$$

これらの近似関数 $\hat{T}, \hat{w}$ を弱形式に代入して有限要素式を求める。

$$\begin{aligned} \int_{x_1^e}^{x_2^e} \sum_{j=1}^2 N_j^e w_j^e v_x \frac{d \sum_{i=1}^2 N_i^e T_i^e}{dx} dx + k \int_{x_1^e}^{x_2^e} \frac{d \sum_{j=1}^2 N_j^e w_j^e}{dx} \frac{d \sum_{i=1}^2 N_i^e T_i^e}{dx} dx &= 0 \\ \sum_{j=1}^2 \sum_{i=1}^2 w_j^e T_i^e \int_{x_1^e}^{x_2^e} N_j^e v_x \frac{d N_i^e}{dx} dx + k \sum_{j=1}^2 \sum_{i=1}^2 w_j^e T_i^e \int_{x_1^e}^{x_2^e} \frac{d N_j^e}{dx} \frac{d N_i^e}{dx} dx &= 0 \end{aligned} \quad (4.2.7)$$

これを行列表示すると以下となる。

$$\mathbf{w}_e^T (\mathbf{A}_e + \mathbf{D}_e) \mathbf{T}_e = \mathbf{0} \quad (4.2.8)$$

なおここで、 $\mathbf{w}_e, \mathbf{A}_e, \mathbf{D}_e, \mathbf{T}_e$ の成分は以下のように示される。

$$\mathbf{T}_e = \begin{Bmatrix} T_1^e \\ T_2^e \end{Bmatrix}, \quad \mathbf{w}_e = \begin{Bmatrix} w_1^e \\ w_2^e \end{Bmatrix} \quad (4.2.9)$$

$$\mathbf{A}_e = \begin{bmatrix} \int_{x_1^e}^{x_2^e} N_1^e v_x \frac{dN_1^e}{dx} dx & \int_{x_1^e}^{x_2^e} N_2^e v_x \frac{dN_1^e}{dx} dx \\ \int_{x_1^e}^{x_2^e} N_1^e v_x \frac{dN_2^e}{dx} dx & \int_{x_1^e}^{x_2^e} N_2^e v_x \frac{dN_2^e}{dx} dx \end{bmatrix}, \quad \mathbf{D}_e = k \begin{bmatrix} \int_{x_1^e}^{x_2^e} \frac{dN_1^e}{dx} \frac{dN_1^e}{dx} dx & \int_{x_1^e}^{x_2^e} \frac{dN_2^e}{dx} \frac{dN_1^e}{dx} dx \\ \int_{x_1^e}^{x_2^e} \frac{dN_1^e}{dx} \frac{dN_2^e}{dx} dx & \int_{x_1^e}^{x_2^e} \frac{dN_2^e}{dx} \frac{dN_2^e}{dx} dx \end{bmatrix} \quad (4.2.10)$$

w_1^e, w_2^e は任意の定数であるので、 $\mathbf{w}_e$ は両辺で消去して考えることができ次の式となる。

$$(\mathbf{A}_e + \mathbf{D}_e) \mathbf{T}_e = \mathbf{0} \quad (4.2.11)$$

この $\mathbf{A}_e, \mathbf{D}_e$ の各成分は計算することができる。例として、各第 11 成分の計算を行う。
まず、内挿関数の微分は次のようになる。

$$\frac{dN_1^e}{dx} = -\frac{1}{h_e}, \quad \frac{dN_2^e}{dx} = \frac{1}{h_e} \quad (4.2.12)$$

したがって $\mathbf{A}_e$ の 11 成分は以下となる。

$$\begin{aligned} \int_{x_1^e}^{x_2^e} N_1^e v_x \frac{dN_1^e}{dx} dx &= \int_{x_1^e}^{x_2^e} \frac{x_2 - x}{h_e} \left(-\frac{1}{h_e} \right) dx = -\frac{1}{h_e^2} \left[x_2 x - \frac{x^2}{2} \right]_{x_1^e}^{x_2^e} = -\frac{1}{h_e^2} \left[\left(x_2 x_2 - \frac{x_2^2}{2} \right) - \left(x_2 x_1 - \frac{x_1^2}{2} \right) \right] \\ &= -\frac{1}{h_e^2} \left[\frac{x_2^2}{2} - x_2 x_1 + \frac{x_1^2}{2} \right] = -\frac{1}{h_e^2} \frac{(x_2 - x_1)^2}{2} = -\frac{1}{2} \end{aligned} \quad (4.2.13)$$

また、 $\mathbf{D}_e$ の 11 成分も同様に計算ができ以下のようになる。

$$\int_{x_1^e}^{x_2^e} \frac{dN_1^e}{dx} \frac{dN_1^e}{dx} dx = \int_{x_1^e}^{x_2^e} \left(-\frac{1}{h_e} \right) \left(\frac{1}{h_e} \right) dx = \frac{1}{h_e^2} [x]_{x_1^e}^{x_2^e} = \frac{1}{h_e^2} [x_2^e - x_1^e] = \frac{1}{h_e} \quad (4.2.14)$$

他の第 12, 21, 22 成分についても同様な計算を行うことで $\mathbf{A}_e, \mathbf{D}_e$ は次のようになる。

$$\mathbf{A}_e = v_x \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \mathbf{D}_e = k \begin{bmatrix} \frac{1}{h_e} & -\frac{1}{h_e} \\ -\frac{1}{h_e} & \frac{1}{h_e} \end{bmatrix} \quad (4.2.15)$$

式 (4.2.11) を全要素について足し合わせることで、領域全体の有限要素方程式が次のように得られる。

$$(\mathbf{A} + \mathbf{D}) \mathbf{T} = \mathbf{0} \quad (4.2.16)$$

この有限要素方程式から節点 i に関する部分を取り出す。 i に関する部分は、先ほどの問題で考えた要素の節点が i と $i+1$ から成る要素と $i-1$ と i からなる要素についてである。計算を簡単にするために、要素の幅が全て等しく h であるものを考える。

ここで、 $i-1$ と i からなる要素の有限要素方程式は以下のようになる。

$$\left(v_x \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + \frac{k}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right) \begin{Bmatrix} T_{i-1} \\ T_i \end{Bmatrix} = \begin{Bmatrix} v_x \left(-\frac{1}{2} T_{i-1} + \frac{1}{2} T_i \right) + k \left(\frac{T_{i-1}}{h} - \frac{T_i}{h} \right) \\ v_x \left(-\frac{1}{2} T_{i-1} + \frac{1}{2} T_i \right) + k \left(-\frac{T_{i-1}}{h} + \frac{T_i}{h} \right) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.2.17)$$

また、 i と $i+1$ からなる要素の有限要素式は以下のようになる。

$$\left(v_x \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + \frac{k}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right) \begin{Bmatrix} T_i \\ T_{i+1} \end{Bmatrix} = \begin{Bmatrix} v_x \left(-\frac{1}{2}T_i + \frac{1}{2}T_{i+1} \right) + k \left(\frac{T_i}{h} - \frac{T_{i+1}}{h} \right) \\ v_x \left(-\frac{1}{2}T_i + \frac{1}{2}T_{i+1} \right) + k \left(-\frac{T_i}{h} + \frac{T_{i+1}}{h} \right) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.2.18)$$

領域全体の有限要素式で節点 i に関する式は、式 (4.2.17) の 2 行目と式 (4.2.18) の一行目を足し合わせた式である。これらの式を足し合わせたものは以下のようになる。

$$v_x \frac{T_{i+1} - T_{i-1}}{2} + k \frac{-T_{i+1} + 2T_i - T_{i-1}}{h} = 0 \quad (4.2.19)$$

この式の両辺を h で割り符号を整理すると次のようになる。

$$v_x \frac{T_{i+1} - T_{i-1}}{2h} - k \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = 0 \quad (4.2.20)$$

この式は中心差分による差分法で表される式と同一である。

風上差分法との類似性 文献 [4, p.45] 『続・有限要素法による流れのシミュレーション』では、この問題を実際に数値解析で解いた結果が示されている。このとき、Peclet 数

$$P_e = \frac{|v_x|h}{2k} \quad (4.2.21)$$

が大きい ($P_e > 1$) ときに数値振動が発生している。Peclet 数の内、流速 v_x や温度伝導率 k は解析対象のデータとして与えられる。解析者が値を決めることができるのは、要素幅 h である。よって、数値振動を回避するには Peclet 数が小さくなる ($P_e \leq 1$) 要素幅 h になるように要素分割を行う必要がある。

しかし、実際には計算機の性能や解析時間など解析者側の都合で要素分割を小さくできない場合がありこのままでは不都合である。この問題を解決するために、当初有限差分法で移流項に対して後退差分することに注目がなされた。

有限差分法とは、微分方程式の解を微分の定義に基づいて解を求める方法である。簡単な例として一階微分方程式を考える。

$$u'(x) = 3u(x) + 2 \quad (4.2.22)$$

これを解くには差分商

$$\frac{u(x+h) - u(x)}{h} \approx u'(x) \quad (h \rightarrow 0) \quad (4.2.23)$$

を用いて

$$u(x+h) = u(x) + h(3u(x) + 2) \quad (4.2.24)$$

と近似する。このようにして差分を使って微分方程式を近似することを差分法という。(例に挙げたような差分のしかたをとくにオイラー法と呼ばれている。)

式 (4.1.1) の移流項に対して後退差分、拡散項に対して中心差分を用いた場合の離散化方程式は以下のようになる。

$$v_x \frac{T_i - T_{i-1}}{h} - k \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = 0 \quad (4.2.25)$$

この近似手法は一次精度風上差分法 (upwind difference method) と呼ばれている。この式は Galerkin 法での離散化方程式 (4.2.20) と形が似ている。実際に Galerkin 法で離散化された式 (4.2.20) は風上差分法で離散化された式 (4.2.25) に k に $k_a = \frac{|v_x|h}{2}$ だけ余分に加えたものと等しい。よって、式 (4.2.25) は次のように変形できる。

$$v_x \frac{T_{i+1} - T_{i-1}}{2h} - (k + k_a) \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = 0 \quad (4.2.26)$$

ここで $k_a = \frac{|v_x|h}{2}$ は人工拡散 (artificial diffusion) 係数である。中心差分と Galerkin 法は等しいので、有限要素法では人工拡散係数があらかじめ加えられた次式

$$v_x \frac{\partial T}{\partial x} - (k + k_a) \frac{\partial^2 T}{\partial x^2} = f \quad \text{in } \Omega \quad (4.2.27)$$

に対して、Galerkin 法が適用された。(ここで f は発生項である。) つまり、

$$\int_0^L \hat{w} v_x \frac{d\hat{T}}{dx} dx + \int_0^L (k + k_a) \frac{d\hat{w}}{dx} \frac{d\hat{T}}{dx} dx = 0 \quad (4.2.28)$$

となる。式 (4.2.28) の離散化手法は **Upwind Galerkin 法** と呼ばれている。しかし、この方法による過剰な人工拡散が入ることによる「なまった」解を与えることが判明した。

そこで、 k_a の代わりに安定性を損なわない必要最小限の人工拡散係数 $k_0 = k_a \xi$ が導入された。ここで、 ξ は最適な人工拡散係数を与えるパラメータであり、次の離散化方程式

$$v_x \frac{T_{i+1} - T_{i-1}}{2h} - (k + k_0) \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = 0, \quad k_0 = \frac{|v_x|h}{2} \xi \quad (4.2.29)$$

の解が移流拡散方程式の解と一致するように以下のように示された。

$$k_0 = \frac{|v_x|h}{2} \xi(P_e) = \frac{|v_x|h}{2} \left(\coth P_e - \frac{1}{P_e} \right), \quad P_e = \frac{|v_x|h}{2k} \quad (4.2.30)$$

ここで、 $\coth x$ は以下で示される関数である。

$$\coth x = \frac{1}{\tanh x} = \frac{\cosh x}{\sinh x} = \frac{\frac{e^x - e^{-x}}{2}}{\frac{e^x + e^{-x}}{2}} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.2.31)$$

4.3 Upwind Petrov-Galerkin 法

Upwind Galerkin 法による離散化方程式 (4.2.28) に最適な人工拡散係数 k_0 をあてはめると次式が得られる。

$$\int_0^L \hat{w} v_x \frac{d\hat{T}}{dx} dx + \int_0^L (k + k_0) \frac{d\hat{w}}{dx} \frac{d\hat{T}}{dx} dx = 0 \quad (4.3.1)$$

そして、上式を変形すると次のようになる。

$$\int_0^L \left(\hat{w} + \frac{k_0}{v_x} \frac{d\hat{w}}{dx} \right) v_x \frac{d\hat{T}}{dx} dx + \int_0^L k \frac{d\hat{w}}{dx} \frac{d\hat{T}}{dx} dx = 0 \quad (4.3.2)$$

このことから、Upwind Galerkin 法は移流項に対する重み関数 $\hat{w}$ の代わりに $\hat{T}$ と異なる近似形の重み関数

$$\tilde{w} = \hat{w} + \frac{k_0}{v_x} \frac{d\hat{w}}{dx} \quad (4.3.3)$$

を用いた重み付き残差方程式になっていると解釈できる。

しかし、この方法は支配方程式中に発生項 f が含まれる場合には離散化式

$$\int_0^L \tilde{w} v_x \frac{d\hat{T}}{dx} dx + \int_0^L k \frac{d\tilde{w}}{dx} \frac{d\hat{T}}{dx} dx - \int_0^L \tilde{w} f dx = 0 \quad (4.3.4)$$

は精度が悪いことが指摘された。この原因は移流項と他の光とで異なる重み関数を用いていることにあり、全ての項に $\tilde{w}$ を用いることが提案された。つまり、次のような式である。

$$\int_0^L \tilde{w} v_x \frac{d\hat{T}}{dx} dx + \int_0^L k \frac{d\tilde{w}}{dx} \frac{d\hat{T}}{dx} dx - \int_0^L \tilde{w} f dx = 0 \quad (4.3.5)$$

ここで、

$$\tilde{w} = \hat{w} + \frac{k_0}{v_x} \frac{d\hat{w}}{dx} = \hat{w} + \tau_e v_x \frac{d\hat{w}}{dx} \quad (4.3.6)$$

である。要素ごとに異なるパラメータをもつ τ_e は安定化パラメータ (stabilization parameter) と呼ばれ、次式で与えられる。

$$\tau_e = \frac{h_e}{2|v_x|} \xi(P_e), \quad \xi(P_e) = \coth P_e - \frac{1}{P_e}, \quad P_e = \frac{|v_x|h_e}{2k} \quad (4.3.7)$$

ここに v_x は各要素における流速、 h_e は要素サイズ、 P_e は Peclet 数である。

式 (4.3.5) は Galerkin 法の重み関数 $\hat{w}$ の代わりに近似関数 $\hat{T}$ と異なる近似関数形の重み関数 $\tilde{w}$ を全ての項に用いた重み付き残差式になっている。このように、近似関数と異なる基底関数を重み関数に用いる方法を **Petrov-Galerkin** 法という。

式 (4.3.5) に式 (4.3.6) を代入すると次式になる。

$$\int_0^L \hat{w} v_x \frac{d\hat{T}}{dx} dx + \int_0^L k \frac{d\hat{w}}{dx} \frac{d\hat{T}}{dx} dx - \int_0^L \hat{w} f dx + \sum_{e=1}^n \int_{x_1^e}^{x_2^e} \left(v_x \frac{d\hat{w}}{dx} \right) \left(v_x \frac{d\hat{T}}{dx} - k \frac{d^2\hat{T}}{dx^2} - f \right) dx = 0 \quad (4.3.8)$$

ここで、左辺第 1 項から第 3 項までは Galerkin 法による離散化と同じなので Galerkin 項と呼び、第 4 項が新たに付加された安定化項である。なお、安定化項のパラメータ τ_e は要素ごとに独立であるので、安定化項は各要素 $\Omega_e = (x_1^e, x_2^e)$ の積分を全要素数 n まで総和を取ったものとしている。前項と同様に、2 節点一次要素を用いて離散化すると、各要素における有限要素方程式は次のようになる。

$$w_j^e \int_{x_1^e}^{x_2^e} v_x N_j^e \frac{dN_i^e}{dx} dx T_i^e + w_j^e \int_{x_1^e}^{x_2^e} k \frac{dN_j}{dx} \frac{dN_i}{dx} dx T_i^e - w_j^e \int_{x_1^e}^{x_2^e} N_j f dx + w_j^e \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} v_x \frac{dN_i}{dx} dx T_i^e - w_j^e \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} f dx = 0 \quad (4.3.9)$$

なお、安定化項における拡散項は 1 次要素を用いる場合には考慮されない。なぜならば、一次要素を用いると近似関数は一次関数になる、2 階微分を施すと 0 になるからである。また、重み関数 w_j^e は任意に選んだ関数であるので、有限要素法方程式は次式となる。

$$\int_{x_1^e}^{x_2^e} v_x N_j^e \frac{dN_i^e}{dx} dx T_i^e + \int_{x_1^e}^{x_2^e} \alpha \frac{dN_j}{dx} \frac{dN_i}{dx} dx T_i^e - \int_{x_1^e}^{x_2^e} N_j f dx + \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} v_x \frac{dN_i}{dx} dx T_i^e - \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} f dx = 0 \quad (4.3.10)$$

そして、各要素の有限要素方程式を行列表示すると次式となる。

$$(\mathbf{A}_e + \mathbf{D}_e + \mathbf{A}_{\delta_e}) \mathbf{T}_e = \mathbf{G}_e + \mathbf{G}_{\delta_e} \quad (4.3.11)$$

ここで、 $\mathbf{G}_e$ は発生項ベクトル、 $\mathbf{A}_{\delta_e}$ と $\mathbf{G}_{\delta_e}$ はそれぞれ安定化項における移流行列と発生項ベクトルであり、各形数行列は個々の要素ごとに以下のように表される。

$$\mathbf{A}_{\delta_e} = \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} v_x \frac{dN_i}{dx} dx T_i^e = \tau_e v_x^2 \begin{bmatrix} \frac{1}{h_e} & -\frac{1}{h_e} \\ -\frac{1}{h_e} & \frac{1}{h_e} \end{bmatrix} \quad (4.3.12)$$

$$\mathbf{G}_e = \int_{x_1^e}^{x_2^e} N_j f dx = \begin{Bmatrix} \frac{f h_e}{2} \\ -\frac{f h_e}{2} \end{Bmatrix} \quad (4.3.13)$$

$$\mathbf{G}_{\delta_e} = \tau_e \int_{x_1^e}^{x_2^e} v_x \frac{dN_j}{dx} f dx = \tau_e v_x \begin{Bmatrix} -f \\ f \end{Bmatrix} \quad (4.3.14)$$

なお、 $\mathbf{G}$ は発生項の英訳 Generation term のイニシャルから名付けている。式 (4.3.11) を全要素について重ね合わせると、全体の有限要素方程式が以下のように得られる。

$$(\mathbf{A} + \mathbf{D} + \mathbf{A}_{\delta}) \mathbf{T} = \mathbf{G} + \mathbf{G}_{\delta} \quad (4.3.15)$$

4.4 SUPG(Streamline Upwind/Petrov-Galerkin) 法

ここでは多次元の移流拡散方程式である式 (4.0.1) に対する Petrov-Galerkin 法の適用について考える。解析領域 Ω を n 個の 1 次要素の要素領域 $\Omega_e (e = 1, 2, \dots, n)$ に分割する。Petrov-Galerkin 法の重み関数 $\tilde{w}$ は式 (4.3.7) の安定化パラメータ τ_e を含む近似関数として以下のように拡張することができる。

$$\tilde{w} = \hat{w} + \tau_e \mathbf{v} \cdot \nabla \hat{w} = \hat{w} + \tau_e v_i \frac{\partial \hat{w}}{\partial x_i} \quad (4.4.1)$$

このとき移流拡散方程式 (4.0.1) の定常問題に対して、この重み関数 $\tilde{w}$ を用いた Petrov-Galerkin 法を適用し、さらに安定化パラメータ τ_e について整理すれば以下の離散化方程式が得られる。

$$\int_{\Omega} \tilde{w} v_i \frac{\partial \hat{T}}{\partial x_i} d\Omega + \int_{\Omega} k \frac{\partial \tilde{w}}{\partial x_i} \frac{\partial \hat{T}}{\partial x_i} d\Omega - \int_{\Omega} \tilde{w} f d\Omega + \sum_{e=1}^n \tau_e \int_{\Omega} v_i \frac{\partial \tilde{w}}{\partial x_i} \left(v_j \frac{\partial \hat{T}}{\partial x_j} - k \frac{\partial^2 \hat{T}}{\partial x_j^2} - f \right) d\Omega = \int_{\Gamma} \tilde{w} g_2 d\Omega \quad (4.4.2)$$

ここで、左辺第1項から第3項及び右辺項は Galerkin 項、第4項は付加された安定化項である。右辺項は Galerkin 項の拡散項を部分積分したことにより生じる境界積分項である。なお、一次要素を用いる場合は近似関数が一次関数となり、安定化項の拡散項は0となり考慮されない結果となる。

安定化項における第1項は

$$\tau_e \int_{\Omega} \left(v_i \frac{\partial \hat{w}}{\partial x_i} \right) \left(v_j \frac{\partial \hat{T}}{\partial x_j} \right) d\Omega \quad (4.4.3)$$

は、係数行列と節点値からなる2次形式となり、流れの流線方向 (streamline) のみに安定化作用を与える人工拡散となる。このことから、Upwind Galerkin 法を多次元に拡張した式 (4.4.2) に基づく方法は、**SUPG(Streamline Upwind/Petrov-Galerkin)** 法と呼ばれている。なお、多次元問題における安定化パラメータ τ_e も1次元問題と同様に以下のように多次元へ拡張して与えられる。

$$\tau_e = \frac{h_e}{2|\mathbf{v}|} \tilde{\xi}(P_e), \quad \tilde{\xi}(P_e) = \coth P_e - \frac{1}{P_e}, \quad P_e = \frac{|\mathbf{v}|h_e}{2k} \quad (4.4.4)$$

なお、文献 [4, pp.53-54] 『続・有限要素法による流れのシミュレーション』によると、 T が急激に変化する流線近傍ではオーバーシュートおよびアンダーシュートの挙動がみられる。この問題に対しては、式 (4.4.2) の左辺に次に示す衝撃捕捉項 (shock capturing) と呼ばれる項を付加することにより回避できる。

$$\sum_{e=1}^n \tau_{LSIC} \int_{\Omega_e} \frac{\partial \hat{w}}{\partial x_i} \frac{\partial \hat{T}}{\partial x_i} d\Omega \quad (4.4.5)$$

ここで、

$$\tau_{LSIC} = \frac{|\mathbf{v}|h_e}{2} \xi'(P_e), \quad \xi'(P_e) = \begin{cases} \frac{P_e}{3} & (P_e \leq 3) \\ 1 & (P_e > 3) \end{cases} \quad (4.4.6)$$

である。

4.5 要素サイズの検討

安定化項を計算する上で、要素サイズ h_e をどう決定するかという問題がある。これについては文献 [4, pp.81-82] 『続・有限要素法による流れのシミュレーション』に以下のような設定方法が記載されていた。

[引用開始]

要素の長さを定義する方法として、三角形の場合、

1. 要素面積と同じ面積の円の直径とする。
2. 要素を構成する辺の最長の長さとする。
3. 要素の2倍の面積と同じ面積の正方形の一片の長さとする。

などが考えられるが、ここでは、上流化という考えに基づき、流れの方向を考慮して要素長を定義する方法について解説する。流れの方向を考慮した要素長は、以下の式で定義される [1,2]。

$$h_e = 2 \left(\sum_{\alpha=1}^{n_{en}} \left| \left(\frac{\bar{u}_i^e}{\|\bar{u}_i^e\|} \right) \left(\frac{\partial N_{\alpha}^e}{\partial x_i} \right) \right| \right)^{-1}$$

[引用終了]

この式において、 n_{en} は要素での節点数 $\bar{u}_i^e$ は節点 i での移流速度、 $\|\bar{u}_i^e\|$ は流速の大きさ

$$\|\bar{u}_i^e\| = \sqrt{\sum_{i=1}^{n_d} (\bar{u}_i^e)^2} \quad (4.5.1)$$

である。 $(n_d$ は次元数である。) これを参考に要素サイズの計算においてこの式をあてはめようと試みた。具体的に四角形一次要素のとき、以下ようになる。

$$h_e = 2 \left(\sum_{\alpha=1}^4 \left| \frac{v_x}{\sqrt{v_x^2 + v_y^2}} \frac{\partial N_{\alpha}^e}{\partial x} + \frac{v_y}{\sqrt{v_x^2 + v_y^2}} \frac{\partial N_{\alpha}^e}{\partial y} \right| \right)^{-1} \quad (4.5.2)$$

この式は流速 v_x, v_y が0または十分小さいときには要素サイズ h_e もまた0となってしまう。そのため、本研究では流速が 0.001[m/s] 以下は流速が十分に小さいとして、要素の各辺の長さの平均値を要素サイズとして設定した。

4.6 二次元定常移流拡散方程式への SUPG 法の適用

この節では、本研究で取り扱う以下で示される二次元定常移流拡散問題に SUPG 法を適用させ、さらに衝撃捕捉項を付加させることで安定化を施した有限要素法の適用を考える。

$$\begin{cases} v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 @(\Omega \text{上}) \\ T = g_1 \Gamma_1 \text{上}, \quad \text{frac} \partial T \partial \mathbf{n} = g_2 @(\Gamma_2 \text{上}) \end{cases} \quad (4.6.1)$$

$$T = g_1 \Gamma_1 \text{上}, \quad \text{frac} \partial T \partial \mathbf{n} = g_2 @(\Gamma_2 \text{上}) \quad (4.6.2)$$

式 (4.6.1) に重み付き残差法を適用させていく。

$$\int_{\Omega} \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} \right) w d\Omega - k \int_{\Omega} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) w d\Omega = 0 \quad (4.6.3)$$

この式に Petrov Galerkin 法を適用させていく。要素分割に四角形一次要素を用いると、要素内での近似関数である内挿関数と重み関数は以下の式になる。

$$\begin{cases} \hat{T} = \sum_{i=1}^4 N_i^e T_i^e = N_1^e T_1^e + N_2^e T_2^e + N_3^e T_3^e + N_4^e T_4^e \\ \hat{w} = \sum_{i=1}^4 N_i^e w_i^e = N_1^e w_1^e + N_2^e w_2^e + N_3^e w_3^e + N_4^e w_4^e \end{cases} \quad (4.6.4)$$

ここで Petrov-Galerkin 法を適用し、重み関数に

$$\tilde{w} = \hat{w} + \tau_e \mathbf{v} \cdot \nabla \hat{w} = \hat{w} + \tau_e v_i \frac{\partial \hat{w}}{\partial x_i} = \hat{w} + \tau_e \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \quad (4.6.5)$$

$$\tau_e = \frac{h_e}{2|\mathbf{v}|} \tilde{\xi}(P_e), \quad \tilde{\xi}(P_e) = \coth P_e - \frac{1}{P_e}, \quad P_e = \frac{|\mathbf{v}| h_e}{2k} \quad (4.6.6)$$

を使う。これらを式 (4.6.3) にあてはめる。

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) \tilde{w} d\Omega - k \int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) \tilde{w} d\Omega = 0 \quad (4.6.7)$$

$$\begin{aligned} & \int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) \left(\hat{w} + \sum_{e=1}^n \tau_e \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \right) d\Omega \\ & - k \int_{\Omega} \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) \left(\hat{w} + \sum_{e=1}^n \tau_e \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \right) d\Omega = 0 \end{aligned} \quad (4.6.8)$$

$$\begin{aligned} & \int_{\Omega} \hat{w} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_x \frac{\partial \hat{T}}{\partial x} \right) d\Omega - \int_{\Omega} k \left(\hat{w} \frac{\partial^2 \hat{T}}{\partial x^2} + \hat{w} \frac{\partial^2 \hat{T}}{\partial y^2} \right) d\Omega \\ & + \sum_{e=1}^n \tau_e \int_{\Omega} \left(v_x \frac{\partial \hat{w}}{\partial x} + v_x \frac{\partial \hat{w}}{\partial x} \right) \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} - k \frac{\partial^2 \hat{T}}{\partial x^2} - k \frac{\partial^2 \hat{T}}{\partial y^2} \right) d\Omega = 0 \end{aligned} \quad (4.6.9)$$

ここで、 n は全要素数である。この式の Galerkin 項に対して、Green の公式を適用して弱形式を求めると以下のようになる。

$$\begin{aligned} & \int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w d\Omega + k \int_{\Omega} \left(\frac{\partial T}{\partial x} \frac{\partial \hat{w}}{\partial x} + \frac{\partial T}{\partial y} \frac{\partial \hat{w}}{\partial y} \right) d\Omega \\ & + \sum_{e=1}^n \tau_e \int_{\Omega} \left(v_x \frac{\partial \hat{w}}{\partial x} + v_x \frac{\partial \hat{w}}{\partial x} \right) \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} - k \frac{\partial^2 \hat{T}}{\partial x^2} - k \frac{\partial^2 \hat{T}}{\partial y^2} \right) d\Omega = k \int_{\Gamma_2} g_2 \hat{w} d\gamma \end{aligned} \quad (4.6.10)$$

一次要素を用いる場合は T が一次関数となるため、 $\frac{\partial T^2}{\partial x^2} = \frac{\partial T^2}{\partial y^2} = 0$ となり安定化項の拡散項は考慮されない結果となる。さらに左辺に衝撃捕捉項

$$\sum_{e=1}^n \tau_{LSIC} \int_{\Omega_e} \frac{\partial \hat{w}}{\partial x_i} \frac{\partial \hat{T}}{\partial x_i} d\Omega = \sum_{e=1}^n \tau_{LSIC} \int_{\Omega_e} \left(\frac{\partial \hat{w}}{\partial x} \frac{\partial \hat{T}}{\partial x} + \frac{\partial \hat{w}}{\partial y} \frac{\partial \hat{T}}{\partial y} \right) d\Omega \quad (4.6.11)$$

を加えて次の式になる。

$$\begin{aligned} \int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) \hat{w} d\Omega + k \int_{\Omega} \left(\frac{\partial T}{\partial x} \frac{\partial \hat{w}}{\partial x} + \frac{\partial T}{\partial y} \frac{\partial \hat{w}}{\partial y} \right) d\Omega + \sum_{e=1}^n \tau_e \int_{\Omega} \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) d\Omega \\ + \sum_{e=1}^n \tau_{LSIC} \int_{\Omega_e} \left(\frac{\partial \hat{w}}{\partial x} \frac{\partial \hat{T}}{\partial x} + \frac{\partial \hat{w}}{\partial y} \frac{\partial \hat{T}}{\partial y} \right) d\Omega = k \int_{\Gamma_2} g_2 \hat{w} d\gamma \end{aligned} \quad (4.6.12)$$

ここで、衝撃捕捉項の安定化パラメータ τ_{LSIC} は以下で示される。

$$\tau_{LSIC} = \frac{|v| h_e}{2} \xi'(P_e), \quad \xi'(P_e) = \begin{cases} \frac{P_e}{3} & (P_e \leq 3) \\ 1 & (P_e > 3) \end{cases} \quad (4.6.13)$$

ここで式 (4.6.4) より $\hat{T}, \hat{w}$ に内挿近似式を代入していく。

$$\begin{aligned} \sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(v_x N_j^e \frac{\partial N_i^e}{\partial x} + v_y N_j^e \frac{\partial N_i^e}{\partial y} \right) d\Omega + k \sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega \\ + \tau_e \sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(v_x \frac{\partial N_j^e}{\partial x} + v_y \frac{\partial N_j^e}{\partial y} \right) \left(v_x \frac{\partial N_i^e}{\partial x} + v_y \frac{\partial N_i^e}{\partial y} \right) d\Omega \\ + \tau_{LSIC} \sum_{i=1}^4 \sum_{j=1}^4 T_i^e w_j^e \int_{\Omega_e} \left(\frac{\partial N_j^e}{\partial x} \frac{\partial N_i^e}{\partial x} + \frac{\partial N_j^e}{\partial y} \frac{\partial N_i^e}{\partial y} \right) d\Omega = \sum_{j=1}^4 w_j^e \int_{\Gamma_2} g_2 N_j^e d\gamma \end{aligned} \quad (4.6.14)$$

これを行列表示すると以下となる。

$$\mathbf{w}_e^T (\mathbf{A}_e + \mathbf{D}_e + \mathbf{A}_{\delta_e} + \mathbf{D}_{\delta_e}) \mathbf{T}_e = \mathbf{w}_e^T \mathbf{F}_e \quad (4.6.15)$$

ここで $\mathbf{w}_e$ は任意に選んだ定数であるので、両辺を除することができ

$$(\mathbf{A}_e + \mathbf{D}_e + \mathbf{A}_{\delta_e} + \mathbf{D}_{\delta_e}) \mathbf{T}_e = \mathbf{F}_e \quad (4.6.16)$$

となる。この式は4元連立一次方程式となっている。

ここに、 $\mathbf{A}_e, \mathbf{D}_e, \mathbf{A}_{\delta_e}, \mathbf{D}_{\delta_e}, \mathbf{T}_e, \mathbf{F}_e$ はそれぞれ、移流行列、拡散行列、安定化行列、衝撃捕捉行列、温度ベクトル、熱フラックスベクトルである。各行列・ベクトルの定義を以下に示す。

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} N_1^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & N_4^e \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) \end{bmatrix} d\Omega \quad (4.6.17)$$

$$\mathbf{D}_e = k \int_{\Omega_e} \begin{bmatrix} \frac{\partial N_1^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_2^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_3^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_4^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_4^e}{\partial y} \end{bmatrix} d\Omega \quad (4.6.18)$$

$$\mathbf{A}_{\delta_e} = \tau_e \int_{\Omega_e} \left[\begin{array}{cccc} \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) \\ \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) & \left(v_x \frac{\partial N_4^e}{\partial x} + v_y \frac{\partial N_4^e}{\partial y} \right) \end{array} \right] d\Omega \quad (4.6.19)$$

$$\mathbf{D}_{\delta_e} = \tau_{LSIC} \int_{\Omega_e} \left[\begin{array}{cccc} \frac{\partial N_1^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_2^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_3^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_4^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_3^e}{\partial y} & \frac{\partial N_4^e}{\partial x} \frac{\partial N_4^e}{\partial x} + \frac{\partial N_4^e}{\partial y} \frac{\partial N_4^e}{\partial y} \end{array} \right] d\Omega \quad (4.6.20)$$

$$\mathbf{T}_e = \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \\ T_4^e \end{Bmatrix}, \quad \mathbf{w}_e = \begin{Bmatrix} w_1^e \\ w_2^e \\ w_3^e \\ w_4^e \end{Bmatrix} \quad (4.6.21)$$

ここで、 $\mathbf{F}$ は隣接する要素間ではお互いに相殺されるため、要素境界辺での境界積分は、重ね合わせを行った段階で消去され、最終的に領域の境界 Γ のみの積分が残ることとなる。例えば、ある要素 Ω_e の境界 Γ_{12} で自然境界条件が与えられている場合以下のようなになる。

$$\mathbf{F}_e = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ 0 \\ 0 \end{Bmatrix} d\gamma \quad (4.6.22)$$

流速 v_x, v_y は要素内で内挿近似することで以下のような近似式となる。

$$v_x \approx \hat{v}_x = \sum_{i=1}^4 N_i^e v_{xi}^e = N_1^e v_{x1}^e + N_2^e v_{x2}^e + N_3^e v_{x3}^e + N_4^e v_{x4}^e \quad (4.6.23)$$

$$v_y \approx \hat{v}_y = \sum_{i=1}^4 N_i^e v_{yi}^e = N_1^e v_{y1}^e + N_2^e v_{y2}^e + N_3^e v_{y3}^e + N_4^e v_{y4}^e \quad (4.6.24)$$

ここで、 v_{xi}^e, v_{yi}^e はある要素 e の要素節点番号 i^* での流速である。

直接剛性法により、全要素数 n まで足し合わせることで全体の有限要素式が次のように求まる。

$$(\mathbf{A} + \mathbf{D} + \mathbf{A}_\delta + \mathbf{D}_\delta) \mathbf{T} = \mathbf{F} \quad (4.6.25)$$

このようにして SUPG 法と衝撃捕捉項による安定化を施した場合の有限要素式が求まった、

第5章 三角形要素での解析

円のように曲線のある領域については四角形要素だけだと領域の形状を再現することが困難な場合がある。そういったときに三角形要素を使うことが有効となる。三角形要素の適用は研究が進んでおり、積分の公式が存在しているため形状関数も比較的簡単に座標変換を行わなくても計算ができる。このように、三角形要素には計算が行いやすい利点がある。一方で、節点が少ないため形状関数を使つての近似が他のものよりも精度が悪いという点もある。

本研究ではあくまで四角形での要素分割がメインであるが、ここまでで述べた利点および、解析の段階での他の文献との比較の点からも三角形要素での解析が行えるように実装を試みた。ここでは、実装に当たって必要と思われる三角形要素での形状関数および積分したときの式について中心に記述する。

5.1 三角形要素での近似関数

三角形領域における近似した温度 $\hat{T}$ を式で表現すると次のようになる。

$$\hat{T} = \alpha_1 + \alpha_2 x + \alpha_3 y \quad (5.1.1)$$

ここで、 $\alpha_1 \sim \alpha_3$ は係数である。

ある要素 Ω_e の要素節点番号 i^* においては次のように表される。

$$\hat{T} = T_i^e = \alpha_1^e + \alpha_2^e x_i^e + \alpha_3^e y_i^e \quad (i = 1 \sim 3) \quad (5.1.2)$$

ここで、 e のついた T_i^e, x_i^e, y_i^e はそれぞれある要素 e の要素節点番号 i^* での温度と座標値であり、同様に $\alpha_1^e \sim \alpha_3^e$ はある要素 e の要素節点番号 i^* での係数である。

これを図に示すと図 5.1 のようになる。この図において数字は要素節点番号を表す。

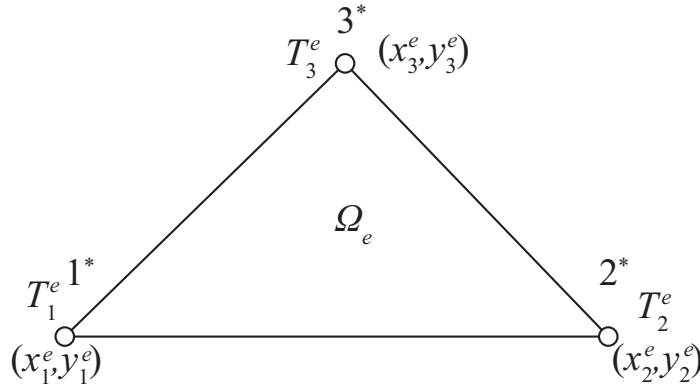


図 5.1: 三角形要素

式 (5.1.2) を行列表示で書くと次のようになる。

$$\begin{bmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{bmatrix} \begin{Bmatrix} \alpha_1^e \\ \alpha_2^e \\ \alpha_3^e \end{Bmatrix} = \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} \quad (5.1.3)$$

これを α について整理する。

$$\begin{Bmatrix} \alpha_1^e \\ \alpha_2^e \\ \alpha_3^e \end{Bmatrix} = \begin{bmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{bmatrix}^{-1} \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} = \frac{1}{\begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix}} \begin{bmatrix} x_2^e y_3^e - x_3^e y_2^e & x_3^e y_1^e - x_1^e y_3^e & x_1^e y_2^e - x_2^e y_1^e \\ y_2^e - y_3^e & y_3^e - y_1^e & y_1^e - y_2^e \\ x_3^e - x_2^e & x_1^e - x_3^e & x_2^e - x_1^e \end{bmatrix} \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} \quad (5.1.4)$$

ここで、

$$D^e = \begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix} = x_2^e y_1^e + x_3^e y_1^e - (x_2^e y_1^e + x_3^e y_2^e + x_1^e y_3^e) \quad (5.1.5)$$

$$= x_1^e (y_2^e - y_3^e) + x_2^e (y_3^e - y_1^e) + x_3^e (y_1^e - y_2^e) \quad (5.1.6)$$

となる。 D^e はある三角形要素 Ω_e の面積の二倍となっている。

$$a_i^{e*} = x_j^e y_k^e - x_k^e y_j^e, \quad b_i^{e*} = y_j^e - y_k^e, \quad c_i^{e*} = x_k^e - x_j^e \quad (i, j, k \text{ は偶置換}) \quad (5.1.7)$$

偶置換とは、 $i=1$ のとき、 $j=2, k=3$ となり、 $i=2$ のとき、 $j=3, k=1$ となり、 $i=3$ のとき、 $j=1, k=2$ となることである。上記の記号を用いて、さらに

$$a_i^e = \frac{a_i^{e*}}{D^e}, \quad b_i^e = \frac{b_i^{e*}}{D^e}, \quad c_i^e = \frac{c_i^{e*}}{D^e} \quad (5.1.8)$$

とすると、次のようになる。

$$\begin{Bmatrix} \alpha_1^e \\ \alpha_2^e \\ \alpha_3^e \end{Bmatrix} = \frac{1}{D^e} \begin{bmatrix} a_1^{e*} & a_2^{e*} & a_3^{e*} \\ b_1^{e*} & b_2^{e*} & b_3^{e*} \\ c_1^{e*} & c_2^{e*} & c_3^{e*} \end{bmatrix} \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} = \begin{bmatrix} a_1^e & a_2^e & a_3^e \\ b_1^e & b_2^e & b_3^e \\ c_1^e & c_2^e & c_3^e \end{bmatrix} \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} \quad (5.1.9)$$

これを連立方程式の形で書きなおすと次のようになる。

$$\begin{cases} \alpha_1^e = a_1^e T_1^e + a_2^e T_2^e + a_3^e T_3^e \\ \alpha_2^e = b_1^e T_1^e + b_2^e T_2^e + b_3^e T_3^e \\ \alpha_3^e = c_1^e T_1^e + c_2^e T_2^e + c_3^e T_3^e \end{cases} \quad (5.1.10)$$

よって、式 (5.1.2) に式 (5.1.10) を代入することで近似関数は次のように表される。

$$\hat{T} = a_1^e T_1^e + a_2^e T_2^e + a_3^e T_3^e + (b_1^e T_1^e + b_2^e T_2^e + b_3^e T_3^e)x + (c_1^e T_1^e + c_2^e T_2^e + c_3^e T_3^e)y \quad (5.1.11)$$

$$= \sum_{i=1}^3 (a_i^e + b_i^e x + c_i^e y) T_i^e \quad (5.1.12)$$

ここで、 $()$ の部分は三角形の面積を表すことから面積座標と呼ばれ、形状関数 L_i とする。

$$L_i^e(x, y) = a_i^e + b_i^e x + c_i^e y \quad (i = 1 \sim 3) \quad (5.1.13)$$

したがって、

$$\hat{T} = \sum_{i=1}^3 L_i^e T_i^e \quad (5.1.14)$$

と近似できる。

ここで、形状関数に関連したものを整理する。

$$\begin{cases} L_1^e = a_1^e + b_1^e x + c_1^e y, & \frac{\partial L_1^e}{\partial x} = b_1^e, & \frac{\partial L_1^e}{\partial y} = c_1^e \\ L_2^e = a_2^e + b_2^e x + c_2^e y, & \frac{\partial L_2^e}{\partial x} = b_2^e, & \frac{\partial L_2^e}{\partial y} = c_2^e \\ L_3^e = a_3^e + b_3^e x + c_3^e y, & \frac{\partial L_3^e}{\partial x} = b_3^e, & \frac{\partial L_3^e}{\partial y} = c_3^e \end{cases} \quad (5.1.15)$$

$$\begin{cases} a_1^e = \frac{a_1^{e*}}{D^e} = \frac{x_2^e y_3^e - x_3^e y_2^e}{D^e}, & a_2^e = \frac{a_2^{e*}}{D^e} = \frac{x_3^e y_1^e - x_1^e y_3^e}{D^e}, & a_3^e = \frac{a_3^{e*}}{D^e} = \frac{x_1^e y_2^e - x_2^e y_1^e}{D^e} \\ b_1^e = \frac{b_1^{e*}}{D^e} = \frac{y_2^e - y_3^e}{D^e}, & b_2^e = \frac{b_2^{e*}}{D^e} = \frac{y_3^e - y_1^e}{D^e}, & b_3^e = \frac{b_3^{e*}}{D^e} = \frac{y_1^e - y_2^e}{D^e} \\ c_1^e = \frac{c_1^{e*}}{D^e} = \frac{x_3^e - x_2^e}{D^e}, & c_2^e = \frac{c_2^{e*}}{D^e} = \frac{x_1^e - x_3^e}{D^e}, & c_3^e = \frac{c_3^{e*}}{D^e} = \frac{x_2^e - x_1^e}{D^e} \end{cases} \quad (5.1.16)$$

$$D^e = \begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix} = x_2^e y_3^e + x_1^e y_2^e + x_3^e y_1^e - (x_2^e y_1^e + x_3^e y_2^e + x_1^e y_3^e) \quad (5.1.17)$$

$$= x_1^e (y_2^e - y_3^e) + x_2^e (y_3^e - y_1^e) + x_3^e (y_1^e - y_2^e) \quad (5.1.18)$$

ここで、ある三角形要素 Ω_e の面積を S^e とすると、 $D^e = 2S^e$ が成立する。
面積座標には積分の公式が存在し、以下ようになる。

$$\int \int L_1^{el} L_2^{em} L_3^{en} dx dy = 2S^e \frac{l!m!n!}{(l+m+n+2)!} \quad (5.1.19)$$

S^e は三角形要素 e の面積を表し、 l, m, n は形状関数の指数を表す。

5.2 安定化を施した有限要素式への適用

本研究で取り扱う問題は以下で示される二次元定常移流拡散問題である。

$$\begin{cases} v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 & (\Omega \text{内}) \\ T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial \mathbf{n}} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (5.2.1)$$

$$\begin{cases} T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial \mathbf{n}} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (5.2.2)$$

この問題に対して三角形要素を使った安定化有限要素法の適用を考える。三角形要素を使うとき、近似関数と近似形の重み関数 $\hat{T}, \hat{w}$ は次のようになる。

$$\hat{T} = \sum_{i=1}^3 N_i^e T_i^e = \sum_{i=1}^3 L_i^e T_i^e, \quad \hat{w} = \sum_{i=1}^3 N_i^e w_i^e = \sum_{i=1}^3 L_i^e w_i^e \quad (5.2.3)$$

ここで、安定化を施すため Petrov-Galerkin 法を適用し重み関数に次の式を用いる。

$$\tilde{w} = \hat{w} + \tau_e \mathbf{v} \cdot \nabla \hat{w} = \hat{w} + \tau_e v_i \frac{\partial \hat{w}}{\partial x_i} = \hat{w} + \tau_e \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \quad (5.2.4)$$

基本的には4.6節と同じようにして有限要素式が導かれる。式の形として異なる点は内挿関数 N_i^e の数が3となる点である。したがって途中の導出過程を省略した。

要素分割に三角形要素を用いて、さらに安定化を施した有限要素式は以下ようになる。

$$(\mathbf{A}_e + \mathbf{D}_e + \mathbf{A}_{\delta_e} + \mathbf{D}_{\delta_e}) \mathbf{T}_e = \mathbf{F}_e \quad (5.2.5)$$

各行列とベクトルは以下のようにそれぞれ 3×3 成分の行列と3行のベクトルなる。

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} N_1^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \right) \\ N_1^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_2^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) & N_3^e \left(v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \right) \end{bmatrix} d\Omega \quad (5.2.6)$$

$$\mathbf{D}_e = k \int_{\Omega_e} \begin{bmatrix} \frac{\partial N_1^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_3^e}{\partial y} \\ \frac{\partial N_2^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_3^e}{\partial y} \\ \frac{\partial N_3^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_3^e}{\partial y} \end{bmatrix} d\Omega \quad (5.2.7)$$

$$\mathbf{A}_{\delta_e} = \tau_e \int_{\Omega_e} \begin{bmatrix} \begin{pmatrix} v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} \begin{pmatrix} v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} & \begin{pmatrix} v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \end{pmatrix} \begin{pmatrix} v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} \\ \begin{pmatrix} v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} \begin{pmatrix} v_x \frac{\partial N_1^e}{\partial x} + v_y \frac{\partial N_1^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} & \begin{pmatrix} v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \\ v_x \frac{\partial N_2^e}{\partial x} + v_y \frac{\partial N_2^e}{\partial y} \end{pmatrix} \begin{pmatrix} v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \\ v_x \frac{\partial N_3^e}{\partial x} + v_y \frac{\partial N_3^e}{\partial y} \end{pmatrix} \end{bmatrix} d\Omega \quad (5.2.8)$$

$$\mathbf{D}_{\delta_e} = \tau_{LSIC} \int_{\Omega_e} \begin{bmatrix} \frac{\partial N_1^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_1^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_1^e}{\partial y} \frac{\partial N_3^e}{\partial y} \\ \frac{\partial N_2^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_2^e}{\partial y} \frac{\partial N_3^e}{\partial y} \\ \frac{\partial N_3^e}{\partial x} \frac{\partial N_1^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_1^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_2^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_2^e}{\partial y} & \frac{\partial N_3^e}{\partial x} \frac{\partial N_3^e}{\partial x} + \frac{\partial N_3^e}{\partial y} \frac{\partial N_3^e}{\partial y} \end{bmatrix} d\Omega \quad (5.2.9)$$

$$\mathbf{T}_e = \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix} \quad (5.2.10)$$

熱フラックスベクトル $\mathbf{F}_e$ は隣接要素境界辺で流束が相殺されるため、直接剛性法による重ね合わせの段階で要素境界辺での境界積分は消去される。そして最終的には領域の境界のみの積分が残る。

したがって、ある要素 Ω_e の境界 Γ_{12} で熱フラックスが既知の場合は以下ようになる。

$$\mathbf{F}_e = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ N_3^e \end{Bmatrix} d\gamma = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ 0 \end{Bmatrix} d\gamma \quad (5.2.11)$$

三角形要素を使うときは内挿関数 N_i^e は形状関数 L_i^e となる。本研究で取り扱う問題は2次元であるため、境界積分においては次元が1次元となる。1次元の場合の面積座標は長さ座標または線座標と呼ばれる。1次元区間Iにおいては以下で示される長さ座標の積分公式が存在する。

$$\int_I L_1^i L_2^j dx = \frac{i!j!L}{(1+i+j)!} \quad (i, j \text{ は整数}) \quad (5.2.12)$$

ここで、Lは区間Iの長さである。

熱フラックスベクトルでの境界積分の各成分に対して長さ座標の積分公式を適用する。各成分は形状関数が1つしかないので1行目と2行目の成分は以下の同じ計算となる。

$$\int_{\Gamma_e} L_1^1 L_2^0 d\gamma = \int_{\Gamma_e} L_1^0 L_2^1 d\gamma = \frac{h_{12}^e}{(1+1)!} = \frac{h_{12}^e}{2} \quad (5.2.13)$$

ここで、 h_{12}^e は三角形要素の辺1-2の長さである。したがって、熱フラックスベクトルは以下で示される。

$$\mathbf{F}_e = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ 0 \end{Bmatrix} d\gamma = \frac{kg_2 h_{12}^e}{2} \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} \quad (5.2.14)$$

他の行列については、さらに $\frac{\partial N_i^e}{\partial x}, \frac{\partial N_i^e}{\partial y}$ の項も計算することができ次のように表される。

$$\mathbf{A}_e = \int_{\Omega_e} \begin{bmatrix} L_1^e (v_x b_1^e + v_y c_1^e) & L_2^e (v_x b_1^e + v_y c_1^e) & L_3^e (v_x b_1^e + v_y c_1^e) \\ L_1^e (v_x b_2^e + v_y c_2^e) & L_2^e (v_x b_2^e + v_y c_2^e) & L_3^e (v_x b_2^e + v_y c_2^e) \\ L_1^e (v_x b_3^e + v_y c_3^e) & L_2^e (v_x b_3^e + v_y c_3^e) & L_3^e (v_x b_3^e + v_y c_3^e) \end{bmatrix} d\Omega \quad (5.2.15)$$

$$\mathbf{D}_e = k \int_{\Omega_e} \begin{bmatrix} b_1^e b_1^e + c_1^e c_1^e & b_1^e b_2^e + c_1^e c_2^e & b_1^e b_3^e + c_1^e c_3^e \\ b_2^e b_1^e + c_2^e c_1^e & b_2^e b_2^e + c_2^e c_2^e & b_2^e b_3^e + c_2^e c_3^e \\ b_3^e b_1^e + c_3^e c_1^e & b_3^e b_2^e + c_3^e c_2^e & b_3^e b_3^e + c_3^e c_3^e \end{bmatrix} d\Omega \quad (5.2.16)$$

$$\mathbf{A}_{\delta_e} = \tau_e \int_{\Omega_e} \begin{bmatrix} (v_x b_1^e + v_y c_1^e)(v_x b_1^e + v_y c_1^e) & (v_x b_2^e + v_y c_2^e)(v_x b_1^e + v_y c_1^e) & (v_x b_3^e + v_y c_3^e)(v_x b_1^e + v_y c_1^e) \\ (v_x b_1^e + v_y c_1^e)(v_x b_2^e + v_y c_2^e) & (v_x b_2^e + v_y c_2^e)(v_x b_2^e + v_y c_2^e) & (v_x b_3^e + v_y c_3^e)(v_x b_2^e + v_y c_2^e) \\ (v_x b_1^e + v_y c_1^e)(v_x b_3^e + v_y c_3^e) & (v_x b_2^e + v_y c_2^e)(v_x b_3^e + v_y c_3^e) & (v_x b_3^e + v_y c_3^e)(v_x b_3^e + v_y c_3^e) \end{bmatrix} d\Omega \quad (5.2.17)$$

$$\mathbf{D}_{\delta_e} = \tau_{LSIC} \int_{\Omega_e} \begin{bmatrix} b_1^e b_1^e + c_1^e c_1^e & b_1^e b_2^e + c_1^e c_2^e & b_1^e b_3^e + c_1^e c_3^e \\ b_2^e b_1^e + c_2^e c_1^e & b_2^e b_2^e + c_2^e c_2^e & b_2^e b_3^e + c_2^e c_3^e \\ b_3^e b_1^e + c_3^e c_1^e & b_3^e b_2^e + c_3^e c_2^e & b_3^e b_3^e + c_3^e c_3^e \end{bmatrix} d\Omega \quad (5.2.18)$$

ここで、いま流速 v_x, v_y は要素内で節点での流速 v_{xi}^e, v_{yi}^e を用いて内挿近似するので、

$$v_x = \sum_{i=1}^3 = L_i^e v_{xi}^e = L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e, \quad v_y = \sum_{i=1}^3 = L_i^e v_{yi}^e = L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e \quad (5.2.19)$$

であることに注意する。流速を内挿近似することで、 $\mathbf{A}_e, \mathbf{A}_{\delta_e}$ の i, j 成分をさらに式展開していく。

$\mathbf{A}_e$ の i, j 成分は

$$\begin{aligned} L_j^e (v_x b_i^e + v_y c_i^e) &= L_j^e \{ (L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e) b_i^e + (L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e) c_i^e \} \\ &= L_j^e L_1^e (v_{x1}^e b_i^e + v_{y1}^e c_i^e) + L_j^e L_2^e (v_{x2}^e b_i^e + v_{y2}^e c_i^e) + L_j^e L_3^e (v_{x3}^e b_i^e + v_{y3}^e c_i^e) = \sum_{k=1}^3 L_j^e L_k^e (v_{xk}^e b_i^e + v_{yk}^e c_i^e) \end{aligned}$$

となる。また、 $\mathbf{A}_{\delta_e}$ の i, j 成分は全体にかかっている τ_e を除いて考えると

$$\begin{aligned} & (v_x b_j^e + v_y c_j^e) (v_x b_i^e + v_y c_i^e) = v_x v_x b_j^e b_i^e + v_x v_y b_j^e c_i^e + v_y v_x c_j^e b_i^e + v_y v_y c_j^e c_i^e \\ &= (L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e) (L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e) b_j^e b_i^e + (L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e) (L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e) b_j^e c_i^e \\ &+ (L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e) (L_1^e v_{x1}^e + L_2^e v_{x2}^e + L_3^e v_{x3}^e) c_j^e b_i^e + (L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e) (L_1^e v_{y1}^e + L_2^e v_{y2}^e + L_3^e v_{y3}^e) c_j^e c_i^e \\ &= \{ L_1^{e2} v_{x1}^e v_{x1}^e + L_1^e L_2^e v_{x1}^e v_{x2}^e + L_1^e L_3^e v_{x1}^e v_{x3}^e + L_2^e L_1^e v_{x2}^e v_{x1}^e + L_2^{e2} v_{x2}^e v_{x2}^e + L_2^e L_3^e v_{x2}^e v_{x3}^e \\ &\quad + L_3^e L_1^e v_{x3}^e v_{x1}^e + L_3^e L_2^e v_{x3}^e v_{x2}^e + L_3^{e2} v_{x3}^e v_{x3}^e \} b_j^e b_i^e \\ &+ \{ L_1^{e2} v_{x1}^e v_{y1}^e + L_1^e L_2^e v_{x1}^e v_{y2}^e + L_1^e L_3^e v_{x1}^e v_{y3}^e + L_2^e L_1^e v_{x2}^e v_{y1}^e + L_2^{e2} v_{x2}^e v_{y2}^e + L_2^e L_3^e v_{x2}^e v_{y3}^e \\ &\quad + L_3^e L_1^e v_{x3}^e v_{y1}^e + L_3^e L_2^e v_{x3}^e v_{y2}^e + L_3^{e2} v_{x3}^e v_{y3}^e \} b_j^e c_i^e \\ &+ \{ L_1^{e2} v_{y1}^e v_{x1}^e + L_1^e L_2^e v_{y1}^e v_{x2}^e + L_1^e L_3^e v_{y1}^e v_{x3}^e + L_2^e L_1^e v_{y2}^e v_{x1}^e + L_2^{e2} v_{y2}^e v_{x2}^e + L_2^e L_3^e v_{y2}^e v_{x3}^e \\ &\quad + L_3^e L_1^e v_{y3}^e v_{x1}^e + L_3^e L_2^e v_{y3}^e v_{x2}^e + L_3^{e2} v_{y3}^e v_{x3}^e \} c_j^e b_i^e \\ &+ \{ L_1^{e2} v_{y1}^e v_{y1}^e + L_1^e L_2^e v_{y1}^e v_{y2}^e + L_1^e L_3^e v_{y1}^e v_{y3}^e + L_2^e L_1^e v_{y2}^e v_{y1}^e + L_2^{e2} v_{y2}^e v_{y2}^e + L_2^e L_3^e v_{y2}^e v_{y3}^e \\ &\quad + L_3^e L_1^e v_{y3}^e v_{y1}^e + L_3^e L_2^e v_{y3}^e v_{y2}^e + L_3^{e2} v_{y3}^e v_{y3}^e \} c_j^e c_i^e = \sum_{k=1}^3 \sum_{l=1}^3 \sum_{\alpha=1}^2 \sum_{\beta=1}^2 L_k^e L_l^e v_{\alpha k}^e v_{\beta l}^e \frac{\partial L_j^e}{\partial \alpha} \frac{\partial L_i^e}{\partial \beta} \end{aligned} \quad (5.2.20)$$

となる。ここで表記を簡潔にするために、 v_x, v_y の x, y が添え字 1, 2 に対応すると考え記号 α, β でおきなおした。

これらの行列の各成分は積分の式となっている。いま形状関数に面積座標を用いており、面積座標の積分公式

$$\int \int L_1^l L_2^m L_3^n dx dy = 2S \frac{l!m!n!}{(l+m+n+2)!} \quad (5.2.21)$$

よりさらに計算することができる。なお、三角形の面積は要素ごとに異なるので S^e とする。 $\mathbf{D}_e, \mathbf{D}_{\delta_e}$ の成分には形状関数 L_i^e が登場していないので、積分公式における $l, m, n = 0$ のときと同じとなる。したがって、この二つに関しては三角形の面積 S^e を掛けることで積分したことになる。

$\mathbf{A}_e, \mathbf{A}_{\delta_e}$ の成分については、 L_i^{e2} の項と $L_i^e L_j^e$ の項とが存在する。 L_i^{e2} の項が存在するときは、公式における l, m, n のいずれか一つのみが 2 で他は 0 という場合で L_i^e の添え字 i によらず、

$$\int \int L_1^2 L_2^0 L_3^0 dx dy = \int \int L_1^0 L_2^2 L_3^0 dx dy = \int \int L_1^0 L_2^0 L_3^2 dx dy = 2S^e \frac{2!}{(2+2)!} = \frac{4S^e}{4!} = \frac{S^e}{3 \cdot 2} = \frac{S^e}{6} \quad (5.2.22)$$

となる。同様に $L_i^e L_j^e$ の項も添え字 i, j によらず、

$$\int \int L_1^1 L_2^1 L_3^0 dx dy = \int \int L_1^0 L_2^1 L_3^1 dx dy = \int \int L_1^1 L_2^0 L_3^1 dx dy = 2S^e \frac{1!1!}{(1+1+2)!} = \frac{2S^e}{4!} = \frac{S^e}{4 \cdot 3} = \frac{S^e}{12} \quad (5.2.23)$$

となる。

本来であれば、 L_i^2 の項の存在する成分と L_i^1 の成分とで場合分けを行って積分の計算をしていく必要がある。しかし、 L_i^2 のある成分では値が $S^e/6$ であり、一方 L_i^1 のある成分は値が $S^e/12$ である。これから分かるように、 L_i^2 のある成分は $S^e/12$ を 2 回足すことで表現できる。そのため、これらの L_i^2 と L_i^1 の混ざった成分のある $\mathbf{A}_e, \mathbf{A}_{\delta_e}$ の計算は次のように行う。 L_i^2 のある成分はその項の係っている係数を予め $S^e/12$ して足しておき、さらに全体を $S^e/12$ 倍して足す。こうすることで、場合分けを行わずに計算を行える。

ここまでのことから、三角形要素を用いた安定化を施した有限要素式の各項は次のようになる。

$$\mathbf{A}_e = \frac{S^e}{12} \begin{bmatrix} v_{x1}^e b_1^e + v_{y1}^e c_1^e & v_{x2}^e b_1^e + v_{y2}^e c_1^e & v_{x3}^e b_1^e + v_{y3}^e c_1^e \\ v_{x1}^e b_2^e + v_{y1}^e c_2^e & v_{x2}^e b_2^e + v_{y2}^e c_2^e & v_{x3}^e b_2^e + v_{y3}^e c_2^e \\ v_{x1}^e b_3^e + v_{y1}^e c_3^e & v_{x2}^e b_3^e + v_{y2}^e c_3^e & v_{x3}^e b_3^e + v_{y3}^e c_3^e \end{bmatrix} + \frac{S^e}{12} \sum_{k=1}^3 \begin{bmatrix} v_{xk}^e b_1^e + v_{yk}^e c_1^e & v_{xk}^e b_2^e + v_{yk}^e c_2^e & v_{xk}^e b_3^e + v_{yk}^e c_3^e \\ v_{xk}^e b_2^e + v_{yk}^e c_2^e & v_{xk}^e b_2^e + v_{yk}^e c_2^e & v_{xk}^e b_2^e + v_{yk}^e c_2^e \\ v_{xk}^e b_3^e + v_{yk}^e c_3^e & v_{xk}^e b_3^e + v_{yk}^e c_3^e & v_{xk}^e b_3^e + v_{yk}^e c_3^e \end{bmatrix} \quad (5.2.24)$$

$$\mathbf{D}_e = S^e k \begin{bmatrix} b_1^e b_1^e + c_1^e c_1^e & b_1^e b_2^e + c_1^e c_2^e & b_1^e b_3^e + c_1^e c_3^e \\ b_2^e b_1^e + c_2^e c_1^e & b_2^e b_2^e + c_2^e c_2^e & b_2^e b_3^e + c_2^e c_3^e \\ b_3^e b_1^e + c_3^e c_1^e & b_3^e b_2^e + c_3^e c_2^e & b_3^e b_3^e + c_3^e c_3^e \end{bmatrix} \quad (5.2.25)$$

$$\begin{aligned} \mathbf{A}_{\delta_e} &= \frac{S^e \tau_e}{12} \sum_{\alpha=1}^2 \sum_{\beta=1}^2 \sum_{k=1}^3 v_{\alpha k}^e v_{\beta k}^e \begin{bmatrix} \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} \end{bmatrix} + \\ &\quad \frac{S^e \tau_e}{12} \sum_{\alpha=1}^2 \sum_{\beta=1}^2 \sum_{k=1}^3 \sum_{l=1}^3 v_{\alpha k}^e v_{\beta l}^e \begin{bmatrix} \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} \end{bmatrix} \\ &= \frac{S^e \tau_e}{12} \sum_{\alpha=1}^2 \sum_{\beta=1}^2 \sum_{k=1}^3 v_{\alpha k}^e \left\{ v_{\beta k}^e \begin{bmatrix} \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} \end{bmatrix} + \sum_{l=1}^3 v_{\beta l}^e \begin{bmatrix} \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_1^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_2^e}{\partial \beta} \\ \frac{\partial L_1^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_2^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} & \frac{\partial L_3^e}{\partial \alpha} \frac{\partial L_3^e}{\partial \beta} \end{bmatrix} \right\} \end{aligned} \quad (5.2.26)$$

$$\mathbf{D}_{\delta_e} = S^e \tau_{LSIC} \begin{bmatrix} b_1^e b_1^e + c_1^e c_1^e & b_1^e b_2^e + c_1^e c_2^e & b_1^e b_3^e + c_1^e c_3^e \\ b_2^e b_1^e + c_2^e c_1^e & b_2^e b_2^e + c_2^e c_2^e & b_2^e b_3^e + c_2^e c_3^e \\ b_3^e b_1^e + c_3^e c_1^e & b_3^e b_2^e + c_3^e c_2^e & b_3^e b_3^e + c_3^e c_3^e \end{bmatrix} \quad (5.2.27)$$

$$\mathbf{T}_e = \begin{Bmatrix} T_1^e \\ T_2^e \\ T_3^e \end{Bmatrix}, \quad \mathbf{F}_e = \frac{k g_2 h_{12}^e}{2} \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} \quad (5.2.28)$$

このように、三角形要素を用いて有限要素方程式を書くことができた。領域全体は直接剛性法により、全要素について足し合わせることで得られる。

三角形要素を適用することで、四角形要素とは異なり面積座標の公式を用いることで座標変換やガウス・ルジャンドル法を用いた数値積分を行わずに積分の計算まで行える。

第6章 熱の移流拡散方程式の解析

ここまでで、解析の対象とする場の方程式とそれを計算機で解くための有限要素式の導出ができた。この章では実際に計算機で解くことについて述べる。

6.1 問題の設定

解析に移る前にこの節で問題の設定やパラメータの設定などについてまとめておく。

6.1.1 場の方程式と有限要素方程式

本研究で扱う問題は二次元定常移流拡散問題で場の方程式は次のように表される。

$$\begin{cases} v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 & (\Omega \text{内}) \\ T = g_1 & (\Gamma_1 \text{上}), \quad \frac{\partial T}{\partial n} = g_2 & (\Gamma_2 \text{上}) \end{cases} \quad (6.1.1)$$

$$\quad (6.1.2)$$

この問題に SUPG 法の適用と衝撃捕捉項の付加を行い、安定化有限要素法の適用を行う。重み付き残差法を適用し、さらに弱形式を求めると以下の式となる。

$$\int_{\Omega} \left(v_x \frac{\partial \hat{T}}{\partial x} + v_y \frac{\partial \hat{T}}{\partial y} \right) w \Omega + k \int_{\Omega} \left(\frac{\partial \hat{T}}{\partial x} \frac{\partial w}{\partial x} + \frac{\partial \hat{T}}{\partial y} \frac{\partial w}{\partial y} \right) d\Omega = k \int_{\Gamma_2} g_2 w d\gamma \quad (6.1.3)$$

ある要素 e が n_{en} 角形一次要素である要素を使って要素分割を行うと、近似関数と近似形の重み関数 $\hat{T}, \hat{w}$ は次のようになる。

$$\hat{T} = \sum_{i=1}^{n_{en}} N_i^e T_i^e, \quad \hat{w} = \sum_{i=1}^{n_{en}} N_i^e w_i^e \quad (6.1.4)$$

本研究では、四角形一次要素と三角形一次要素での要素分割を想定するため n_{en} は 3 か 4 となる。ここで、安定化を施すため Petrov-Galerkin 法を適用し重み関数に次の式を用いる。

$$\tilde{w} = \hat{w} + \tau_e \mathbf{v} \cdot \nabla \hat{w} = \hat{w} + \tau_e v_i \frac{\partial \hat{w}}{\partial x_i} = \hat{w} + \tau_e \left(v_x \frac{\partial \hat{w}}{\partial x} + v_y \frac{\partial \hat{w}}{\partial y} \right) \quad (6.1.5)$$

これらの近似関数を代入し、さらに衝撃捕捉項を付加することで安定化を施した有限要素式が以下のように求まる。

$$(\mathbf{A}_e + \mathbf{D}_e + \mathbf{A}_{\delta_e} + \mathbf{D}_{\delta_e}) \mathbf{T}_e = \mathbf{F}_e \quad (6.1.6)$$

ここに $\mathbf{A}_e, \mathbf{D}_e, \mathbf{A}_{\delta_e}, \mathbf{D}_{\delta_e}, \mathbf{T}_e, \mathbf{F}_e$ はそれぞれ、移流行列、拡散行列、安定化行列、衝撃捕捉行列、温度ベクトル、熱フラックスベクトルである。これらの行列とベクトルの i, j 成分は以下で示される。なお、拡散行列のように、二階微分項からなる積分領域を行う行列を剛性行列という慣習がある。

$$\mathbf{A}_e = \int_{\Omega_e} N_j^e \left(v_x \frac{\partial N_i^e}{\partial x} + v_y \frac{\partial N_i^e}{\partial y} \right) d\Omega, \quad \mathbf{D}_e = k \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega \quad (6.1.7)$$

$$\mathbf{A}_{\delta_e} = \tau_e \int_{\Omega_e} \left(v_x \frac{\partial N_j^e}{\partial x} + v_y \frac{\partial N_j^e}{\partial y} \right) \left(v_x \frac{\partial N_i^e}{\partial x} + v_y \frac{\partial N_i^e}{\partial y} \right) d\Omega, \quad \mathbf{D}_{\delta_e} = \tau_{LSIC} \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega \quad (6.1.8)$$

$$\mathbf{T}_e = \mathbf{T}_i, \quad \mathbf{w}_e = \mathbf{w}_i \quad (6.1.9)$$

ここで流速 v_x, v_y は以下のような内挿近似式として示される。

$$v_x \approx \hat{v}_x = \sum_{i=1}^{nen} N_i^e v_{xi}^e, \quad v_y \approx \hat{v}_y = \sum_{i=1}^{nen} N_i^e v_{yi}^e \quad (6.1.10)$$

v_{xi}^e, v_{yi}^e はある要素 e の各節点 i での流速である。

$\mathbf{F}$ は隣接する要素間ではお互いに相殺されるため、要素境界辺での境界積分は、重ね合わせを行った段階で消去され、最終的に領域の境界 Γ のみの積分が残ることとなる。例えば、ある四角形一次要素 Ω_e の境界 Γ_{12} で自然境界条件が与えられている場合以下のようなになる。

$$\mathbf{F}_e = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ N_3^e \\ N_4^e \end{Bmatrix} d\gamma = kg_2 \int_{\Gamma_e} \begin{Bmatrix} N_1^e \\ N_2^e \\ 0 \\ 0 \end{Bmatrix} d\gamma \quad (6.1.11)$$

また、安定化項と衝撃捕捉項の安定化パラメータ τ_e, τ_{LSIC} はそれぞれ以下で示される。

$$\tau_e = \frac{h_e}{2|\mathbf{v}|} \tilde{\xi}(P_e), \quad \tilde{\xi}(P_e) = \coth P_e - \frac{1}{P_e} = \frac{1}{\tanh P_e} - \frac{1}{P_e}, \quad P_e = \frac{|\mathbf{v}|h_e}{2k} \quad (6.1.12)$$

$$\tau_{LSIC} = \frac{|\mathbf{v}|h_e}{2} \xi'(P_e), \quad \xi'(P_e) = \begin{cases} \frac{P_e}{3} & (P_e \leq 3) \\ 1 & (P_e > 3) \end{cases} \quad (6.1.13)$$

安定化パラメータ τ_e, τ_{LSIC} に登場する要素サイズ h_e は次のように決めた。

$$h_e = \begin{cases} 2 \left(\sum_{\alpha=1}^m \left| \frac{v_x}{|\mathbf{v}|} \frac{\partial N_\alpha^e}{\partial x} + \frac{v_y}{|\mathbf{v}|} \frac{\partial N_\alpha^e}{\partial y} \right| \right)^{-1} & (|\mathbf{v}| > 0.001) \\ \text{要素の各辺の長さの平均値} & (|\mathbf{v}| \leq 0.001) \end{cases} \quad (6.1.14)$$

6.1.2 解析対象の物性値

場の方程式にあわれるパラメータとして現れる温度伝導率 k に注目する。

k は $k = \frac{\kappa}{\rho c_p}$ [m²/s] で表される物理量である。このうち κ, ρ, c_p は物質によって異なる物性値である。したがって、このパラメータ k は物質によって異なる値であるといえる。地球上でもっとも埋蔵量が多いとされている金属の鉄に関する物性値を解析に用いるとする。この物性値のデータは文献 [7] 『理科年表 平成 22 (机上版)』より引用し、下記の表に記す。なお物性値の後ろのページ数は文献 [7] の参照ページとなっている。

表 6.1: 物性値

元素記号	Fe	単位	参照ページ
原子量	55.845	[g/mol]	[p.365]
密度 ρ	7.874	[10 ³ kg/m ⁻³] (20 °C)	[p.371]
熱伝導率	κ	[W·m ⁻¹ ·K ⁻¹]	[p.405]
定圧モル熱容量	c_p	[J·K ⁻¹ ·mol ⁻¹]	[p.488]

ここで熱伝導率と定圧モル熱容量は温度に依存し次の表で示される。

表 6.2: 熱伝導率 κ [W·m⁻¹·K⁻¹]

温度	t/[K]				
	173.25	273.15	373.15	573.15	973.15
κ	99	83.5	72	56	34

表 6.3: 定圧モル熱容量

温度	c_p [J·K ⁻¹ ·mol ⁻¹]				
	100K	200 K	298.15 K	400 K	600 K
c_p	12.05	21.46	24.97	27.4	32.1

本研究では、定圧モル熱容量は 298.15 K での値で他の温度でも一定とし、熱伝導率も 273.15 K での値を使い他の温度でも一定とした。なお、定圧モル熱容量を計算しやすいように mks 単位系に換算した、定圧比熱容量に直すと以下のようなになる。

$$24.97 \div \text{原子量 } 55.845 = 0.44713045 \left[\frac{\text{J}}{\text{g K}} \right] \approx 447.13 \left[\frac{\text{J}}{\text{kgK}} \right] \quad (6.1.15)$$

表 6.4: 入力データの物性値

元素記号	Fe	単位	値
熱伝導率	κ	$[\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}]$	83.5
密度	ρ	$[\text{kg}/\text{m}^3](20^\circ\text{C})$	7874
定圧比熱容量	c_p	$[\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$	447.13

これらのデータより、入力データに使う値は以下の表 6.4 に示した。
これらの物性値から、温度拡散率 k を求めると以下ようになる。

$$k = \frac{\kappa}{\rho c_p} = \frac{83.5}{7874 \times 447.13} = 2.3716 \times 10^{-5} [\text{m}^2/\text{s}] \quad (6.1.16)$$

6.2 解析システムの構築

解析システムの構築においては Fortran 言語を用いて fem2dh_steady_AD という名称の解析プログラムを作成した。
解析プログラムの作成にあたって、福井が文献 [11] で作成した fem2dh_steady_test という 2 次元定常拡散問題を解析するプログラムをベースに開発を行った。

解析の流れ

計算機を使って解析を行う場合、一般的に次の 3 つの処理を行う。

- プリプロセッサ（前処理）：入力データの作成
- ソルバ（メインプロセッサ）：計算により式を解く
- ポストプロセッサ（後処理）：結果の可視化

解析を行うために、定められた形式で入力データを用意することをプリプロセッサと呼んでいる。プリプロセッサから受け取ったデータを使って計算を行うことをソルバと呼んでいる。ソルバとは解くという意味であり、厳密には連立方程式を解く部分のことである。しかし、一般的には入力データを受け取って計算をして結果を出すまでの段階までをさす。（本来の言葉の定義からするとメインプロセッサと呼ぶのがふさわしい）ソルバから出た結果は数値の羅列で人間がそれを見て理解するのは困難である。そこで、その数値の羅列をグラフなどにして可視化することで結果がどうなったかを理解しやすくする処理を行う。この処理をポストプロセッサと呼んでいる。

本研究においてこれらの 3 つの処理をどう行ったかを示す。

プリプロセッサ プリプロセッサが必要な要件としては、要素の物性値、要素の形状要素の番号と要素を構成する節点の番号、節点の番号と座標値、節点の流速値、境界条件がある。

当初はこれらを手入力していた。しかし、要素数を増やして解析を行う際に入力に時間がかかりすぎる。そのため、melement4 という長方形の領域全体で均一な四角形要素を作成するためのプログラムを作った。また melement3 という長方形領域全体で均一な直角三角形を要素を作成するためのプログラムを作った。それによって、要素の番号と要素を構成する節点の番号および、節点の番号と座標を設定する。さらに、境界条件については領域の各辺について辺ごとに値を設定するようにもした。

ソルバ ソルバとしては、有限要素法を用いた解析プログラムの fem2dh_steady_AD により多元連立一次方程式を形成し、それをガウスの消去法で解く。その結果を節点番号や節点の座標および温度として出力する。

ポストプロセッサ ポストプロセッサとしては、ソルバで出力した座標と温度のデータから、グラフ描画ソフトである gnuplot を使い 3 次元のグラフを描画する。gnuplot の使い方は文献 [10] を参考にした。

または、もともと fem2dh_steady_test に実装されていた領域の分割図を表示するサブルーチンである d.mesh および、同じ温度の地点を同色で表示するサブルーチン conter により出力される結果をそのままポストプロセッサとして採用した。

このような構成で解析システムを構築した。

6.3 プログラムでの解析と結果

この節では、実際にプログラム fem2dh_steady_AD を実行して得られた結果について述べる。

6.3.1 4要素での解析

まず、この項では安定化を計算に組み込まずに計算を行い、解の不安定性の以前の問題として、きちんとプログラムができているかの確認を行う。

次の図 6.1 に示される四角形領域を対象とし、領域を四角形要素（正方形要素）4 つに分割して解析を行った。

なお、図 6.1 において丸数字は節点番号を表し、[] で囲まれた数字は要素番号を表す。この領域を対象にして、作成したプログラムの妥当性 (validation) の確認を行った。領域全体で流速を 0 とすると、実質的に拡散方程式と同じである。そこで次のような表 6.5 に示される条件で解析を行った。

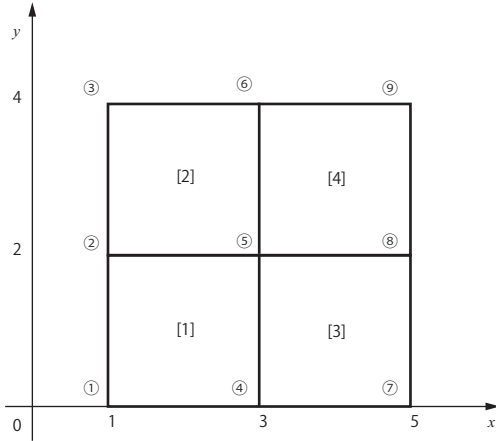


図 6.1: 4 要素での解析領域

表 6.5: 入力条件	
パラメータ	値
基本境界条件	$T(1, y) = 300, T(5, y) = 200 \quad (0 \leq y \leq 2)$
自然境界条件	$\frac{\partial T(x, y)}{\partial y} = 0 \quad (y = 0, 2)$
流速 v_x, v_y	全節点で 0.0

一次元定常拡散問題との比較

拡散方程式での理論解は場の方程式

$$\kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 \quad (6.3.1)$$

を解くことで求まる。

図 6.1 の場合では、自然境界条件と基本境界条件から、 x 方向に一様に熱が伝わると考えられる。つまり、各 x での値での y 方向の温度は等しくなると考えられ、 y 方向への温度勾配の $\frac{\partial T}{\partial y}$ は 0 になり、 $\frac{\partial^2 T}{\partial y^2}$ も 0 になると考えられる。従って図 6.1 の場合においては温度を表す関数 T は x にのみ依存すると考えることができる。そのため、一次元定常拡散問題

$$\kappa \left(\frac{d^2 T}{dx^2} \right) = 0 \quad (6.3.2)$$

となり、これを解くと

$$T = Ax + B (A, B \text{ は積分定数}) \quad (6.3.3)$$

となる。これに基本境界条件を当てはめると、 $T(1) = 400, T(5) = 200$ より

$$300 = A + B$$

$$200 = 5A + B$$

となる。これらを連立させて解くと、 $A = -25, B = 325$ となることがわかる。よって、理論解は

$$T = -25x + 325 \quad (6.3.4)$$

であることがわかる。

上記の結果を使って、プログラムを実行して得られた結果と理論解を表 6.6 にまとめた。この結果から、理論解と値が一致しているため拡散の問題を解くことはできているといえる。拡散問題を解析するプログラムをベースに作成したため当然と言える結果である。

表 6.6: 拡散問題として解析した結果

節点番号	座標 (x,y)	解析結果 [K]	理論解 [K]
1	1,0	300.00	300.00
2	1,2	300.00	300.00
3	1,4	300.00	300.00
4	3,0	250.00	250.00
5	3,2	250.00	250.00
6	3,4	250.00	250.00
7	5,0	200.00	200.00
8	5,2	200.00	200.00
9	5,4	200.00	200.00

表 6.7: 移流拡散問題としての解析結果

節点番号	座標 (x,y)	解析結果 [K]
1	1,0	300.00
2	1,2	300.00
3	1,4	300.00
4	3,0	434.08
5	3,2	185.89
6	3,4	90.76
7	5,0	200.00
8	5,2	200.00
9	5,4	200.00

移流拡散問題として解いた結果

続いて、流速を $v_x = v_y = 1.0$ として解析を行った結果を載せる。移流が入った問題は非線形であるため、拡散方程式のように理論解を求めるのは困難である。それゆえに、比較となる理論解は求めている。しかし、与えられた境界条件と表 6.6 に示される拡散問題の結果から値が極端に異なっているかどうかの確認ができ、移流拡散の問題が解けているかの確認を目的とした。

この表 6.7 の結果からわかることとして、境界条件の与えられていない節点 4,5,6 において値が急激に変化しており、領域の両端に与えた境界条件よりも値が大きくなったり、小さくなるというような拡散問題の結果からも考えにくい解析結果となっている。したがって、この結果からプログラムが移流拡散問題を解けていないではないかという疑念が生じた。

この結果から、次項においてもさらに条件を変更して解析を行い、他の妥当と言える結果と照らし合わせて、本研究で作成したプログラムはどこまでの計算を行え、どこに問題点があるかを探った。

6.3.2 一次元定常移流拡散問題での検証

移流拡散において流速が 0 のときである前項で扱った拡散の問題については、理論解と比較して妥当な結果が得られた。そこで、この項では移流拡散問題の理論解との比較を考える。

本研究で解析の対象である二次元定常移流拡散方程式は以下で示される。

$$v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 \quad (6.3.5)$$

この式の理論解をこのまま求めることは困難である。しかし、4.1 節で扱ったように一次元問題になれば理論解を求めることは可能である。

そこで、境界条件と流速を一次元問題と同じになるように定めることで、一次元定常移流拡散問題の理論解と解析の結果を比較した。

一次元定常移流拡散問題の理論解

以下で示される境界条件を持つ一次元定常移流拡散問題について考える。

$$\begin{cases} v_x \frac{dT}{dx} - k \frac{d^2 T}{dx^2} = 0 & (0 < x < L) \\ T(0) = 1, T(L) = 0 \end{cases} \quad (6.3.6)$$

$$(6.3.7)$$

この問題の特殊解は式 (4.1.12) で既に導出した次の式である。

$$T = 1 - \frac{1 - e^{\frac{v_x}{k}x}}{1 - e^{\frac{v_x}{k}L}} = 1 - \frac{e^{\frac{v_x}{k}x} - 1}{e^{\frac{v_x}{k}L} - 1} \quad (6.3.8)$$

この式において、 $L=1$ でさらに $v_x = 1.0, k = 1.0$ とすると理論解は次のようになる。

$$T = 1 - \frac{1 - e^x}{1 - e} \quad (6.3.9)$$

x 方向に対する一次元問題との比較

式 (6.3.9) より、一次元移流拡散問題の理論解が求まった。この理論解との比較を行うために、本研究で作成した解析プログラムでも表 6.8 に示した一次元定常移流拡散問題となるような条件で解析を行った。要素分割は図 6.2 に示したような四角形（正方形）要素 400 個による分割で行った。

表 6.8: 一次元移流拡散問題での解析条件

流速	全節点で $v_x = 1.0, v_y = 0.0$
温度拡散率	$k=1.0$
解析領域	1×1 の正方領域
要素分割	四角形要素 400 個
基本境界条件	$T(0,y)=1.0, T(1,y)=0.0 \quad (0 \leq y \leq 1)$

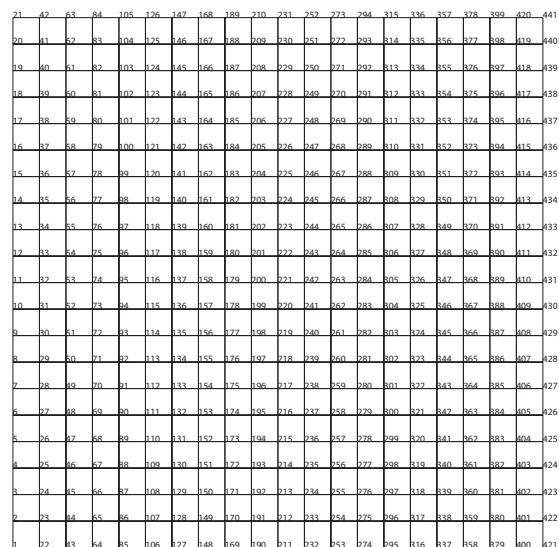


図 6.2: 四角形要素 400 個での要素分割図

グラフ描画ソフトである gnuplot で、これらの一次元定常移流拡散問題の理論解と同じ条件で解析を行った結果を図 6.3 に載せた。

図 6.3 において上側の曲線が理論解で下側の曲線が数値解である。一次元問題の比較において流速が逆方向に働いているような結果が得られた。そこで、流速が逆に働いているのではないかと疑い、流速を逆にした結果を図 6.4 に示した。

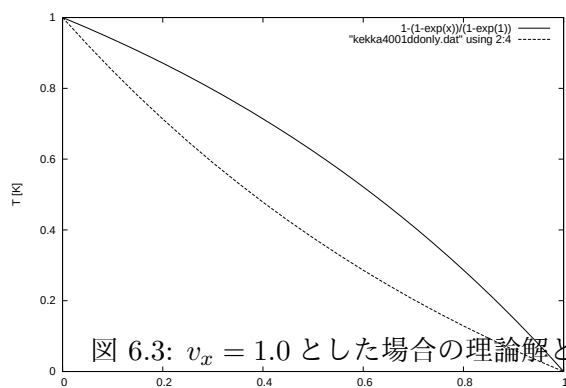


図 6.3: $v_x = 1.0$ とした場合の理論解との比較

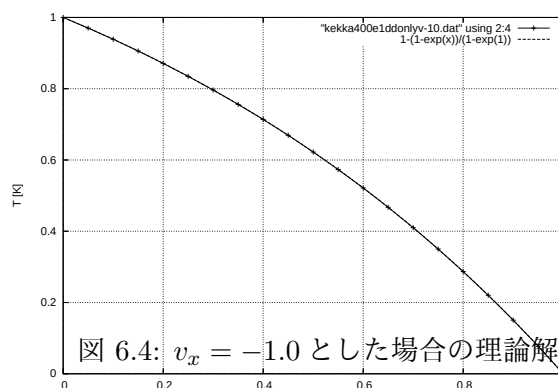


図 6.4: $v_x = -1.0$ とした場合の理論解との比較

図 6.4 の結果から、流速を逆にすると厳密解と結果が一致していることがわかる。このことから、本研究で作成したプログラムでは x 方向については流速が逆方向に作用しているということがわかった。

y 方向の一次元問題との比較

同様に、今度は y 方向の一次元問題との比較を考える。境界条件として、下端 ($y = 0$) に $T = 1.0$ 、上端 ($y = 1.0$) に $T = 0$ を与えて流速を $v_y = 1.0$ 及び -1.0 とした。 $v_y = 1.0$ とした結果を図 6.5 に、 $v_y = -1.0$ とした結果を図 6.6 に載せた。

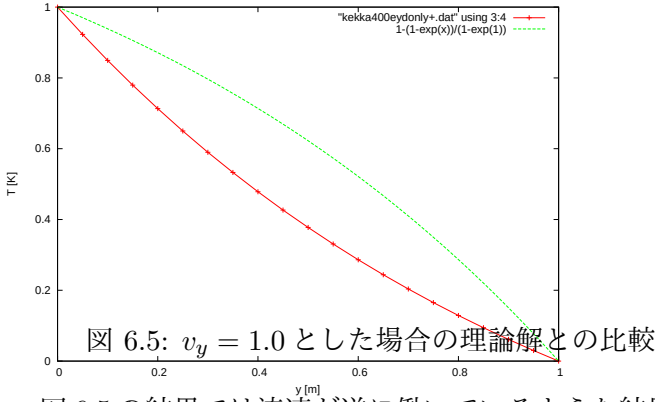


図 6.5: $v_y = 1.0$ とした場合の理論解との比較

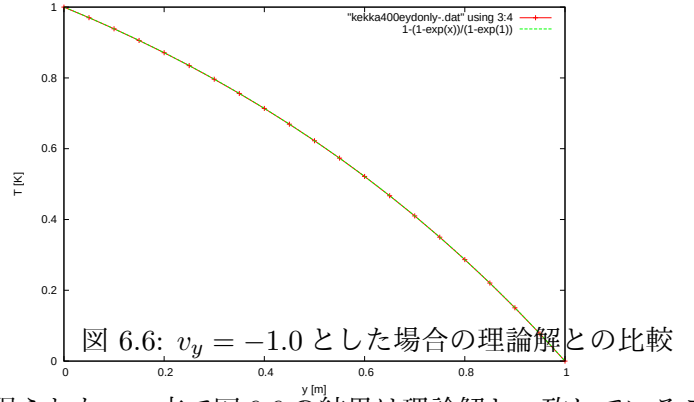


図 6.6: $v_y = -1.0$ とした場合の理論解との比較

図 6.5 の結果では流速が逆に働いているような結果が得られた。一方で図 6.6 の結果は理論解と一致していることがわかる。これらの結果から、 y 方向の一次元問題として解こうとすると、 x 方向と同様に流速が逆に働いているような結果が得られた。

ここまでの一次元定常移流拡散問題の理論解との比較から、 x 方向、 y 方向の両方で設定した流速が逆向きに作用しているような結果が得られた。プログラムの確認を行ったが、誤りを見つけることができなかった。流速を逆にすることで理論解と一致するという結果は、根本的な計算ができていなければ得られない結果だと判断できる。したがって、基本的な計算はできていると判断した。

このことを踏まえて、一次元定常移流拡散問題を二次元定常移流拡散問題に拡張して解析結果の検討を行い、計算ができていくかどうかの確認を行った。

6.3.3 二次元定常移流拡散問題での検証

結果の予想できるものとの比較

一次元問題の比較で、流速が逆に働いているものの計算はできているような結果が得られた。そこで、結果の予想できるような条件で実行して予想通りとなるか結果を比較する。ここで、予想される結果とは温度の分布が流れに伴って流下方向へ広がる結果である。

解析条件は境界条件を領域の 4 辺に $T = 0$ [K]、領域の中心である節点番号 221 に $T = T(0.5, 0.5) = 1$ [K] とし、全節点で $v_x = -10, v_y = 0$ [m/s] とした。境界条件と流速以外は表 6.8 と同じ領域、温度拡散率 k とした。それを図にしたものを図 6.7 に示した。この条件で予想される結果は図 6.8 のように流れの下流方向に分布が広がる状態である。

この条件で解析して得られた結果をルーチン conter で可視化処理したものとして図 6.9 に示した。この結果から、図 6.8 のように予想される結果が得られた。この結果からだと計算はできているように判断できる。

4 章で扱ったように、移流が大きくなると解が不安定になる問題が生じるはずである。そこで、図 6.9 と流速の大きさのみ変化させて他は同じ条件で解析を行った場合の結果を見ていく。

流速を $v_x = -50, -500$ [m/s] とした結果をそれぞれ図 6.10 及び図 6.11 に示した。

図 6.10 の結果では $T = 1$ [K] の境界条件を与えた地点付近の上流側に値がマイナスとなる箇所が存在している。図 6.11 の結果では、領域内 $T = 1$ [K] の境界条件を与えた地点での x 軸に沿って値が振動し、領域内で値がマイナスとなる箇所が図 6.10 よりも増加していることがわかる。この領域内で値がマイナスとなる現象は $v_x = -30$ [m/s] 付近から観測された。このことから流速が大きくなると解が不安定になることがわかる。

ここまでの解析においては、解の不安定性が問題とならなかったため、安定化を行わずに解の結果がどうなるか論じた。この図 6.10 及び図 6.11 で初めて本研究の解析結果では解の不安定性に遭遇した。この問題に対して 4 章で論じた安定化有限要素法として SUPG 法の適用と衝撃捕捉項を付加することで対処することとし、安定化項と衝撃捕捉項 $A_{\delta_e}, D_{\delta_e}$ を有限要素式に組み込んで計算を行う。

不安定性が顕著である図 6.11 の条件に対して、安定化を施して解析を行った結果を図 6.12 に示した。この図 6.12 の結果から、解が安定した結果が得られた。そのため、安定化の計算が適切に行われていることが判断できる。

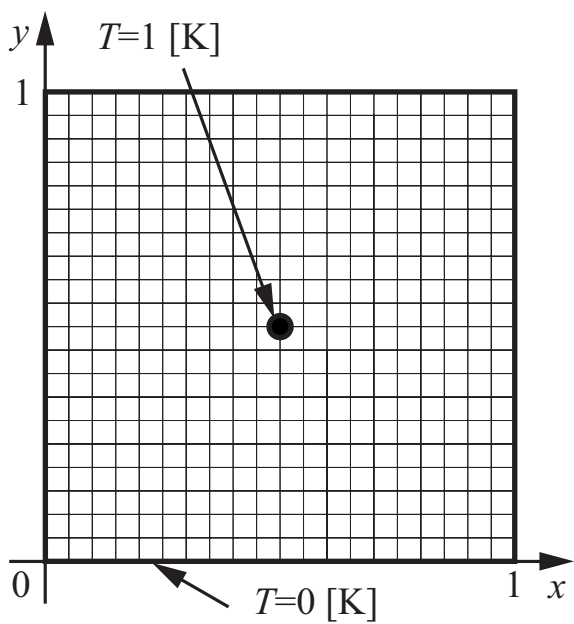


図 6.7: 解析条件

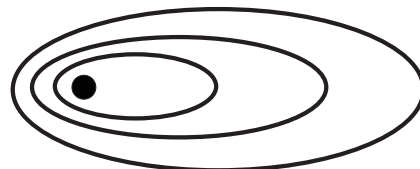


図 6.8: 予想される結果

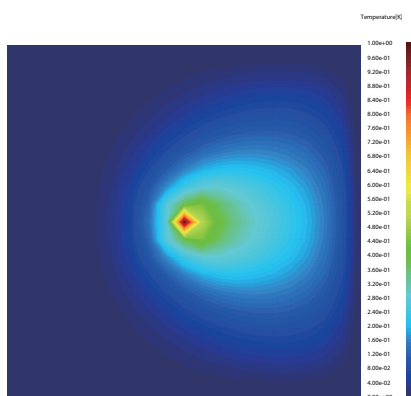


図 6.9: 流速 $v_x = -10$ [m/s] とした結果

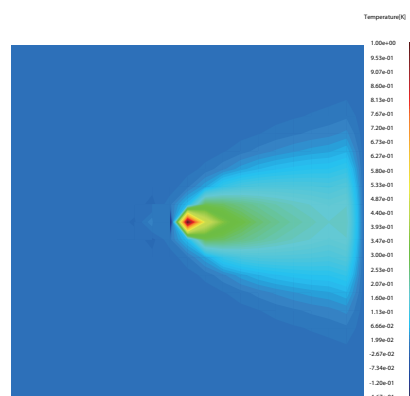


図 6.10: 流速 $v_x = -50$ [m/s] とした結果

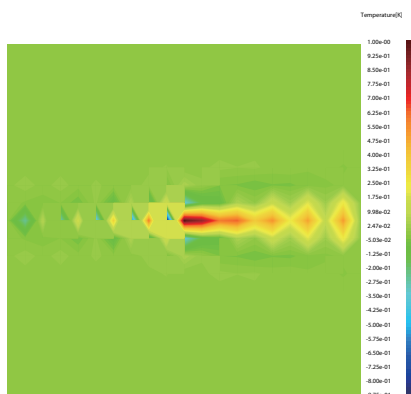


図 6.11: 流速 $v_x = -500$ [m/s] とした結果

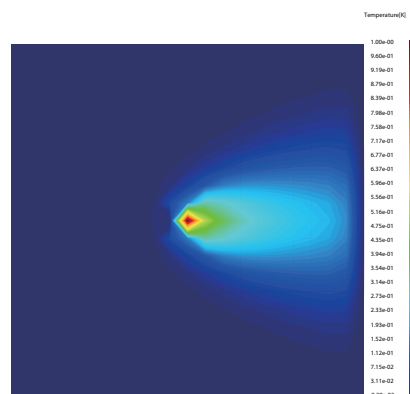
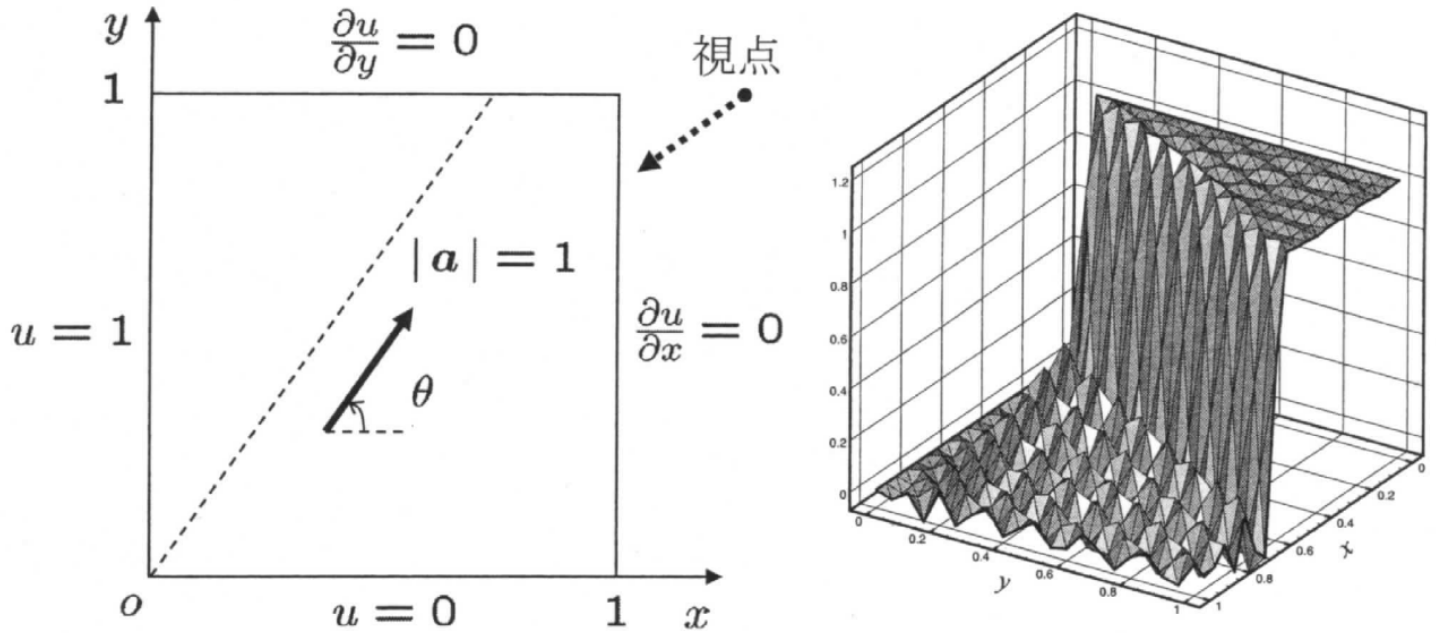


図 6.12: $v_x = -500$ [m/s] に安定化を施した結果

文献との比較

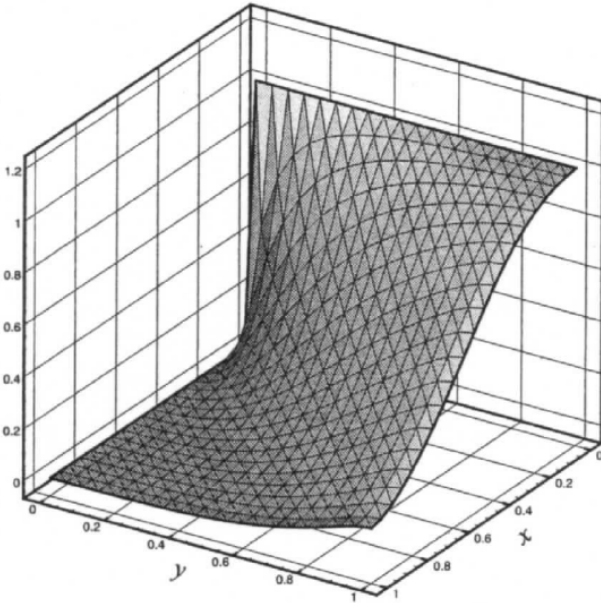
前項での検証はあくまで予想通りとなるかどうかだけで、厳密にあっていないかどうかの検証はできない。そこで、本項では他の文献で示されている結果を引用し、同じ条件で本研究の解析プログラムで解析を行い同じ結果が得られるかどうかの確認を行い、本研究で開発した解析プログラムの妥当性を確かめた。

比較に当たっては文献 [4, p.53] 『続・有限要素法による流れのシミュレーション』に示されている結果を用いた。文献 [4] で用いられたの解析領域及び解析結果を図 6.13 に引用して掲載した。

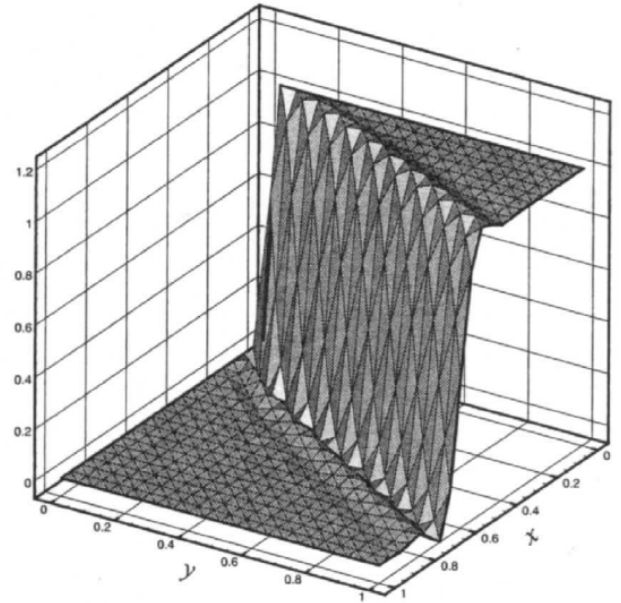


(a) 斜め移流問題の境界条件

(b) Galerkin 法



(c) Upwind Galerkin 法



(d) SUPG 法

図 6.13: 文献の解析領域と解析結果 (文献 [4, p.53] 『続・有限要素法による流れのシミュレーション』より引用)

なお、この文献では安定化の影響を検証するために (b)Galerkin 法で解析したもの、(c)Upwind Galerkin 法で解析したもの、(d)SUPG 法で解析したものの 3 パターンの解析結果が示されている。本研究では Upwind Galerkin 法での解析は対応していないため、比較を行うのは (b)Galerkin 法と (d)SUPG 法とに限られる。

図 6.13 において、 $\mathbf{a}$ は流速ベクトルを表し、流速は $\theta = 60^\circ$ 方向に流れている。つまり、流速は $v_x = 0.5$, $v_y = 0.5\sqrt{3}$ である。また、 u は未知関数を表し、本研究の T に該当する。なお、文献 [4] では拡散係数 $k = 1.0 \times 10^{-6}$ 、要素分割は 20×20 の四角形要素を三角形に割って形成された形態の三角形要素 800 個による要素分割で解析が行われている。

本研究では四角形要素での解析を行っている。そのため、解析条件として 20×20 の四角形要素を用いた。それ以外の条件は同じにした。前節の結果より、流速が逆に作用しているような結果がでたので、流速をマイナス方向にし

て解析を行った。

文献 [4] との比較のために本研究で解析を行う際の解析条件を表 6.9 にまとめた。

表 6.9: 文献と比較するための解析条件

領域	1×1 の正方領域
要素分割	四角形要素 400 個
基本境界条件	$T(0, y) = T(x, 0) = 1.0 \ (0 \leq x, y \leq 1.0)$
自然境界条件	$\frac{\partial T(1.0, y)}{\partial y} = \frac{\partial T(x, 1.0)}{\partial x} = 0 \ (0 \leq x, y \leq 1.0)$
流速	$v_x = -0.5, v_y = -0.5\sqrt{3}, \mathbf{v} = 1$
拡散係数	$k = 10^{-6}$

解析結果は各項のあるなしでどのような影響が出るか確認するために、拡散項のみ、移流項と拡散項、移流項と拡散項と安定化項、移流項と拡散項と安定化項と衝撃捕捉項の 4 パターンをそれぞれ図 6.14～図 6.17 に示した。

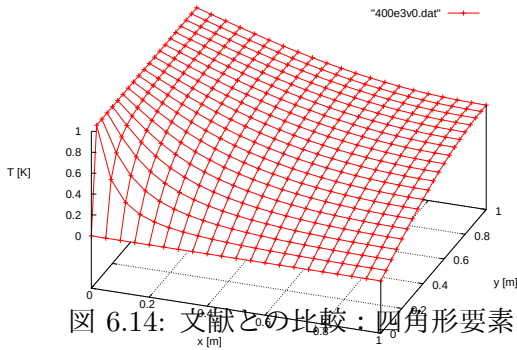


図 6.14: 文献との比較：四角形要素, 拡散項のみ

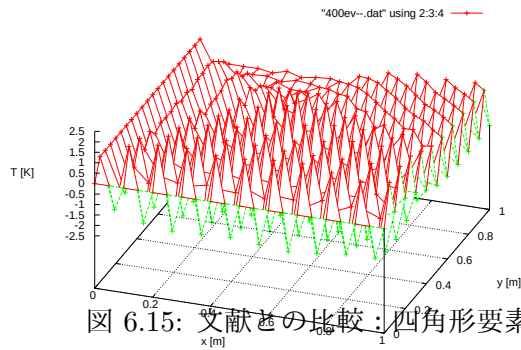


図 6.15: 文献との比較：四角形要素, 移流拡散

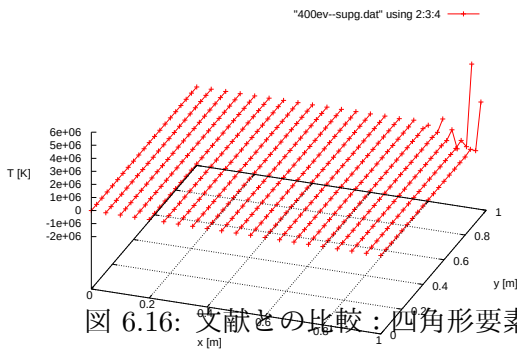


図 6.16: 文献との比較：四角形要素, SUPG 法

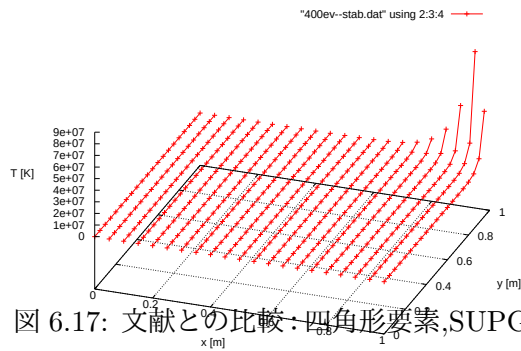


図 6.17: 文献との比較：四角形要素, SUPG 法 + 衝撃捕捉項

これらの結果と文献の結果である図 6.13 との対応関係を示す。本研究で解析した結果である図 6.15 と図 6.16 はそれぞれ文献の図 6.13 の (b)Galerkin 法, (d)SUPG 法と対応すると考えられる。

図 6.13 に示される文献の結果をみると、境界条件値が流れによって領域に広がっていることが読み取ることができる。それに対して、図 6.15～図 6.17 に示される本研究での解析結果はその様子が見受けられない。文献の図 6.13 の (b)Galerkin 法と本研究での解析結果である図 6.15 を比較すると、移流項が入ることで領域全体で数値振動が発生することは程度の差はあるものの共通の結果として得られた。しかし、振動の大きさが文献 [4] よりも大きい。また、前述の境界条件の不連続性の分布が維持されていない。

続いて、文献の図 6.13 の (d)SUPG 法 と図 6.16 の比較を行う。安定化が施されることで解の振動が収まるはずであり、文献の結果がそれを示している。しかし、図 6.16 の結果では安定化項が入ることで振動が抑えられるべきところが逆にさらなる数値振動を引き起こしている。本研究で行ったこの解析結果は文献 [4, p.53] の結果と異なり、四角形要素で解析を行った。そのため、要素分割の違いがこのような結果をもたらしたとも考えることができる。しかし、要素分割が異なる場合においても同じ問題を解析しているため、計算の誤差ともいえるような結果の違いは想定されるものの、このような解が振動してしまうような違いを与えるとは考えにくい。この結果から、本研究で開発したプログラムの計算では二次元定常移流拡散問題の解析までは行えないと判断できる。

自然境界条件として断熱条件 $\partial T(1, y)/\partial x = \partial T(x, 1)/\partial y = 0$ ($x = y = 0 \sim 1$) を与えているにも関わらず、図 6.16 及び図 6.17 では、 $x = y = 1$ 付近で急激な勾配が発生しており、この条件を満たしていないような結果となっている。よって、自然境界条件の処理がうまくいっていないといえる。しかし、プログラムの該当箇所を確認を行ったがそのような誤りは見つからなかった。

三角形要素での解析

本研究では、当初四角形要素のみを領域の要素分割として想定した解析を行っていた。しかし、図 6.13 に示される文献と同じ条件で比較をする上で文献と同じ三角形要素での解析が必要と感じた。そこで三角形要素を使った解析も実装した。

まず、二次元定常移流拡散問題を解析する前の段階で三角形要素での解析が妥当であるかの検証を行う。検証には前項でも行った一次元定常移流拡散問題の理論解との比較を行う。表 6.8 に示される解析条件のうち、要素分割のみが三角形要素 800 個となる点を除けば全く同じ条件である。

三角形要素で解析する際の領域の要素分割図を図 6.10 に示した。

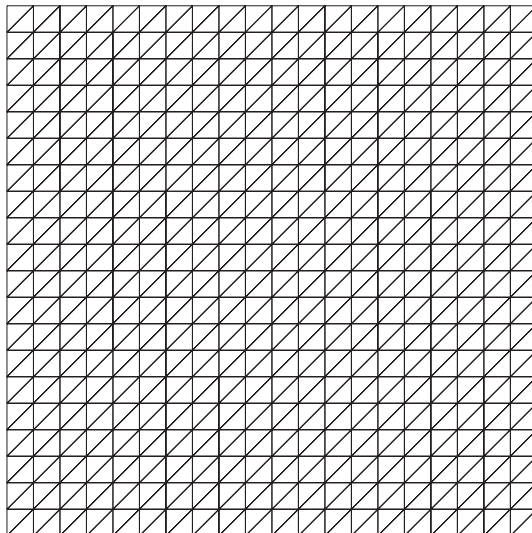


表 6.10: 三角形要素 800 個での要素分割図

x 方向の一次元問題として解いて、流速を $v_x = 1.0$ として解析した結果を図 6.18 に、流速を $v_x = -1.0$ として解析した結果を図 6.19 に示した。

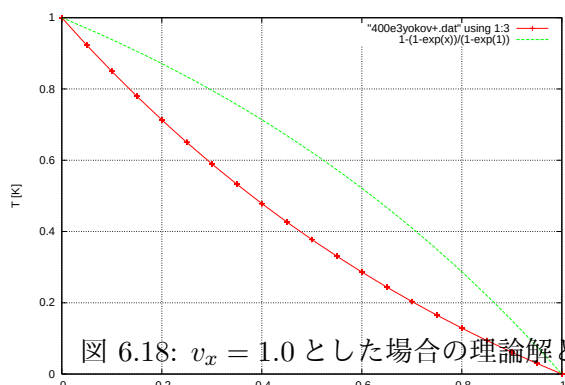


図 6.18: $v_x = 1.0$ とした場合の理論解との比較

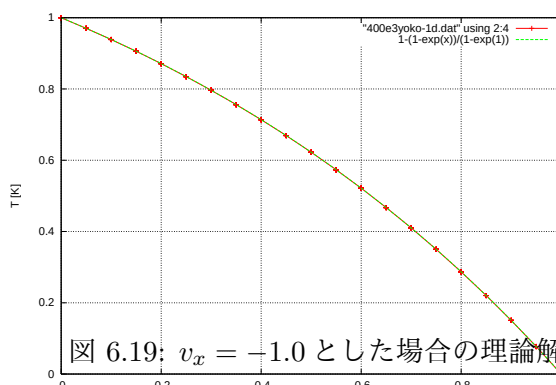


図 6.19: $v_x = -1.0$ とした場合の理論解との比較

これらの図 6.18 および図 6.19 の結果から四角形要素での解析と同様に x 方向について流速が逆転する現象がみられた。同様にして、 y 方向の一次元問題として解いて、流速を $v_y = 1.0$ として解析した結果を図 6.20 に、流速を $v_y = -1.0$ として解析した結果を図 6.21 に示した。

これらの図 6.20 および図 6.21 の結果から y 方向の一次元問題として解く場合も流速の逆転現象がみられた。これらの図 6.19 及び図 6.21 の解析結果は四角形要素での解析と異なり、データをみると理論解と完全に一致するわけではなくわずかな誤差を生じている。しかしながら、目視で違いを判別することが困難であるため計算誤差の許容範囲内と判断した。したがって、流速の逆転現象を考慮すると計算ができていると判断した。

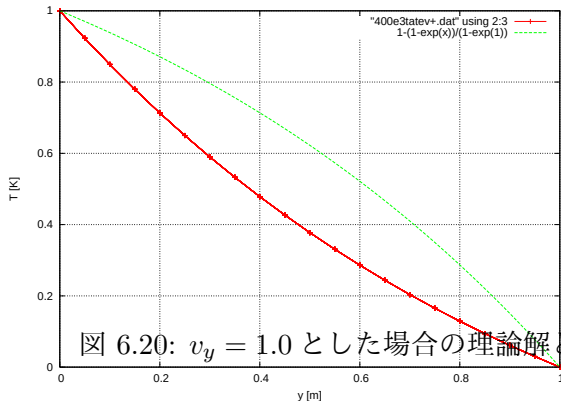


図 6.20: $v_y = 1.0$ とした場合の理論解との比較

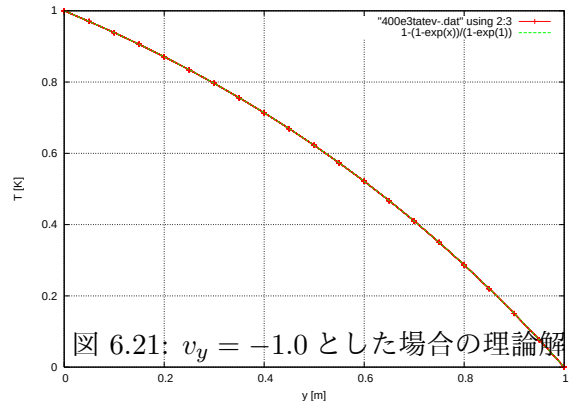


図 6.21: $v_y = -1.0$ とした場合の理論解との比較

そのため、三角形要素でも図 6.13 に示される文献の条件で解析を行い、流速の逆転現象を除くプログラムの妥当性を検証した。四角形要素での解析に使った表 6.9 に示した条件うち、要素が三角形要素 800 個に変更された以外は表 6.9 と同じである。要素分割図は一次元定常移流拡散問題の解析と同じ図 6.10 である。四角形要素で解析したときと同様に文献の解析条件で拡散項のみ、拡散項と移流項、SUPG 法を適用した場合、SUPG 法に衝撃捕捉項を加えた場合の 4 パターンで解析を行った。その結果を図 6.22～図 6.25 に示した。

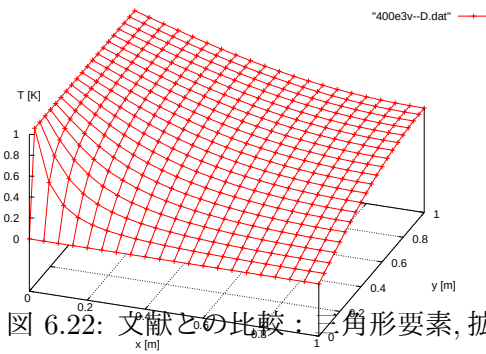


図 6.22: 文献との比較：三角形要素, 拡散項のみ

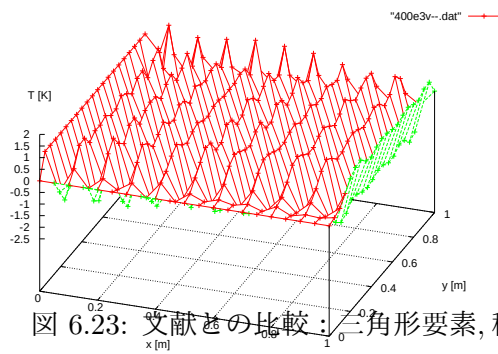


図 6.23: 文献との比較：三角形要素, 移流拡散

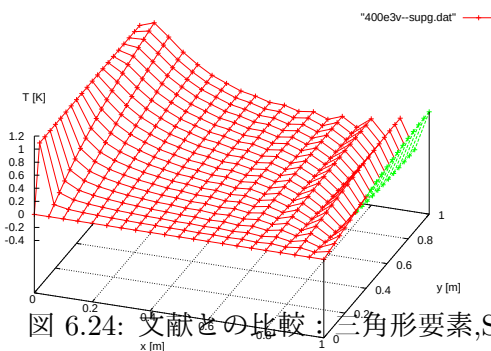


図 6.24: 文献との比較：三角形要素, SUPG 法

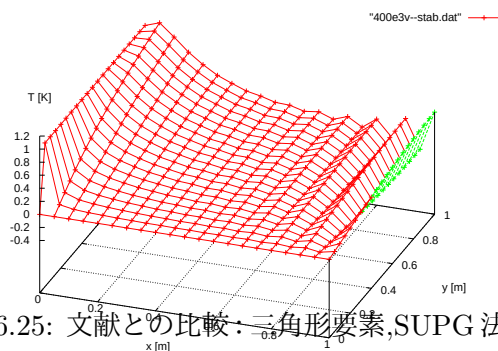


図 6.25: 文献との比較：三角形要素, SUPG 法 + 衝撃捕捉項

これらの三角形要素を用いた解析と文献の図 6.13 対応関係を示す。本研究で三角形要素を用いた解析結果である図 6.23 と図 6.24 がそれぞれ文献の解析結果である図 6.13 の (b)Galerkin 法および (d)SUPG 法に対応していると考えられる。

これらの対応を照らし合わせてみても、本研究での解析結果は文献 [4] の解析結果と異なっていることが分かる。図 6.23 を図 6.13 の (b)Galerkin 法と比較すると、領域全体で解が振動しているのは共通である。しかし、その振動の大きさが異なっており、本研究で得られた結果は文献の結果よりも振動が大きくなっている。図 6.24 と図 6.13 の (d)SUPG 法と比較する。本研究で得られた結果では、安定化を施す前の図 6.23 よりも振動が収まっており安定化の効果があったといえる。しかし、局所的に振動が残っているため安定化が不完全であるといえる。文献で示された境界条件値が流れによって領域全体に広がっているという現象も見受けられず、温度分布も異なっている。

さらに、本研究で四角形要素を用いた解析結果である図 6.14～図 6.17 と三角形要素での解析結果である図 6.22～図 6.25 との比較をする。拡散項のみの場合である図 6.14 と図 6.22 は同じ結果となった。残りの安定化を施した結

果同士を比べると、四角形要素での結果である図??及び図??では安定化の効果は見られず逆に振動が大きくなった。三角形要素での結果では図 6.24 及び図 6.25 では安定化の効果がみられる結果となった。また三角形要素での解析結果のうち、図 6.24 と図 6.25 を見比べると、衝撃捕捉項の影響が無いような結果となっている。

これらの結果から、三角形要素で文献と同じ条件で解析を行っても妥当な結果が得られず、本研究で開発したプログラムでは二次元定常移流拡散問題は解くことができないと判断できる。

他の解析プログラムでの解析結果

文献 [4] に掲載されている図 6.13 の結果をもとに比較を行ってきた。しかし、文献 [4] の結果がそもそも妥当なのかどうかという問題がある。図 6.13 の右上の (b) Galerkin 法については安定化を施さずに解いた結果であるが、本当に領域全体でこのような細かな数値振動が生じるのかという点が疑わしい。また同じ図の (C) SUPG 法での結果についても、このように解の不連続性が保たれたままの状態では解析結果がでることが妥当かどうか検証するすべがない。

そこで、同じ文献 [4, p.35] 内で紹介されている有限要素法を使ったフリーの解析ソフトである『FreeFEM++』[14] を用いて図 6.13 の条件で解析を行い、文献 [4] 及び本研究の解析プログラムの結果との比較を行う。

『FreeFEM++』の特徴は、ソースコードに領域の定義と境界条件、偏微分方程式の弱形式を入力することで、自動で要素分割を行い結果を出力することである。そのため、比較的簡単に有限要素法による数値解析が行える。

なお『FreeFEM++』のマニュアルは基本的に英語であるため、操作方法を理解するのが困難であった。そこで、マニュアルを一部日本語に翻訳した文献 [15] や斎藤宣一の講義ノート [16] を参考に『FreeFEM++』を用いた。

本研究の拡散の結果と同じになるか確認するために、流速を 0 として拡散の問題として解いた結果を図??に示した。文献の結果である図 6.13 と同じ条件で解析を行うために、表 6.9 のうち要素分割のみ三角形要素である条件で解析を行った結果を図 6.27 に掲載した。

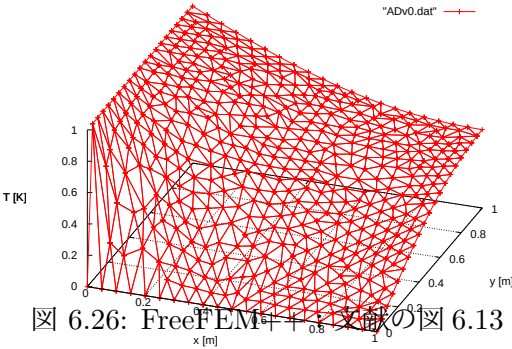


図 6.26: FreeFEM++：文献の図 6.13 の条件で流速 0

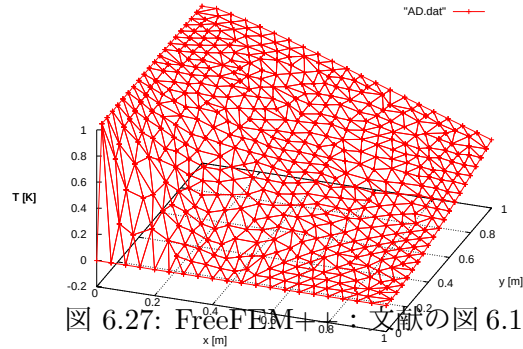


図 6.27: FreeFEM++：文献の図 6.13 と同じ条件

図 6.27 の結果を得るために使用した『FreeFEM++』のソースコードをリスト 6.1 に示す。なお、図 6.26 の結果はリスト 6.1 の 1 行目で行っている流速の設定で $v_x=0.0, v_y=0.0$ とすることで得られる。

リスト 6.1: 『FreeFEM++』を使って図 6.27 の結果を出力する時のソースコード

```
1 real vx=0.5,vy=0.5*sqrt(vx);
2 real k=1.0/10^6;
3
4 border a0 (t = 0, 1) { x = t; y = 0; }
5 border a1 (t = 0, 1) { x = 1; y = t; }
6 border a2 (t = 0, 1) { x = 1-t; y = 1; }
7 border a3 (t = 0, 1) { x = 0; y = 1-t; }
8
9 mesh Th = buildmesh (a0(10) + a1(10)+a2(10)+a3(10));
10 fespace Vh (Th, P1);
11 Vh u, v;
12
13 solve Laplace (u,v) = int2d (Th) (k*dx(u)*dx(v) + k*dy(u)*dy(v))
14   +int1d(Th) (vx*dx(u)*v+vy*dy(u)*v)
15   +on (a3, u = 1)+on (a0, u = 0);
16
17 plot (u);
18 //savemesh(Th,"Th.msh") ;
19 {
20   ofstream ug("u-g.txt");
21   for (int i=0; i<Th.nt; i++) {
22     for (int j=0; j<3; j++) {
23       ug << Th[i][j].x << " " << Th[i][j].y << " " << u[][Vh(i,j)]<<endl;
24     }
25     ug << Th[i][0].x << " " << Th[i][0].y << " " << u[][Vh(i,0)]<<"\n\n\n";
26   }
27 }
```

この FreeFEM++で解析を行った結果と文献及び本研究で行った結果の図の対応関係を示す。

FreeFEM++で解析を行った結果の図 6.26 は四角形要素での解析結果の図 6.14 及び三角形要素での解析結果の図 6.22 に対応する。また同様に FreeFEM++で解析を行った結果の図 6.27 は文献の解析結果の図 6.13 の (b)Galerkin 法と四角形要素での解析結果の図 6.15 及び三角形要素での解析結果の図 6.23 に対応する。

図 6.26 の結果の対応をみると本研究での解析結果である図 6.14 及び図 6.22 と同一であるような結果が得られた。このことから拡散問題として解く場合は本研究で作ったプログラムは妥当であるといえる。

しかし、図 6.27 の結果をみると文献の結果とも本研究で作成したものとも異なる結果が得られた。この中で、最も図 6.27 と結果が近いのが文献の図 6.13 の (c)Upwind Galerkin 法である。しかし、FreeFEM++での解析結果は同じ図 6.13 の (b) と対応することが予想されていた。そのため、この結果は想定外であった。

他の解析プログラムである FreeFEM++を使って得られた結果は、本研究で得られた結果でも文献の結果とも異なるものを示した。したがって、文献の結果である図 6.13 が妥当であるという判断はできない。また、FreeFEM++の操作方法が誤っていることもあり得る。しかし、文献と FreeFEM++の両者が示した結果と異なる結果を示した本研究で開発したプログラムの解析結果はやはり妥当でないといえる。

第7章 結言

7.1 結果

本研究で行ったことと得られた結果について述べる。本研究においては、熱の移流拡散問題を有限要素法により解くことを目的として、fortran 言語を用いた解析プログラムを作成してきた。そのために、まず二次元定常移流拡散方程式を導出した。そして、有限要素法を適用するために二次元定常移流拡散方程式の有限要素式を導出した。移流拡散問題は移流項の卓越により解の不安定性が懸念されたため、安定化として SUPG 法を適用し、さらに衝撃捕捉項を加えることで解の安定化を試みた。そして、安定化を施した有限要素式も導出した。有限要素法適用に当たっては四角形一次要素及び三角形一次要素による要素分割を想定した。

有限要素法による数値計算を行うプログラムとして fem2dh_steady_AD という名前のフォルダ内に解析プログラムのソースコードを作成した。

プログラムの実行結果について述べる。まず、拡散問題として解いた結果について述べる。移流拡散問題で流速を 0 とした問題である二次元拡散問題については理論解と同一な結果が得られた。一次元定常移流拡散問題として解いた結果について述べる。流速が逆方向に作用する結果が得られたものの四角形要素を使った場合には x, y 方向のそれぞれの一次元問題として解析を行う理論解と同一な結果が得られた。また、三角形要素を使う場合でも流速の逆転現象は起きたが、 x, y 方向の一次元問題としてそれぞれ解いた場合に十分な精度の結果が得られた。二次元定常移流拡散問題として解いた結果について述べる。まず、想定できる移流拡散現象として、流下方向に温度の分布が広がる様子が確認できるかどうか境界条件を設定して四角形要素で解析を行った。その結果、予想通り流下方向に分布が広がる結果が得られた。また、ここまでの解析では解の不安定性が問題とならなかったため安定化を施さずに計算を行っていた。しかし、この問題において流速を大きくすると領域内で値が振動する解の不安定性がみられた。これに対して、安定化を施すことで解の振動が収まった結果が得られた。

他の文献と比較した結果について述べる。文献 [4, p.53] に示される条件で解析を行い、この文献と同じ結果が得られるかどうか検証した。要素分割のみ四角形要素 400 個にしてこの文献の条件で解析を行うと、安定化を施さない場合においては、領域全体で数値振動が発生する点で共通の結果が得られた。しかし、振動の大きさが異なり、文献 [4] において境界条件値が流れによって領域全体に広がっている様子が見られるが、本研究で得られた結果にはそのような様子がみられなかった。また、安定化を施した場合については、振動が収まらずさらに解が大きく振動する結果が得られ、文献と異なる結果となった。文献と同じ三角形要素でも解析を行った。安定化を施さない場合では振動の形式が異なるものの四角形要素での結果と同様に、領域全体で振動が起きたがその振動の大きさや、境界条件が領域に広がる様子が異なっていた。安定化を施した場合、四角形要素での結果と異なり、安定化の効果はみられた。しかし、局所的に振動がみられ文献とは異なる結果となった。

この文献 [4, p.53] の結果が妥当であるかの確認として他の解析プログラム『FreeFEM++』により得られた結果を用いて、文献と本研究で得た結果との比較を行った。その結果 FreeFEM++ の結果文献とも本研究で得られたものとも異なる結果を示した。

これらの実行結果から、本研究で作成した解析プログラムは二次元定常拡散問題と一次元定常移流拡散問題においては妥当な結果が得られるが、二次元定常移流拡散問題については妥当な結果が得られないといえる。

7.2 今後の課題

この章では本研究では成しえなかったが、今後研究を行う上での課題や展望などについて項ごとに述べていく。

7.2.1 移流項の計算と非定常問題への拡大の必要性

二次元定常移流拡散問題を扱う上で移流項が入る場合において妥当な結果が得られなかった。そのため、今後の課題としては移流項の計算ができるようにプログラムを改良することが第一に挙げられる。一次元定常移流拡散問題までは解くことができたが、二次元定常移流拡散問題となると解が振動して妥当な結果が得られなかった。一次元問題についても流速が逆方向に作用する現象が起きていたため、この流速の逆転現象も解決することが望まれる。

これらの不具合については本研究を進める上で問題となり、ソースコードにおける誤りの探索を行った。この探索は 6.3.1 項で論じた 4 要素での移流拡散問題の解析を対象にして、要素マトリックスの計算を行う ek4nod や境界条件の組み込みを行う bound さらに連立一次方程式を解く bandsolve のルーチン内で誤った計算を行っていないか、Microsoft 社の表計算ソフトである Microsoft Excel を用いて検算を行ったものである。この検算の中では、計算誤差の許容範囲と呼べる値の違いはあったもののそれ以外の誤りは見つからなかった。そのため、もっと根本的なところで誤りが存在しているのではないかと推測される。

解の振動については、6.3.3 項で論じた文献 [4, p.53] との比較において、本研究で得られた図 6.16 及び図 6.17 の結果では自然境界条件を満たしていないような結果が得られた。そのため、境界条件の処理について問題があることが疑われる。

本研究では理論解の求まる結果として、一次元定常拡散問題及び一次元定常移流拡散問題との比較を取り扱った。ここで、定常移流拡散問題における拡散係数が $k = 0$ のときである定常移流問題は理論解が求まらないため本研究では比較として用いなかった。しかしながら、本研究で問題となっている移流項の計算がきちんとできているか確認するためにも移流方程式との結果の比較を行うべきである。非定常問題の移流方程式は差分法などによる数値解の導出が研究されている。これらの他の数値解析手法を用いた結果と比較を行うために、非定常移流拡散問題を有限要素法で解析できるように改良を行う必要がある。非定常問題の解析については野寄氏の研究 [12] や文献 [4] など非定常問題を解析する手法についての研究や書物を参考にされたい。

7.2.2 境界積分項の計算での問題点

境界積分項で自然境界条件が組み込まれる温度フラックスベクトル $\mathbf{F}$ の計算において、本研究では自然境界条件値 g_2 が一定である場合のみを対象にしていた。そのため、実際のプログラムでは境界積分の計算を行おうが行わなかりうが結果としては同じとなる。そのため、プログラムでは 3.2.6 項のように節点番号の対応する右辺ベクトルに自然境界条件値 g_2 をそのまま足しこむことで自然境界条件の組み込みを行った。しかし、自然境界条件値が一定でなくなんらかの関数である場合や、自然境界条件値 g_2 が一定であっても、例えば四角形領域において右端の辺と下辺とでことなる値を採用した場合の右下の節点付近で値が異なる場合の対応が問題となる。

また、本研究で開発した解析プログラムでは境界積分項での計算で材質ごとに異なる物性値である温度伝導率 k をかけずに計算を行っているという致命的なミスがある。したがって、温度伝導率 k が掛かっていないため、自然境界条件の考慮された問題の解析では妥当な結果が得られないことが想定される。この k が掛かっていない問題は、もともとの本研究で開発する際に基本とした解析プログラムが定常拡散問題を取り扱っていたことに起因する。定常拡散問題の場合は定式化で両辺を k で除すことができ、実際には k は考慮しなくても解析ができたためである。そのため、この問題に気づくのが遅れ本研究で対処できなかった。

これらの境界積分項の問題を解決するために、きちんと k をかけて自然境界条件値 g_2 も境界上で内挿近似すべきである。このことについては、文献 [4, pp.33-34] や文献 [8, pp.256-266] 及び文献 [3, pp.66-67] に説明がある。これらを参考にプログラムを改善することが望まれる。

7.2.3 Upwind Galerkin 法での安定化の実装

本研究では、安定化の手法として SUPG 法と衝撃捕捉項の付加を取り入れた。しかし、Upwind Galerkin 法による安定化は解の精度という点からも採用せず、プログラムにも実装しなかった。そのため、6.3.3 項における文献 [4] に示されている結果との比較において、安定化を施した場合と安定化に SUPG 法を施した場合のそれぞれ図 6.13 の (b)Galerkin 法と (d)SUPG 法との比較で終わった。つまり、(c)Upwind Galerkin 法との比較については行うことができなかった。安定化を行うという点から考えれば、精度の高い SUPG 法を適用すれば十分だと考えられる。しかし、安定化の影響がどのようにでているか確認し、安定化の妥当性の検証を行うためにも、安定化を行うという点からは不必要だと考えられる Upwind Galerkin 法による安定化を適用できるようにプログラムを準備すべきである。Upwind Galerkin 法での安定化を実装し、文献 [4] との比較でやり残した図 6.13 の (c)Upwind Galerkin 法との比較を行うことが望まれる。

7.2.4 三角形要素と四角形要素の計算について

本研究では最終的に四角形要素と三角形要素での解析を可能にした。しかし、三角形要素については 5.2 節で示したように移流項の入っている行列の計算が煩雑となっており、もしかしたら途中で計算が間違っていたり、プログラムで書き間違えた可能性もある。また、一般的に要素分割を行う際には四角形と三角形要素を比べると、近似に使える節点の情報数や領域内での要素の対称性の点からも四角形要素の方が解析精度が高いといわれている。[6, p.195,p249][5]

これらのことから、6.3.3 項で示したように三角形要素と四角形要素の解析結果の違いは説明できると考える。しかし、単なる式展開上での計算ミスなのか、解析の精度の問題なのかの違いは特に前者の場合は問題となる。したがって、四角形要素と三角形要素とで解析を行う場合に、どれほどの精度の差が生じるのか知る必要がある。その情報をもとにして、計算ミスなのかどうか判断することができる。

7.2.5 熱伝導率の非等方性の取り扱い

本研究では熱伝導率 k が等方性であることを仮定した。しかしながら、異なる板材を貼り合わせて作られたような材質については熱伝導率に等方性を仮定できない場合が考えられる。そのため、熱伝導率が非等方性の場合も扱えるようにすべきである。

非等方性の熱伝導係数 k を扱う場合は文献 [6, pp.191-193] に示されるような定式化上での処理を行うことで扱うことができる。これを参考に非等方性の熱伝導率をもつ材質についても計算を行えるように改善することが望まれる。

7.2.6 その他—三次元問題への拡大、発熱率や流速及び物性値の取り扱い—

本研究では移流拡散問題解析の基礎として二次元問題を扱った。現実問題に適応するために、三次元問題へ拡張することが望まれる。また、単純化のため定式化の段階で発熱率 f は 0 として、流速は既知、物性値である κ, ρ, c_p は定数と置いた。

発熱率についても、輻射などの影響を考慮するために式に残したまま有限要素式を導出すべきであった。流速については、本来なら場所によって異なる値を示すはずで、領域全体の流速が分かっていることは考えにくい。したがって、将来的には流速も流体の運動方程式を解くことで式を立てて、別の解析プログラムで流速の値を求めるような解析を行い、それで得られた値を本研究の流速として扱えるようにしたい。

物性値 κ, ρ, c_p についても、実際には温度や圧力などの関数として表されるはずであるが、本研究では問題の単純化のため定数とした。可能であるならば、これらのパラメータも関数として扱うことが望ましい。

また、本研究では 6.1.2 項で解析の対象として鉄を想定して物性値のデータを用意した。しかしながら、解析のプログラムが未完成であったためこれらの物性値は使わなかった。解析プログラムが完成したら、材質を替えることでどのように温度分布が変化するか検討したい。

参考文献

- [1] 倉前宏行, 松本政秀『数値解析・演習』(大阪工業大学 工学部 環境工学科, 2011)
- [2] 矢野雄幸, 佐藤弘三『拡散方程式入門』(公害研究対策センター, 1978)
- [3] 菊池文雄『FEM+BEM=3 有限要素法概説 [新改訂版] -理工学における基礎と応用-』(サイエンス社, 2009)
- [4] 日本計算工学会流れの有限要素法研究委員会『続・有限要素法による流れのシミュレーション』(シュプリンガー・ジャパン, 2008)
- [5] 矢川元基, 吉村忍『有限要素法』計算力学と CAE シリーズ 1 (培風館, 1991)
- [6] 福森栄次『よくわかる有限要素法』(株式会社オーム社, 2005)
- [7] 自然科学研究機構 国立天文台『理科年表 平成 22 (机上版)』(丸善株式会社, 2009)
- [8] 森正武『FORTRAN 77 数値計算プログラミング (増補版)』(岩波コンピュータサイエンス) (岩波書店, 1996)
- [9] 戸川隼人『NS ライブラリ=1 ザ・FORTRAN77』(サイエンス社, 1997)
- [10] 矢吹道郎, 大竹敢『つかいこなす GNUPLOT』(株式会社テクノプレス, 1997)
- [11] 福井悠太『有限要素法による二次元定常ストークス流れの解析』(大阪工業大学 工学部 環境工学科, 2011)
- [12] 野寄哲也『有限要素法による軸対象熱伝導問題の解析』(大阪工業大学 工学部 環境工学科, 2011)
- [13] 赤司宏彰『有限要素法による二次元定常熱伝導問題の解析』(大阪工業大学 工学部 環境工学科, 2012)
- [14] 「freeFEM.org」〈<http://www.freefem.org/>〉(2012/3/21 アクセス)
- [15] 「FreeFem++ Japan」〈<https://sites.google.com/a/comfos.org/comfos/ffempp/jp>〉(2012/3/21 アクセス)
- [16] 齊藤宣一「有限要素法と非線形楕円型方程式の解の可視化」〈<http://www.infsup.jp/saito/ln/efem08h.pdf>〉(2012/3/21 アクセス)

謝辞

本論文は大阪工業大学工学部環境工学科において学部4回生における研究成果をまとめたものである。

研究を行うにあたり、同専攻 知能設計工学研究室 教授 松本 政秀先生には指導教官として本研究実施のテーマ及び機会を与えてくださり、日々のご指導・ご相談及びに要素形成プログラムの melement3, melement4 の作成にあたり多大なご支援を賜りました。ここに深く感謝申し上げます。

有限要素法の理論についてご教授くださり、また研究を進める中で、プログラムの開発で行き詰ったときにご助言をくださいました大阪工業大学 工学部 技術マネジメント学科 計算工学研究室 准教授 倉前 宏行先生に深く感謝申し上げます。

知能設計工学研究室 修士課程2回生 平田 慎太郎氏には、研究をする上でわからないところをその都度ご教授くださったり、ゼミで有限要素法の理論についてご教授くださいました。特に、私の研究で理解が不十分だった重み関数 w が境界上で0となる理由についてご教授くださいました。ここに感謝の意を表します。

同研究室 修士課程1回生 野寄 哲也氏にはゼミで有限要素法の理論についてご教授くださり、また tex や gnuplot の使い方などご説明くださり、卒業研究をする上でたいへんお世話になりました。ここに感謝申し上げます。

赤司 宏彰氏と新城 武氏はプレ配属の段階から同じ研究室でした。同じ研究室で有限要素法の勉強を行い、分からないところを教えてもらったりして苦難を乗り越えることができました。赤司氏の研究は本研究における三角形要素での解析において、大きな参考にさせていただきました。新城氏の地道にプログラムの間違いを探す姿勢は、本研究の後半に生じた文献との比較で生じたプログラムの問題点を探すうえで参考になりました。お二人に感謝いたします。

大阪工業大学 工学部 一般教育科 物理研究室 教授 林 正人先生には、本研究で焦点となった移流拡散問題の文献の結果である図 6.13 の理論解の導出で私の計算があっているか確認してくださいました。結果としては境界条件が不連続であるため理論解は存在しないという結果となり、本論文にはその導出過程の掲載に至りませんでした。微分方程式の考え方などたいへん勉強になりました。深く感謝申し上げます。

同研究室 准教授 中野 正浩先生には、定式化のところで移流方程式の導出が困難であったため、ご相談させていただきました。温度 T はある基準点 T_0 とそこからの差 dT の和という形で表されるという助言をくださり、無事に導出することができました。深く感謝申し上げます。

この他にも、研究を行う上でお世話になった環境工学科事務室の方々や、大学で授業をなさってくださいました先生方、研究発表を見に来て下さった方々へ感謝申し上げます。

付 録 A 付 録

付録として、本研究で作成したプログラムのツリー図やプログラムのソースコードを掲載し、ソースコードで使われている主な変数の説明をまとめた。さらに、本論文において式に登場する記号や用語について表にまとめて掲載した。

A.1 プログラム fem2dh_steady_AD の構造

本研究で開発した二次元定常移流拡散問題を有限要素法で解析するプログラムの fem2dh_steady_AD のツリー図を図 A.1 に示した。なお、fem2dh_steady_AD は fortran 言語で記述されている。

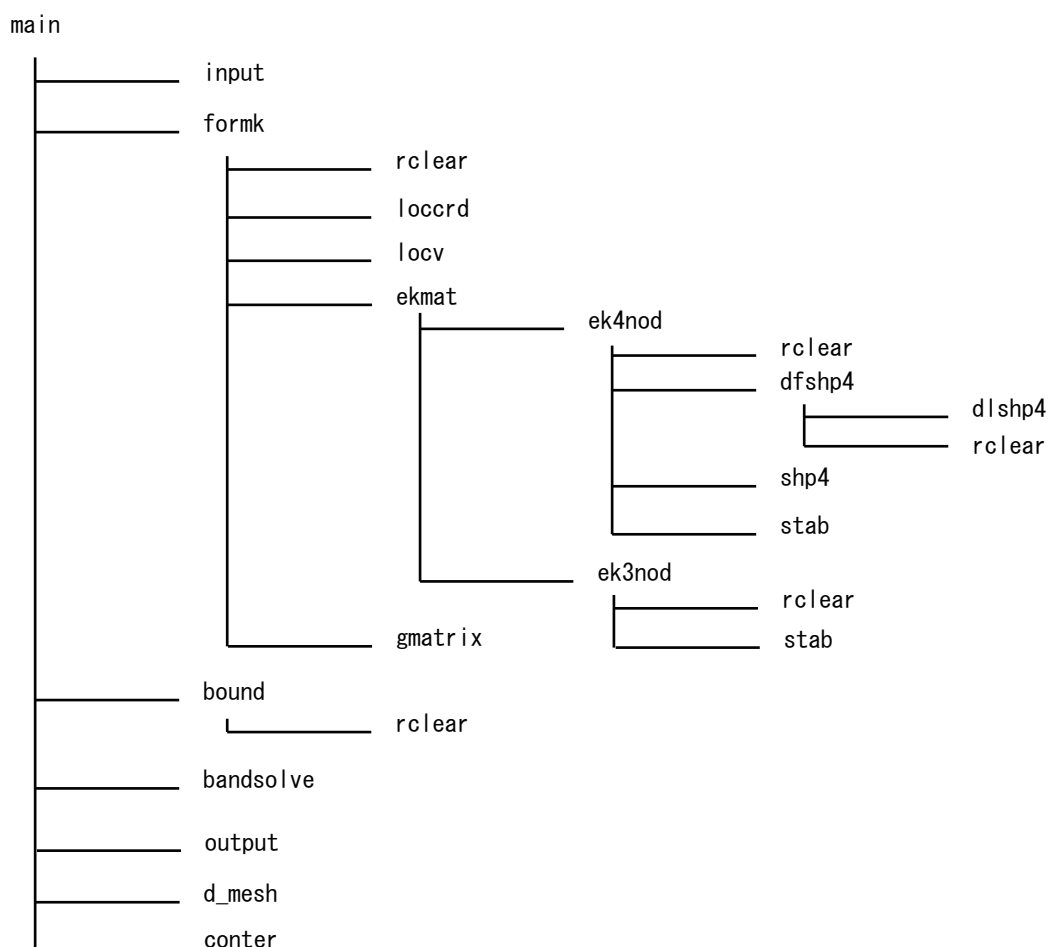


図 A.1: fem2dh_steady_AD のツリー図

A.2 プログラムのルーチンの説明

ルーチンとはあるひとまとまりの手続きをまとめたものである。本研究で開発したプログラムはメインルーチンからサブルーチンを呼び出すことで繰り返し登場する処理や作業単位を分割できるようにした。図 A.1 に登場するルーチンと図 A.1 に登場していないが用いたルーチンを表 A.1 に示し、表内でそれぞれのルーチンの役割を説明した。

表 A.1: ルーチンの意味

ルーチン名	意味
main	メインルーチン
input	入力データの設定
formk	係数行列を形成
rclear	配列のゼロクリア
loccrd	ある要素の節点座標を用意
locv	ある要素の節点流速を用意
ekmat	要素の選択
ek4nod	四角形要素での要素マトリックスを形成する
blkdta000	要素の節点数と数値積分での座標値及び重みの値を用意
dfshp4	四角形要素の全体座標系での内挿関数の偏微分値を計算
dlshp4	四角形要素の局所座標系での内挿関数の偏微分値を計算
shp4	四角形要素での内挿関数と内挿近似した流速を計算
stab	安定化パラメータを計算
ek3nod	三角形要素での要素マトリックスの形成
gmatrix	全体マトリックスに要素マトリックスを足しこむ
bound	自然境界条件と基本境界条件の組み込み
bandsolve	ガウスの消去法により連立一次方程式を計算
output	結果の出力
d_mesh	要素分割図の描画
conter	温度分布図の描画

A.3 解析プログラム fem2dh_steady_AD のソースコードの掲載

表 A.1 で新たに自作したルーチンは、locv,ek3nod,gmatrix,stab である。また、本研究で開発したプログラムのベースである福井氏 [11] が作成した二次元定常拡散問題を解析するプログラム fem2dh_steady_test を改変して利用したものは main,input,formk,ek4nod,shp4,bound,bandsolve,output である。これらの内、プログラム内で新しい処理が行われた主要なものである input,locv,ek4nod,shp4,ek3nod,stab,gmatrix,bound,bandsolve のソースコードを掲載し、新しく改変した部分などを説明していく。

input のソースコードを以下に示した。

リスト A.1: input のソースコード

```

1      SUBROUTINE INPUT(CRD,NOP,KMATR,KELMN,IA,VN,BV,BVN,NOPBV,NOPBVN)
2      C
3      C reading input data from the device unit 5(keyboard in default)
4      C
5      COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD, NT
6      COMMON /SIZE/ NP, NE, NHBW, ND, NN
7      COMMON /MATR/ RKAPPA(5), HEATCAP(5), DENSITY(5), RK(5)
8      COMMON /ELMN/ LELM(6)
9      DIMENSION VN(NFD,NPD), NOPBV(NPD), BV(NPD), NOPBVN(NPD), BVN(NPD)
10     C
11     DIMENSION CRD(NFD,NPD), NOP(NPED,NED), KMATR(NED), KELMN(NED)
12     DIMENSION IA(*)
13
14     C 材質の種類数を読み込む
15     READ(10,500) NS
16     500 FORMAT(40I5)
17     IF(NS.GT.5) THEN
18         WRITE(6,*) ' TOO MANY MATERIALS. MUST BE LESS THAN 6 '
19         STOP
20     ENDIF
21
22     C 材質の物性値を読み込む
23     READ(10, "(1P3E10.0)") (RKAPPA(I), HEATCAP(I), DENSITY(I), I=1, NS)
24     C 温度伝導率の計算
25     DO I=1, NS
26         RK(I)=RKAPPA(I)/HEATCAP(I)/DENSITY(I)
27     END DO
28
29     C 要素数の読み込み
30     READ(10,500) NE
31     C 要素の情報 (要素の材質、形状、節点番号) を読み込む
32     DO I=1, NE
33         READ(10,500) N, KMATR(I), KELMN(I), (NOP(J,I), J=1, LELM(KELMN(I)))
34     END DO
35
36     C 節点数を読み込む
37     READ(10,500) NP
38     C 節点の情報 (節点の座標、自由度番号 (節点にある未知数の番号) ) を読み込む
39     DO I=1, NP
40         READ(10,530) N, (CRD(J,N), J=1, NFD), (IA((I-1)*NT+J), J=1, NT)

```

```

41      END DO
42      530 FORMAT(I5,2E10.0,I8)
43
44      C 速度の読み込み
45      C 速度仮入力
46      DO I=1,NP
47      c      VN(1,I)=-0.5e-0
48      c      VN(2,I)=VN(1,I)*SQRT(3.0)
49      VN(1,I)=-1.0
50      VN(2,I)=0.0
51      END DO
52
53      C READ OF DIRICHLET CONDITIION
54      READ(10,"(I5)") ND
55      DO I=1,ND
56      READ(10,"(I5,E10.0)") NOPBV(I),BV(I)
57      END DO
58
59      C READ OF NEUMANN CONDITION
60      READ(10,"(I5)") NN
61      DO I=1,NN
62      READ(10,"(I5,E10.0)") NOPBVN(I),BVN(I)
63      END DO
64
65
66      CC 入力データの書き出し
67      C 毎回入力データを出力することは不要なので、ここで入力データを出力するかしないか指定
68      C INP=1 とすると入力データを出力する
69      INP=0
70      IF(INP.EQ.1)THEN
71
72      C 材質の種類数を出力
73      WRITE(*,"('Number of materials is ',I5)") NS!材質の種類数
74      C 材質の物性値を出力
75      WRITE(*,100) (I,RKAPPA(I),HEATCAP(I),DENSITY(I),RK(I),I=1,NS)
76      100 FORMAT(' No. RKAPPA [W/m/K] HEATCAP[J/kg/K] DENSITY [kg/m^3]
77      & k[m^2/s] ',/(1H,I5,1P4E15.5))
78
79      C 要素数を出力
80      WRITE(*,"(' Number of elements is ',I5)") NE
81      C 要素の情報(要素の材質、形状、節点)を読み込む
82      WRITE(*,*)' No. (Mat:Elm) Points'
83      DO I=1,NE
84      WRITE(6,110) I,KMATR(I),KELMN(I),(NOP(J,I),J=1,LELM(KELMN(I)))
85      END DO
86      110 FORMAT(1H,I5,' (',I1,':',I1,') ',40I5)
87
88      C 節点数を出力
89      WRITE(*,"(' Number of points is ',I5)") NP
90      C 節点の情報(節点の座標、自由度番号(節点にある未知数の番号))を出力
91      WRITE(*,*)' No. Coordinates'
92      DO I=1,NP
93      WRITE(*,120) I,(CRD(J,I),J=1,NFD),(IA((I-1)*NT+J),J=1,NT)
94      END DO
95      120 FORMAT(1H,I5,1P2E15.5,I8)
96
97      C 節点での流速を出力
98      WRITE(*,*)'Node No. Vx Vy '
99      DO I=1,NP
100     WRITE(*,"(1H,I5,1P2E15.5)") I,(VN(J,I),J=1,2)
101     END DO
102
103      C 基本境界条件数を出力
104      WRITE(*,"(' Number of essential boundary condition is ',I5)") ND
105      IF(ND.GT.0) WRITE(*,*)' No. Value'
106      C 基本境界条件が与えられている節点番号とその値を出力
107      DO I=1,ND
108      WRITE(*,"(1H,I5,1PE15.5)") NOPBV(I),BV(I)
109      END DO
110
111      C 自然境界条件数を出力
112      WRITE(*,"(' Number of narural boundary conditions is ',I5)") NN
113      IF(NN.GT.0) WRITE(*,"(' No. Value')")
114      C 自然境界条件が与えられている節点番号とその値を出力
115      DO I=1,NN
116      WRITE(*,"1H,I5,1PE15.5)") NOPBVN(I),BVN(I)
117      END DO
118
119      END IF
120
121      RETURN
122      END

```

input では各種の入力データをファイルから読み込んで設定する。福井氏 [11] の作成したプログラムでは材質の物性値として熱伝導率を表す RKAPPA のみ読み込んでいた。移流拡散方程式において温度伝導率が登場するため、22 行目で熱伝導率 RKAPPA の他に定圧比熱容量と密度であるそれぞれ HEATCAP,DENSITY を読み込むようにした。温度伝導率は RK として 26 行目で計算した。また、福井氏が作ったプログラムでは境界条件は別のサブルーチンである load,bound で読み込むようにしていた。しかし、理解のしやすさから入力を一か所に統一しておきたいと考え、本研究では input の 53 行目から 63 行目で境界条件値を読み込むようにした。

流速の入力についても本研究で新しく実装した。ただし、流速の入力についてはファイルから読み込むのではな

く、45 行目から 51 行目で領域の全節点に一樣な値を直接設定している。
locv のソースコードを以下に示した。

リスト A.2: locv のソースコード

```

1 C copying coordinates of the element from V to ZV
2 C using the data of NOP
3 C KEL is a kind of element (1~5)
4 C
5 SUBROUTINE LOCv(KEL,VN,NOP,ZV)
6 COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD
7 COMMON /ELMN/ LELM(6)
8 C
9 DIMENSION VN(NFD,NPD), NOP(NPED), ZV(NFD,NPED)
10 C
11 DO I=1,LELM(KEL)
12 DO J=1,NFD
13 ZV(J,I) = VN(J,NOP(I))
14 C WRITE(*,*)VN(J,NOP(I))
15 END DO
16 END DO
17
18 C WRITE(*,*) 'write by locv'
19 C DO 10 J=1,2
20 C DO 10 I=1,4
21 C WRITE(6,*)ZV(J,I)
22 C 10 CONTINUE
23 RETURN
24 END

```

locv は本研究で新しく作成したサブルーチンである。locv では locrd と同様に、ある要素を構成する節点での流速を取り出して ZV にコピーしている。こうすることで、shp4 や ek3nod で流速を内挿近似する際にデータを取り扱いやすくなる。

以下に ek4nod のソースコードを示した。

リスト A.3: ek4nod のソースコード

```

1 SUBROUTINE EK4NOD(KEL,KMAT,CRD,EK,EA,EA_de,EK_de,VN,Pe,HEN)
2 C ガウスの数値積分を行い要素マトリックスを作る
3 C making an element stiffness matrix of 4-node element
4 C
5 COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD, NT
6 COMMON /MATR/ RKAPPA(5),HEATCAP(5),DENSITY(5),RK(5)
7 COMMON /GAUS/ AI(5,5),WI(5,5) !数値積分での座標 AI と重み WI
8 COMMON /SIZE/ NP, NE, NHBW,ND,NN
9 C
10 DIMENSION CRD(2,4), DNDX(2,4) !DNDX 全体座標系の微分
11 DIMENSION EK(NT*NPED,NT*NPED), EA(NT*NPED, NT*NPED)
12 DIMENSION EA_de(NT*NPED, NT*NPED), EK_de(NT*NPED, NT*NPED)
13 DIMENSION WK(4,4),WA(4,4), WA_de(4,4),WK_de(4,4)
14 DIMENSION RN(4)
15 DIMENSION VN(2,4),V(2)
16 DIMENSION HEN(4)
17 C
18 DATA IPT /2/ !積分点の数
19
20 CALL RCLEAR(EK,NT*NPED*NT*NPED)
21 CALL RCLEAR(EA,NPED*NPED)
22 CALL RCLEAR(EA_de,NT*NPED*NT*NPED)
23 CALL RCLEAR(EK_de,NT*NPED*NT*NPED)
24
25 C WRITE(*,*)'VN write by ek4nod 1'
26 C WRITE(*,*)((VN(J,M),M=1,4),J=1,2)
27
28 DO 100 IY = 1,IPT
29 DO 100 IX = 1,IPT
30 C
31 CALL DFSHP4(AI(IX,IPT),AI(IY,IPT),CRD,DNDX,DETJ) !形状関数の定義
32 C
33 C 局所座標系の D_e の中身 (偏微分のかけ算、の足し算) の計算
34 C WK=(dN_i dN_j)/(d xi d xi)+(dN_i dN_j)/(d eta d eta)
35 DO J = 1,4
36 DO I = 1,4
37 WK(I,J) = 0.0
38 DO K = 1,2
39 WK(I,J) = WK(I,J) + DNDX(K,I)*DNDX(K,J)
40 C WRITE(6,*) DNDX(K,I),DNDX(K,J)
41 END DO
42 END DO
43 END DO
44 550 FORMAT(1P4E13.4)
45 C WRITE(6,*) 'Write WK by ek4nod'
46 C WRITE(6,550) (WK(1,J),WK(2,J),WK(3,J),WK(4,J),J=1,4)
47
48 DO 20 J = 1,4
49 DO 20 I = 1,4
50 cc EK(I,J)=0.0
51 C 局所座標系から変換して全体座標での D_e をガウスの数値積分を使って計算
52 C EK=k Σ Σ WI_x W_y {(dN_i dN_j)/(dx dx)+(dN_i dN_j)/(dy dy)}dxdy
53 C =k Σ Σ WI_x W_y {(dN_i dN_j)/(d xi d xi)+(dN_i dN_j)/(d eta d eta)} |J| d xi d eta
54

```

```

55      EK(I,J)=EK(I,J) + DETJ*WI(IX,IPT)*WI(IY,IPT)
56      & *WK(I,J)*RK(KMAT)
57 C      WRITE(*,*)'VN write by ek4nod 2'
58 C      WRITE(*,*)((V(J,M),M=1,4),J=1,2)
59
60      20  CONTINUE
61 C      WRITE(6,*)'EK write by ek4nod'
62 C      WRITE(6,550) ((EK(I,J),J=1,4),I=1,4)
63
64      CALL SHP4(AI(IX,IPT),AI(IY,IPT),RN,CRD,VN,V)
65      WRITE(6,*)'V write by ek4nod'
66 C      WRITE(6,550) ((V(I)),I=1,2)
67 CC     WRITE(*,*)'J,I,K,RN(I),V(K),DNDX(L,I)'
68
69 C  局所座標系の A_e の中身 ( ) の計算
70 C  WA=v_x N_j (dN_i)/(d xi) + v_y N_j (dN_i)/(d eta)
71 C      WRITE(*,*)"RN(I) BY EK4NOD"
72 C      WRITE(*, "(4E13.4)") (RN(I),I=1,4)
73 c      WRITE(*,*)"V(I) BY EK4NOD",V(1),V(2)
74 c      WRITE(*,*)"RN(I),DNDX(1,I),DNDX(2,I) BY EK4NOD"
75 c      WRITE(*, "(3E15.5)") (RN(I),(DNDX(J,I),J=1,2),I=1,4)
76
77      DO J = 1,4
78      DO I = 1,4
79      WA(I,J) = 0.0
80 150  FORMAT(I2,I2,I2,I2,1H ,1PE13.4,1PE13.4,1PE13.4)
81      DO K = 1,2
82 C      WRITE(*,150)J,I,L,RN(I),V(K),DNDX(L,I)
83
84      WA(I,J) = WA(I,J) + RN(J)*V(K)*DNDX(K,I)
85 c      WA(I,J) = WA(I,J) + RN(J)*V(1)*DNDX(1,I)+RN(J)*V(2)*DNDX(2,I)
86      END DO
87      END DO
88      END DO
89 c      WRITE(6,*)'WA IXIY write by ek4nod',IX,IY
90 c      WRITE(6,550) ((WA(I,J),J=1,4),I=1,4)
91
92 c      WRITE(*,*)"DETJ,WI(IX,IPT),WI(IY,IPT)"
93 c      WRITE(*,*)DETJ,WI(IX,IPT),WI(IY,IPT)
94
95 C  局所座標系から変換して全体座標での A_e をガウスの数値積分を使って計算
96      DO J = 1,4
97      DO I = 1,4
98      EA(I,J)=EA(I,J) + DETJ*WI(IX,IPT)*WI(IY,IPT)*WA(I,J)
99      END DO
100     END DO
101 c      WRITE(6,*)'EA write by ek4nod'
102 c      WRITE(6,550) ((EA(I,J),J=1,4),I=1,4)
103
104 c  安定化項のパラメータ tau_e, tau_L を計算
105      CALL STAB(KEL,KMAT,tau_e,tau_L,V,DNDX,Pe,CRD,HEN,HE,VE)
106 C      WRITE(*,*)'tau_e,L write by ek4nod'
107 C      WRITE(6,550) tau_e,tau_L
108
109 C  局所座標系の A_de の中身 ( ) の計算
110
111      DO 50 J = 1,4
112      DO 50 I = 1,4
113      WA_de(I,J) = 0.0
114      WA_de(I,J) = ( V(1)*DNDX(1,J)+ V(2)*DNDX(2,J))*
115      & ( V(1)*DNDX(1,I)+ V(2)*DNDX(2,I))
116
117 C      WRITE(6,*) DNDX(A,I),DNDX(A,J)
118 C      WRITE(6,*) I,J,WA_de(I,J)
119 50  CONTINUE
120
121 C
122      DO 60 J = 1,4
123      DO 60 I = 1,4
124 C  局所座標系から変換して全体座標での A_de をガウスの数値積分を使って計算
125
126      EA_de(I,J)= tau_e*DETJ*WI(IX,IPT)*WI(IY,IPT)*WA_de(I,J)
127
128 60  CONTINUE
129
130 C  局所座標系の K_de の中身 ( 偏微分 の かけ算、 の 足し算 ) の計算
131      DO 70 J = 1,4
132      DO 70 I = 1,4
133      WK_de(I,J) = 0.0
134      DO 72 K_de = 1,2
135      WK_de(I,J) = WK_de(I,J) + DNDX(1,I)*DNDX(1,J)
136      WK_de(I,J) = WK_de(I,J) + DNDX(2,I)*DNDX(2,J)
137 C      WRITE(6,*) DNDX(K,I),DNDX(K,J)
138 C 72  CONTINUE
139 C      WRITE(6,*) I,J,WK(I,J)
140 70  CONTINUE
141 C
142      DO 80 J = 1,4
143      DO 80 I = 1,4
144 C  局所座標系から変換して全体座標での K_de をガウスの数値積分を使って計算
145      EK_de(I,J)=EK_de(I,J)+tau_L*DETJ*WI(IX,IPT)*WI(IY,IPT)*WK_de(I,J)
146
147 80  CONTINUE
148
149 100 CONTINUE
150 c      WRITE(*,*)'HE,VE write by EK4NOD'
151 c      WRITE(*,*) HE,VE

```



```

152 c      WRITE(*,*)'V write by EK4NOD'
153 c      WRITE(*,*) (V(I),I=1,2)
154
155 c      WRITE(6,*)'RN write by ek4nod'
156 c      WRITE(6,550) (RN(I),I=1,4)
157
158 c      WRITE(6,*) 'WK  Write by ek4nod'
159 c      WRITE(6,550) ((WK(I,J),J=1,4),I=1,4)
160 c      WRITE(6,*)'EK  write by ek4nod'
161 c      WRITE(6,550) ((EK(I,J),J=1,4),I=1,4)
162
163 c      WRITE(6,*)'WA write by ek4nod END'
164 c      WRITE(6,550) ((WA(I,J),J=1,4),I=1,4)
165
166 c      WRITE(*,*)'tau_e,L write by ek4nod'
167 c      WRITE(6,550) tau_e,tau_L
168
169      RETURN
170      END

```

ek4nod では四角形要素での拡散項、移流項、安定化項、衝撃捕捉項の各成分の計算を行っている。当初は拡散の要素マトリックスである EK のみが計算されていた。移流拡散方程式を解けるようにするため、64 行目から 147 行目に移流項 EA と安定化項 EA_DE 及び衝撃捕捉項 EK_DE を計算するように追加した。

以下に shp4 のソースコードを示した。

リスト A.4: shp4 のソースコード

```

1      SUBROUTINE SHP4(XI,ETA,RN,CRD,VN,V)
2      DIMENSION RN(4),CRD(2,4),VN(2,4),V(2)
3
4      C      WRITE(*,*)"XI.ETA BY SHP4 "
5      C      WRITE(*,*) XI,ETA
6
7      C      内挿関数の計算
8      RN(1) = 0.25 * (1.0-XI) * (1.0-ETA)
9      RN(2) = 0.25 * (1.0+XI) * (1.0-ETA)
10     RN(3) = 0.25 * (1.0+XI) * (1.0+ETA)
11     RN(4) = 0.25 * (1.0-XI) * (1.0+ETA)
12
13     C      速度 V の内挿近似
14     V(1) = 0.0
15     V(2) = 0.0
16     C      WRITE(6,*)'VN write by shp4'
17     C      WRITE(6,550) ((VN(J,I),J=1,2),I=1,4)
18     550 FORMAT(4E15.5)
19     DO 20 I=1,4
20     V(1)=V(1)+RN(I)*VN(1,I)
21     V(2)=V(2)+RN(I)*VN(2,I)
22     20 CONTINUE
23
24     RETURN
25     END

```

shp4 は野寄氏 [12] が開発したプログラムを参考に本研究で新しく追加したサブルーチンである。ここでは、四角形要素の局所座標系での内挿関数である RN を計算し、さらに流速 V を内挿近似して計算している。

ek3nod のソースコードを以下に示した。

リスト A.5: ek3nod のソースコード

```

1      SUBROUTINE EK3NOD(KEL,KMAT,CRD,EK,EA,EA_de,EK_de,VN,Pe,HEN)
2      C making an element stiffness matrix of 3-node element
3      C
4      COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD, NT
5      COMMON /MATR/ RKAPPA(5),HEATCAP(5),DENSITY(5),RK(5)
6      COMMON /SIZE/ NP, NE, NHBW,ND,NN
7
8      C
9      DIMENSION DNDX(2,3)
10     DIMENSION CRD(2,3)
11     DIMENSION EK(NT*NPED,NT*NPED), EA(NT*NPED,NT*NPED)
12     DIMENSION EA_de(NT*NPED,NT*NPED), EK_de(NT*NPED,NT*NPED)
13     DIMENSION VN(2,3),V(2)
14     DIMENSION HEN(3)
15
16     C
17     CALL RCLEAR(EK,NT*NPED*NT*NPED)
18     CALL RCLEAR(EA,NPED*NPED)
19     CALL RCLEAR(EA_de,NT*NPED*NT*NPED)
20     CALL RCLEAR(EK_de,NT*NPED*NT*NPED)
21
22     C      WRITE(*,*)"CRD(1,J),CRD(2,J), BY EK3NOD"
23     C      WRITE(*,"(2F10.5)") (CRD(1,J),CRD(2,J),J=1,3)
24
25     c      要素の面積
26     S=(CRD(1,1)*(CRD(2,2)-CRD(2,3))+CRD(1,2)*(CRD(2,3)-CRD(2,1))
27     & +CRD(1,3)*(CRD(2,1)-CRD(2,2)))/2.0
28     D=2.0*S
29
30     c      局所座標系の偏微分値
31     DNDX(1,1)=(CRD(2,2)-CRD(2,3))/D
32     DNDX(1,2)=(CRD(2,3)-CRD(2,1))/D

```

```

32      DNDX(1,3)=(CRD(2,1)-CRD(2,2))/D
33
34      DNDX(2,1)=(CRD(1,3)-CRD(1,2))/D
35      DNDX(2,2)=(CRD(1,1)-CRD(1,3))/D
36      DNDX(2,3)=(CRD(1,2)-CRD(1,1))/D
37
38  C 要素内の流速の成分の計算
39      V(1)=S/3*(VN(1,1)+VN(1,2)+VN(1,3))
40      V(2)=S/3*(VN(2,1)+VN(2,2)+VN(2,3))
41
42  C      WRITE(*,*) "S,DNDX BY EK3NOD"
43  C      WRITE(*,*) S
44  C      WRITE(*,*) "DNDX(1,J),DNDX(2,J)"
45  C      WRITE(*,"(2F10.5)") (DNDX(1,J),DNDX(2,J),J=1,3)
46
47
48  C 拡散項の計算
49      DO J = 1,3
50          DO I = 1,3
51              EK(I,J)= S*RK(KMAT)*(DNDX(1,I)*DNDX(1,J)+DNDX(2,I)*DNDX(2,J))
52          END DO
53      END DO
54
55  C 移流項の計算
56      DO J = 1,3
57          DO I = 1,3
58              EA(I,J)=EA(I,J) + S/12*(VN(1,J)*DNDX(1,I)+VN(2,J)*DNDX(2,I))
59          DO K = 1,3
60              EA(I,J)=EA(I,J) + S/12*(VN(1,K)*DNDX(1,I)+VN(2,K)*DNDX(2,I))
61          END DO
62      END DO
63  END DO
64
65  c IF文を使った場合の計算法
66      IF(J.EQ. K) THEN
67          c      EA(I,J)=EA(I,J) + (VN(1,K)*DNDX(1,I)+VN(2,K)*DNDX(2,I))*S/6
68          c      ELSE
69          c      EA(I,J)=EA(I,J) + (VN(1,K)*DNDX(1,I)+VN(2,K)*DNDX(2,I))*S/12
70          c      END IF
71
72      CALL STAB(KEL,KMAT,tau_e,tau_L,V,DNDX,Pe,CRD,HEN,HE,VE)
73
74  C 安定化項の計算
75      DO 50 J = 1,3
76      DO 50 I = 1,3
77      DO 50 K = 1,2
78      DO 50 L = 1,2
79      DO 50 M = 1,3
80          EA_de(I,J)= S/12*TAU_E*VN(K,M)*VN(L,M)* DNDX(K,J)*DNDX(L,I)
81      DO 50 N = 1,3
82          EA_de(I,J)= S/12*TAU_E*VN(K,M)*VN(L,N)* DNDX(K,J)*DNDX(L,I)
83      50 CONTINUE
84
85  cc IF文を使った場合の計算法
86  C      DO J = 1,3
87  C      DO I = 1,3
88  CC      DO K = 1,3
89  C      DO L = 1,2
90  C      DO M = 1,2
91  C      IF (K.EQ. 3) THEN
92  C      EA_de(I,J)=
93  C      & (S/6*VN(L,K)*VN(M,K)+S/12*(VN(L,K)*VN(M,1)+VN(L,1)*VN(M,K)))
94  C      & * DNDX(L,J)*DNDX(M,I)*TAU_E
95
96  C      & (S/6*ZV(1,K)*ZV(1,K)+S/12*ZV(1,K)*ZV(1,1))
97  C      & * DNDX(1,J)*DNDX(1,I)
98
99  C      & (S/6*ZV(1,K)*ZV(2,K)+S/12*ZV(1,K)*ZV(2,1)+ZV(1,1)*ZV(2,K))
100 C      & * (DNDX(1,J)*DNDX(2,I)+DNDX(2,J)*DNDX(1,I))
101
102 C      & (S/6*ZV(2,K)*ZV(2,K)+S/12*ZV(2,K)*ZV(2,1))
103 C      & * DNDX(2,J)*DNDX(2,I)
104
105 C      ELSE
106 C      EA_de(I,J)=
107 C      & (S/6*VN(L,K)*VN(M,K)+S/12*(VN(L,K)*VN(M,K+1)+VN(L,K+1)*VN(M,K)))
108 C      & * DNDX(L,J)*DNDX(M,I)*TAU_E
109
110 C      & (S/6*ZV(1,K)*ZV(1,K)+S/12*ZV(1,K)*ZV(1,K+1))
111 C      & * DNDX(1,J)*DNDX(1,I)
112
113 C      & (S/6*ZV(1,K)*ZV(2,K)+S/12*ZV(1,K)*ZV(2,K+1)+ZV(1,K+1)*ZV(2,K))
114 C      & * (DNDX(1,J)*DNDX(2,I)+DNDX(2,J)*DNDX(1,I))
115
116 C      & (S/6*ZV(2,K)*ZV(2,K)+S/12*ZV(2,K)*ZV(2,K+1))
117 C      & * DNDX(2,J)*DNDX(2,I)
118
119 C      END IF
120 C      END DO
121 C      END DO
122 C      END DO
123 C      END DO
124 C      END DO
125
126
127  C 衝撃捕捉項の計算
128      DO J = 1,3

```

```

129      DO I = 1,3
130      EK_de(I,J)=S*TAU_L*(DNDX(1,I)*DNDX(1,J)+DNDX(2,I)*DNDX(2,J))
131      END DO
132  END DO
133
134  RETURN
135  END

```

ek3nod は本研究で新しく追加したサブルーチンである。ek3nod では三角形要素での拡散項、移流項、安定化項、衝撃捕捉項の各成分の計算をしている。三角形要素を用いれば四角形要素と異なり、座標変換や数値積分を用いなくても面積座標の積分公式を用いることで積分の計算ができるため、dfshp4 dlshp4 shp4 に相当するサブルーチンが不要となっている。

stab のソースコードを以下に示した。

リスト A.6: stab のソースコード

```

1  SUBROUTINE STAB(KEL,KMAT,tau_e,tau_L,V,DNDX,Pe,CRD,HEN,HE,VE)
2
3  COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD, NT
4  COMMON /MATR/ RKAPPA(5), HEATCAP(5), DENSITY(5), RK(5)
5  COMMON /SIZE/ NP, NE, NHBW
6  COMMON /ELMN/ LELM(6)
7
8  DIMENSION V(2), DNDX(2,LELM(KEL)), CRD(2,LELM(KEL))
9  DIMENSION HEN(LELM(KEL))
10
11 C  WRITE(*,*) ((DNDX(I,J),J=1,4),I=1,2)
12
13  Ve =0.0
14  Pe =0.0
15  tau_e =0.0
16  tau_L =0.0
17  he=0.0
18  Ve=SQRT(V(1)**2+V(2)**2)
19
20 c 要素サイズの計算
21 IF(Ve .GT. 0.001) THEN
22   DO J=1,LELM(KEL)
23    he = he+ABS(V(1)/Ve*DNDX(1,J)+V(2)/Ve*DNDX(2,J))
24   END DO
25   he=2.0/he
26 ELSE
27 cc 要素の平均の一辺の長さを計算
28   DO I=1,LELM(KEL)-1
29    HEN(I)=SQRT((CRD(1,I)-CRD(1,I+1))**2+(CRD(2,I)-CRD(2,I+1))**2)
30   END DO
31   HEN(LELM(KEL))=SQRT((CRD(1,1)-CRD(1,LELM(KEL)))**2
32   & +(CRD(2,1)-CRD(2,LELM(KEL)))**2)
33   DO I=1,LELM(KEL)
34    HE=HE+HEN(I)
35   END DO
36   HE=HE/REAL(LELM(KEL))
37 ENDIF
38 c ペクレ数の計算
39 Pe = Ve * he / (2.0 *RK(KMAT))
40 C  WRITE(*,*)'HE,VE write by stab',HE,VE
41
42 c 安定化パラメータの計算
43 tau_e = he / (2.0 * Ve) * (1.0/tanh(Pe) - 1.0/Pe)
44
45 IF(Pe .GT. 3.0) THEN
46   tau_L = Ve * he / 2.0
47 ELSE
48   tau_L = Ve * he / 2.0 * Pe / 3.0
49 ENDIF
50 RETURN
51 END

```

stab は本研究で新しく作成したサブルーチンである。安定化の英語である stabilization から名付けた。stab は安定化パラメータである τ_e, τ_{LSIC} の計算を行うサブルーチンである。18 行目で要素内での流速の大きさを計算し VE に代入している。20 行目から 27 行目で要素サイズの計算を行っている。本研究では、流速の大きさ VE が 0.001 以下の場合は流速が十分に小さいと判断し、要素の平均の一片の長さを要素サイズとして設定している。この部分は式 (6.1.14) と対応している。26 行目でペクレ数 Pe を計算し、これらの要素サイズとペクレ数を用いて、安定化パラメータ τ_e, τ_{LSIC} を 42 行目から 49 行目で計算している。この部分は式 (6.1.12) 及び式 (6.1.13) と対応する。

gmatrix のソースコードを以下に示した。

リスト A.7: gmatrix のソースコード

```

1  SUBROUTINE GMATRIX(KEL,NOP,EK,EA,EA_de,EK_de,A,IA)
2  COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD, NT
3  COMMON /SIZE/ NP, NE, NHBW,ND,NN
4  COMMON /ELMN/ LELM(6)
5
6 C
7  DIMENSION NOP(NPED)
8  DIMENSION A(NP,NP), IA(NP)
9  DIMENSION EK(NT*NPED,NT*NPED),EA(NT*NPED,NT*NPED)
10 DIMENSION EA_de(NT*NPED,NT*NPED),EK_de(NT*NPED,NT*NPED)

```

```

10 C THE FOLLOWING FUNCTION GIVES THE LOCATION IN 1-D ARRAY FROM 2-D INDEX (I,J)
11
12
13 DO I=1, LELM(KEL)
14 DO J=1, LELM(KEL)
15 A(NOP(I),NOP(J))=A(NOP(I),NOP(J))
16 & +EK(I,J)
17 & +EA(I,J)
18 & +EA_de(I,J)
19 & +EK_de(I,J)
20 END DO
21 END DO
22
23 RETURN
24 END

```

gmatrix は本研究で新しく作成したサブルーチンである。gmatrix では全体マトリックスに要素マトリックスの足しこみを行っている。global matrix の名前から gmatrix というルーチン名を名付けた。福井氏 [11] が作成した二次元定常拡散問題の解析プログラムでは係数マトリックスの対称性を利用して、係数マトリックスの半分の値だけをメモリに保持することで、メモリの節約を行って全体マトリックスを作成する lc2glb というサブルーチンが使われていた。しかし、本研究が取り扱う二次元定常移流拡散問題では移流項が入ることで係数マトリックスが非対称になるため、福井氏が作成した lc2glb は適用できないと判断した。そこで、マトリックスの節約を行わないフルマトリックスでの全体マトリックスを作成するプログラムとして gmatrix を新しく作成した。

bound のソースコードを以下に示した。

リスト A.8: bound のソースコード

```

1 SUBROUTINE BOUND(NOPBV,NOPBVN,BV,BVN,A,B,IA)
2 COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD
3 COMMON /SIZE/ NP, NE, NHBW,ND,NN
4 DIMENSION IA(NP),B(NP)
5 DIMENSION A(NP,NP),NOPBV(ND),BV(ND),NOPBVN(NN),BVN(NN)
6
7 C B = RIGHT VECTOR 右辺ベクトル
8 CALL RCLEAR(B,NP)
9
10 C 自然境界条件の入力
11 C INPUT NOUEMAN CONDITION TO B
12 DO I=1,NN
13 B(NOPBVN(I)) = B(NOPBVN(I)) + BVN(I)
14 END DO
15
16 c WRITE(6,*)'右辺ベクトルB write by bound'
17 C WRITE(6,850) (I,B(I),I=1,NP)
18
19 C INPUT DIRICHLET CONDITION(BV) TO B
20 C Bに境界条件を足している
21 DO I=1,ND
22 B(NOPBV(I)) = B(NOPBV(I)) + BV(I)
23 C B(IA((N-1)*NFD+L)) = B(IA((N-1)*NFD+L)) + V
24 END DO
25 c WRITE(6,*)'B(I) write by BOUND ND'
26 c WRITE(6,*) (B(I),I=1,NP)
27
28
29 C INPUT 0.0 TO B ROW WITH HAVE DIRICHLET CONDITION ,AND
30 C INPUT 1.0 TO PIVOT
31 C 基本境界条件のある行を0にして対角成分 (ピボット) に1を入れる
32
33 DO I=1,NP
34 DO J=1,ND
35 A(NOPBV(J),I)=0.0
36 A(NOPBV(J),NOPBV(J))=1.0
37 END DO
38 END DO
39
40
41 cc add A*T to right vector with Dirichlet condition
42 C 係数マトリックスAの対称性を保つために右辺に補正を行う
43 C 但し移流拡散を解くときは係数行列は非対称になるからこの補正は不要
44 cc DO 50 I=1,NP
45 cc DO 50 J=1,ND
46 cc IF(I.NE.NOPBV(J)) THEN
47 c WRITE(6,*) I,J,A(I,NOPBV(J))
48 cc B(I)=B(I)-A(I,NOPBV(J))*BV(J)
49 cc A(I,NOPBV(J))=0.0
50 cc ENDIF
51 c WRITE(*,*)'I,J,B(I)',I,J,B(I)
52 cc 50 CONTINUE
53
54 RETURN
55 END

```

bound では有限要素式に自然境界条件と基本境界条件の組み込みを行っている。この操作は3.2.6項で説明したように行っている。

bandsolve のソースコードを以下に示した。

リスト A.9: bandsolve のソースコード

```

1      SUBROUTINE BANDSOLVE (A,B,T)
2      COMMON /DIMS/ NPD, NED, NPED, NFD, NHBWD
3      COMMON /SIZE/ NP, NE, NHBW,ND,NN
4      DIMENSION IA(NP),B(NP),A(NP,NP),T(NP),IP(NP)
5      DIMENSION AA(1000,1000),AT(1000),BB(1000)
6
7      DATA EPS / 1.0E-75 /
8
9      c 係数行列の成分の表示
10     c WRITE(6,*)'A(I,J) write by bandsolve first'
11     c DO I=1,NP
12     c DO J=1,NP
13     cc 動的に行列を改行する。$は改行せずに一文を出力する意味を持つ
14     c WRITE(6,"(1PE9.2,$)") A(I,J)
15     c IF(J .EQ. NP)WRITE(*,*)
16     cc 何も書かないwrite文で改行なしを中断
17     c ENDDO
18     c ENDDO
19
20     c AAやAT、BBは連立一次方程式の計算ができているか確認に使った変数
21     c AAは最初の係数行列の値、BBは最初の右辺ベクトルの値
22     c WRITE(*,*)"AA(I,J) BY BAND FIRST"
23     c DO I=1,NP
24     c DO J=1,NP
25     c AA(I,J)=A(I,J)
26     c END DO
27     c END DO
28
29     c DO I=1,NP
30     c BB(I)=B(I)
31     c END DO
32
33
34     C   ガウスの消去法で連立一次方程式を解く
35
36     c-----
37     C   ピボット選択
38     c-----
39     C   対角成分が0だと途中で成分/0という計算がでて解けなくなるのでそれを回避するために
40     C   予め対角成分（ピボット）が大きくなるように入れ替える
41
42     DO I=1,NP
43     IP(I)=I
44     END DO
45
46     c.... loop on pivot row
47     C   ピボットのある行をループ
48     DO 100 K = 1, NP - 1
49     CC   ピボット入れ換え exchange pivot
50     c
51     L=K
52     XAM=ABS(A(IP(L),K))
53     DO I=K+1,NP
54     IF (ABS(A(IP(I),K)) .GT. XAM) THEN
55     L=I
56     XAM=ABS(A(IP(L),K))
57     c WRITE(*,*)'入れ換えた行'
58     c WRITE(*,*)IP(K),IP(L)
59     ENDIF
60     END DO
61
62     IF(L .NE. K) THEN
63     C-----row exchange-----
64     LV=IP(K)
65     IP(K)=IP(L)
66     IP(L)=LV
67     ENDIF
68     cc   ピボットがあまりにも小さい場合は計算を中断する
69     IF (ABS(A(IP(K),K)) .LE. EPS) GO TO 900
70
71
72     c-----
73     c   FORWARD ELIMINATION 前進消去
74     c-----
75
76     C   ある行の対角成分（ピボット）
77     P=A(IP(K),K)
78
79     c..... loop on column to be eliminated
80     C   K行目の係数をピボットで割ってピボットの係数を1にする
81     DO J=K+1,NP
82     A(IP(K),J)=A(IP(K),J)/P
83     END DO
84     c..... loop on row to be modified
85     C   右辺も割る
86     B(IP(K))=B(IP(K))/P
87     C   ピボットより下の成分を引いて消す
88     DO 300 I=K+1,NP
89     C   下の行の成分
90     Q=A(IP(I),K)
91     C   下の行の係数に対して、最初の行の係数に下の行の係数倍して引いて下の行の成分を消去
92     DO J=K+1,NP
93     A(IP(I),J)=A(IP(I),J)-Q*A(IP(K),J)

```

```

94         END DO
95
96 c  右辺も引く
97         B(IP(I))=B(IP(I))-Q*B(IP(K))
98     300         CONTINUE
99     100 CONTINUE
100
101 c      WRITE(6,*)'B(I) write by bandsolve fr '
102 c      WRITE(6,*) (B(M),M=1,NP)
103 c      WRITE(6,*)'B(IP(I)) write by bandsolve fr '
104 c      WRITE(6,*) (B(IP(M)),M=1,NP)
105
106 c-----
107 c      BACK SUBSTITUTION  後退代入
108 c-----
109
110 c.... loop on equations backward
111 c 一番下の解を出す
112     T(NP)=B(IP(NP))/A(IP(NP),NP)
113     DO K = NP-1,1,-1
114         S=B(IP(K))
115 c..... loop on substitution
116     DO J = K+1,NP
117 c 一個上の解は一個上の右辺ベクトルから一個下の解に係数をかけたものを引くと求まる
118         S = S-A(IP(K),J)*DBLE(T(J))
119 c DBLEは倍精度にする組込み関数
120     END DO
121     T(K)=S
122     END DO
123
124 cc 連立一次方程式が解けているか確認
125 cc ATに計算で求めた解Tと係数行列の掛け算をした結果を保存
126 c      DO I=1,NP
127 c          AT(I)=0.0
128 c          DO J=1,NP
129 c              AT(I)=AT(I)+AA(I,J)*T(J)
130 c          END DO
131 c      END DO
132
133 c  もともとの右辺BBと解Tと係数AAの掛け算で求めた右辺ATがどれくらい違うか比較
134 c      WRITE(*,*)"AT(I) BY BANDSOLVE"
135 c      DO I=1,NP
136 c          IF(AT(I)-BB(I) .GT. 1.0E-6)
137 c              WRITE(6,*) AT(I)-BB(I)
138 cc 何も書かないwrite文で改行なしを中断
139 c      ENDDO
140
141     RETURN
142
143     900 CONTINUE
144     WRITE(*,2001)K
145     2001 FORMAT('係数行列は特異 (write by bandsolve)')
146     RETURN
147     END

```

bandsolve は連立一次方程式をガウスの消去法で解くサブルーチンである。gmatrix の箇所でも説明したが、本研究がベースにしていた福井氏 [11] の作成した二次元定常拡散問題の解析プログラムでは係数マトリックスの対称性を利用したメモリの節約が行われている。そのため、連立一次方程式を解くサブルーチン bandsolve でもマトリックスの対称性を利用して計算が行われていた。本研究の取り扱う二次元定常移流拡散問題ではマトリックスが非対称になるため、bandsolve をフルマトリックスで解くように修正した。また、ガウスの消去法を行う場合は対角成分であるピボットが0やそれに準じる非常に小さい場合において、計算ができなくなったり桁落ちにより適切な解が求まらない。そこで、42 行目から 67 行目においてはピボットの値が大きくなるように予め行の入れ替えを行っている。このサブルーチン bandsolve については文献 [8, pp.52-69] 及び文献 [9, pp.56-58] の説明とサンプルのソースコードを大いに参考にした。

A.4 入力データ作成支援プログラム melement4 及び melement3 のソースコードの掲載

プログラムの妥当性を確認するために、領域の要素分割数を増やし要素数を増やすと入力に必要なデータも増える。4 要素程度ならば手で入力データを作成することは容易であったが、400 要素ともなると手で入力データを作成することに数時間かかり数値実験を行う上で効率が悪い。そこで、要素の分割数や境界条件値を設定することである程度自動で入力データを作成するプログラムを本研究の解析プログラム fem2dh_steady_AD とは別に作成した。まず、fem2dh_steady_AD を実行するために読み込むファイルの形式を以下の入力ファイルを例として示した。

リスト A.10: 入力ファイルの例

```

1 TEST01:Case File 01 ! テストのタイトル
2     2 ! 材質種類数
3     1.0000E+0 1.0000E+0 1.0000E+0 ! 熱伝導率 密度 定圧比熱容量 の順
4     0.8350E+2 0.7874E+4 4.4713E+2 ! 鉄の物性値
5     4 ! 要素数

```

6	1	1	1	1	4	5	2	！要素番号	材質番号	要素種類番号	要素を構成する節点の番号	の順
7	2	1	1	2	5	6	3					
8	3	1	1	4	7	8	5					
9	4	1	1	5	8	9	6					
10	9	！節点数										
11	1	1.000000	0.000000				1	！節点番号	座標(x,y)	自由度番号	の順	
12	2	1.000000	2.000000				2					
13	3	1.000000	4.000000				3					
14	4	3.000000	0.000000				4					
15	5	3.000000	2.000000				5					
16	6	3.000000	4.000000				6					
17	7	5.000000	0.000000				7					
18	8	5.000000	2.000000				8					
19	9	5.000000	4.000000				9					
20	6	！基本境界条件のかかる節点数										
21	1	300.00	！節点番号	境界条件値	の順							
22	2	300.00										
23	3	300.00										
24	7	200.00										
25	8	200.00										
26	9	200.00										
27	1	！自然境界条件のある節点数										
28	4	200.00	！節点番号	境界条件値	の順							
29	END											

リスト A.10 において、！以降は入力時には無視される。基本的にはリスト A.10 内で！以降で示したような意味の数値の順となっている。なお、自然境界条件を設定しない場合は 27 行目の値を 0 とし、以降の自然境界条件のかかっている節点番号と境界条件値を省略することで実現する。また、6 行目から 9 行目での要素を構成する節点番号は反時計まわりの順番に入力する必要がある。

入力ファイルであるリスト A.10 において、要素種類番号は本研究では四角形要素は 1、三角形要素は 2 という対応をしている。要素の形状が異なれば、節点番号や要素種類番号が異なる。そのため、四角形要素の入力データを作成するプログラムを melement4、三角形要素の入力データを作成するものを melement3 という名称で作成した。これらのプログラムを実行することでリスト A.10 における 5 行目から 28 行目に相当する部分のデータを実行結果として得ることができる。実行によって得られた結果をコピーし、物性値までの入力を済ませた新たな入力データにペーストすることで、入力データの作成を容易となる。このように、実行して得られた結果をコピーペーストすることで新しく入力データを作成することを想定して melement4 及び melement3 を開発した。なお、これらのプログラムは fem2dh_steady_AD と同様に fortran 言語で作成した。プログラムの名称は make element から melement と名付け、後ろの数字は作成する要素の形状を意図した。

melement4 のソースコードを以下に掲載した。

リスト A.11: melement4 のソースコード

```

1  C 二次元領域全体で一様な四角形要素（全て同じ要素）のデータを作るプログラムです。
2  c 節点番号と要素番号は左下から始まって、下から上についてます。
3
4  c 基本境界条件と自然境界条件についても4辺それぞれについて入力できるようにしたけれど、
5  c 縦と横とが交わる場所では二重に設定してしまうので注意
6  c コンパイルするときは
7  c Linuxはf77 melement4.f -o melement4 cygwin g77 melement4.f -o melement4
8  C PROGRAMED BY K.SENOO ADVISED BY PRO.MATSUMOTO 2012-02-09
9
10      DIMENSION CRD(2,10000) !最大節点数10000
11
12  cc      N=20 !横または縦の要素分割数
13  WRITE(*,*) "横または縦の要素分割数（整数で書く）"
14  READ(*,*) N
15  C 領域の設定
16  cc      XMIN=0.0
17  cc      XMAX=1.0
18  cc      YMIN=0.0
19  cc      YMAX=1.0
20
21      WRITE(*,*) "領域の設定XMIN,YMIN,XMAX,YMAX"
22  READ(*,*) XMIN,YMIN,XMAX,YMAX
23
24  C 基本境界境界条件の設定
25  c 値をあたえないときは-1
26  c      BVXL=0.0 !左端の値
27  c      BVXR=0.0 !右端の値
28  c      BVYL=-1 !下端の値
29  c      BVYU=-1 !上端の値
30
31      WRITE(*,*) "基本境界境界条件の設定,左端,右端,下端,上端の順,与えないときは負の数にする"
32  READ(*,*) BVXL,BVXR,BVYL,BVYU
33
34
35  c 自然境界条件の設定
36  C 値をあたえないときは0.0
37  c      BVNXL=0.0 !左端の値
38  c      BVNXR=0.0 !右端の値
39  c      BVNYL=0.0 !下端の値
40  c      BVNYU=0.0 !上端の値
41

```

```

42 WRITE(*,*)"自然境界境界条件の設定,左端,右端,下端,上端の順,与えないときは0"
43 READ(*,*) BVNXL,BVNXR,BVNYL,BVNYU
44
45 DX=(XMAX-XMIN)/N!要素の横幅
46 DY=(YMAX-YMIN)/N!要素の縦幅
47 C 要素数
48 WRITE(*,101) N*N,"!要素数"
49 101 FORMAT(I5,A8)
50 C 要素番号 要素材質 要素形状 要素の節点番号 節点番号は左下から反時計回りについている
51 NE=0
52 DO 10 I=1,N
53 DO 20 J=1,N
54 NE=NE+1
55
56 WRITE(*,100) NE,1,1,((I-1)*N+I+J-1),((N+1)*I+J),((N+1)*I+J+1)
57 & ,((I-1)*N+I+J)
58 20 CONTINUE
59 10 CONTINUE
60 100 FORMAT(40I5)
61 C 全節点数の表示
62 WRITE(*,101) (N+1)*(N+1),"!節点数"
63 NP=0
64 C 節点番号と節点の座標 (x,y)
65 DO 30 I=1,N+1
66 DO 40 J=1,N+1
67 NP=NP+1
68 CRD(1,NP)=XMIN+DX*(I-1)
69 CRD(2,NP)=YMIN+DY*(J-1)
70 WRITE(*,200) NP,CRD(1,NP),CRD(2,NP),NP
71 40 CONTINUE
72 30 CONTINUE
73 200 FORMAT(I5,2F10.6,I8)
74
75 c 境界条件の読み込み
76 ND=0
77
78 IF(BVXL .GT. -0.1) ND=ND+N+1
79 IF(BVXR .GT. -0.1) ND=ND+N+1
80 IF(BVYL .GT. -0.1) ND=ND+N+1
81 IF(BVYU .GT. -0.1) ND=ND+N+1
82
83 WRITE(*,"(I5,A16)" )ND,"!基本境界条件数"
84 cc 基本境界条件のある節点と境界条件値
85 DO I=1,NP
86 IF(XMIN .EQ. CRD(1,I)) THEN
87 IF(BVXL .GT. -0.1) THEN
88 WRITE(*,210)I,BVXL
89 ENDF
90 ENDF
91 ENDDO
92
93 DO I=1,NP
94 IF(XMAX .EQ. CRD(1,I)) THEN
95 IF(BVXR .GT. -0.1) THEN
96 WRITE(*,210)I,BVXR
97 ENDF
98 ENDF
99 END DO
100
101 DO I=1,NP
102 IF(YMIN .EQ. CRD(2,I)) THEN
103 IF(BVYL .GT. -0.1) THEN
104 WRITE(*,210)I,BVYL
105 ENDF
106 ENDF
107 END DO
108
109 DO I=1,NP
110 IF(YMAX .EQ. CRD(2,I)) THEN
111 IF(BVYU .GT. -0.1) THEN
112 WRITE(*,210)I,BVYU
113 ENDF
114 ENDF
115 END DO
116
117 210 FORMAT(I5,F10.2)
118
119 C 自然境界条件の読み込み
120 NN=0
121 IF(BVNXL .NE. 0.0) NN=NN+N+1
122 IF(BVNXR .NE. 0.0) NN=NN+N+1
123 IF(BVNYL .NE. 0.0) NN=NN+N+1
124 IF(BVNYU .NE. 0.0) NN=NN+N+1
125
126 WRITE(*,"(I5,A16)" )NN,"!自然境界条件数"
127 cc 自然境界条件のある節点と境界条件値
128 DO I=1,NP
129 IF(XMIN .EQ. CRD(1,I)) THEN
130 IF(BVNXL .NE. 0.0) THEN
131 WRITE(*,210)I,BVNXL
132 ENDF
133 ENDF
134 ENDDO
135
136 DO I=1,NP
137 IF(XMAX .EQ. CRD(1,I)) THEN

```



```

138      IF(BVNXR .NE. 0.0) THEN
139        WRITE(*,210)I,BVNXR
140      ENDIF
141    ENDIF
142  END DO
143
144  DO I=1,NP
145    IF(YMIN .EQ. CRD(2,I))THEN
146      IF(BVNYL .NE. 0.0) THEN
147        WRITE(*,210)I,BVNYL
148      ENDIF
149    ENDIF
150  END DO
151
152  DO I=1,NP
153    IF(YMAX .EQ. CRD(2,I))THEN
154      IF(BVNYU .NE. 0.0) THEN
155        WRITE(*,210)I,BVNYU
156      ENDIF
157    ENDIF
158  END DO
159
160
161  END

```

melement3 のソースコードを以下に示した。

リスト A.12: melement3 のソースコード

```

1  C 二次元領域全体で一様な三角形要素（全て同じ要素）のデータを作るプログラムです。
2  c 三角形の形状は「/」というような形。
3  c 要素番号は左下から始まって、下から上についてます。
4  c 節点番号は「┐」というように左下右下右上、右上左上左下というようにについている。
5
6  c 基本境界条件と自然境界条件についても4辺それぞれについて入力できるようにしたけれど、
7  c 縦と横とが交わる場所では二重に設定してしまうので注意。
8  c コンパイルするときは
9  c Linuxは f77 melement3.f -o melement3
10 c Cygwinは g77 melement3.f -o melement3
11 C PROGRAMED BY K.SENOO ADVISED BY PRO.MATSUMOTO 2012-02-22
12
13      DIMENSION CRD(2,10000) !最大節点数10000
14
15  c      N=20 !横または縦の要素分割数
16      WRITE(*,*) "横または縦の要素分割数（整数で書く）"
17      READ(*,*) N
18  C 領域の設定
19  c      XMIN=0.0
20  c      XMAX=1.0
21  c      YMIN=0.0
22  c      YMAX=1.0
23
24      WRITE(*,*) "領域の設定XMIN,YMIN,XMAX,YMAX"
25      READ(*,*) XMIN,YMIN,XMAX,YMAX
26
27  C 基本境界境界条件の設定
28  c 値をあたえないときは-1
29  c      BVXL=1.0 !左端の値
30  c      BVXR=-1.0 !右端の値
31  c      BVYL=0.0 !下端の値
32  c      BVYU=-1 !上端の値
33
34      WRITE(*,*) "基本境界境界条件の設定,左端,右端,下端,上端の順,与えないときは負の数にする"
35      READ(*,*) BVXL,BVXR,BVYL,BVYU
36
37
38  c 自然境界条件の設定
39  C 値をあたえないときは0.0
40  c      BVNXL=0.0 !左端の値
41  c      BVNXR=0.0 !右端の値
42  c      BVNYL=0.0 !下端の値
43  c      BVNYU=0.0 !上端の値
44
45      WRITE(*,*) "自然境界境界条件の設定,左端,右端,下端,上端の順,与えないときは0"
46      READ(*,*) BVNXL,BVNXR,BVNYL,BVNYU
47
48
49      DX=(XMAX-XMIN)/N !要素の横幅
50      DY=(YMAX-YMIN)/N !要素の縦幅
51  C 要素数
52      WRITE(*,101) N*N*2,"!要素数"
53      101 FORMAT(I5,A8)
54  C 要素番号 要素材質 要素形状 要素の節点番号 節点番号は左下から反時計回りにについている
55      NE=0
56      DO 10 I=1,N
57        DO 20 J=1,N
58          NE=NE+1
59          C      NOP(1,I)=((I-1)*N+I+J-1)
60          C      NOP(2,I)=((N+1)*I+J+1)
61          C      NOP(3,I)=((N+1)*I+J+1)
62          C      NOP(4,I)=((I-1)*N+I+J+1)

```

```

63 C      I1=((I-1)*N+I+J-1)
64 C      I2=((N+1)*I+J+1)
65 C      I3=((N+1)*I+J+1)
66 C      I4=((I-1)*N+I+J+1)
67
68      WRITE(*,100) NE,1,2,((I-1)*N+I+J-1),((N+1)*I+J),((N+1)*I+J+1)
69      NE=NE+1
70      WRITE(*,100) NE,1,2,((N+1)*I+J+1),((I-1)*N+I+J),((I-1)*N+I+J-1)
71      20 CONTINUE
72      10 CONTINUE
73      100 FORMAT(40I5)
74 C 全節点数の表示
75      WRITE(*,101) (N+1)*(N+1),"!節点数"
76      NP=0
77 C 節点番号と節点の座標 (x,y)
78      DO 30 I=1,N+1
79          DO 40 J=1,N+1
80              NP=NP+1
81              CRD(1,NP)=XMIN+DX*(I-1)
82              CRD(2,NP)=YMIN+DY*(J-1)
83              WRITE(*,200) NP,CRD(1,NP),CRD(2,NP),NP
84          40 CONTINUE
85      30 CONTINUE
86      200 FORMAT(I5,2F10.6,I8)
87
88 c 境界条件の読み込み
89      ND=0
90
91      IF(BVXL .GT. -0.1) ND=ND+N+1
92      IF(BVXR .GT. -0.1) ND=ND+N+1
93      IF(BVYL .GT. -0.1) ND=ND+N+1
94      IF(BVYU .GT. -0.1) ND=ND+N+1
95
96      WRITE(*,"(I5,A16)" )ND,"!基本境界条件数"
97 cc 基本境界条件のある節点と境界条件値
98      DO I=1,NP
99          IF(XMIN .EQ. CRD(1,I)) THEN
100              IF(BVXL .GT. -0.1) THEN
101                  WRITE(*,210)I,BVXL
102              ENDIF
103          ENDIF
104      ENDDO
105
106      DO I=1,NP
107          IF(XMAX .EQ. CRD(1,I)) THEN
108              IF(BVXR .GT. -0.1) THEN
109                  WRITE(*,210)I,BVXR
110              ENDIF
111          ENDIF
112      END DO
113
114      DO I=1,NP
115          IF(YMIN .EQ. CRD(2,I)) THEN
116              IF(BVYL .GT. -0.1) THEN
117                  WRITE(*,210)I,BVYL
118              ENDIF
119          ENDIF
120      END DO
121
122      DO I=1,NP
123          IF(YMAX .EQ. CRD(2,I)) THEN
124              IF(BVYU .GT. -0.1) THEN
125                  WRITE(*,210)I,BVYU
126              ENDIF
127          ENDIF
128      END DO
129
130      210 FORMAT(I5,F10.2)
131
132 C 自然境界条件の読み込み
133      NN=0
134      IF(BVNXL .NE. 0.0) NN=NN+N+1
135      IF(BVNXR .NE. 0.0) NN=NN+N+1
136      IF(BVNYL .NE. 0.0) NN=NN+N+1
137      IF(BVNYU .NE. 0.0) NN=NN+N+1
138
139      WRITE(*,"(I5,A16)" )NN,"!自然境界条件数"
140 cc 自然境界条件のある節点と境界条件値
141      DO I=1,NP
142          IF(XMIN .EQ. CRD(1,I)) THEN
143              IF(BVNXL .NE. 0.0) THEN
144                  WRITE(*,210)I,BVNXL
145              ENDIF
146          ENDIF
147      ENDDO
148
149      DO I=1,NP
150          IF(XMAX .EQ. CRD(1,I)) THEN
151              IF(BVNXR .NE. 0.0) THEN
152                  WRITE(*,210)I,BVNXR
153              ENDIF
154          ENDIF
155      END DO
156
157      DO I=1,NP
158          IF(YMIN .EQ. CRD(2,I)) THEN

```

```

159      IF(BVNYL .NE. 0.0) THEN
160        WRITE(*,210)I,BVNYL
161      ENDIF
162    ENDIF
163  END DO
164
165  DO I=1,NP
166    IF(YMAX .EQ. CRD(2,I)) THEN
167      IF(BVNYU .NE. 0.0) THEN
168        WRITE(*,210)I,BVNYU
169      ENDIF
170    ENDIF
171  END DO
172
173  END
174

```

これらのプログラムは長方形領域を設定し、領域内で様な四角形及び三角形要素で要素分割した場合の入力データを作成する。長方形領域の境界についても各辺ごとに基本境界条件および自然境界条件をそれぞれ設定できるようにした。なお、連続する辺に境界条件を設定する場合はその接合点で二重に境界条件を設定してしまうため、実行結果をコピーして入力データに貼り付けた後で手動で調整する必要がある。

A.5 fem2dh_steady_AD におけるルーチンでの変数の役割

ソースコードに登場する記号の内、主要なものについてはその意味や役割を表 A.2 にまとめて掲載した。

表 A.2: 記号の意味

記号 意味	表すもの	関連記号, 引数	主な 登場ルーチン
NPMAX 最大節点数	Number of node Points MAX	=NPD	main
NEMAX 最大要素数	Number of Elements MAX	=NED	main
NPEMAX 要素内最大節点数	Number of node Points Element MAX	=NPED	main
NF 空間の次元数	Number of Fields	=NFD	main
NT 各点での自由度	各点での独立な（未知）数の数		main
A(NP,NP) 係数マトリックス			main
B(NP) 右辺ベクトル			main
NS 材質数	Number of Substance		input
RKAPPA(NS) 熱伝導率 κ	入力データから読み込む物性値 κ	入力データ	input
HEATCAP(NS) 定圧比熱容量 c_p	入力データから読み込む物性値 c_p	入力データ	input
DENSITY(NS) 密度 ρ	入力データから読み込む物性値 ρ	入力データ	input
RK(NS) 温度伝導率 k	$k = \kappa / (\rho c_p)$	入力データ	input,ek4nod stab,ek3nod
NE 要素数	Number of Elements	入力データ	input
KMATR(NE) 要素の材質番号	Kind of MATeRial	入力データ,KEL	input
KELMN(NE) 要素の形状番号	Kind of ELeMeNt	入力データ,KMAT	input
NOP(LELM(6),NE) 節点番号	Number Of Points	入力データ	input
NP 節点数	Number of node Points	入力データ	input

記号 意味	表すもの	関連記号, 引数	主な 登場ルーチン
CRD(NFD,NP) 節点の座標	CooRDinates	入力データ	input,loccrd
IA(NP) 節点の自由度番号	degree of freedom	入力データ	input
VN(NFD,NP) 各節点での流速	Verocity Numbers	入力データ	input,locv
ND 基本境界条件数	Number of Dirichlet conditions	入力データ	input,bound
NOPBV(ND) 基本境界条件の節点番号	NOP Basic bound Value	入力データ	input,bound
BV(ND) 基本境界条件値	Basic bound Value	入力データ	input,bound
NN 自然境界条件数	Number of Neumann conditions	入力データ	input,bound
NOPBVN(NN) 自然境界条件の節点番号	NOP Bound Value Neumann	入力データ	input,bound
BVN(NN) 自然境界条件値	Bound Value Neumann	入力データ	input,bound
IA(NP) 節点の自由度番号	degree of freedom	入力データ	input
LELM(6) 要素の形状	要素の頂点の数を記憶		blkdat000 input
KEL 要素種類	Kind of ELement	KELMN(NE)	ekmat
KMAT 材質種類	Kind of MATerial	KMATR(NE)	ekmat
ZCRD(NFD,LELM(KEL)) ある要素の節点の座標	Z CooRDinates	CRD(NFD,NP)	loccrd
ZV(NFD,LELM(KEL)) ある要素の節点の流速	Z Verocity	VN(NFD,NP)	locv
IPT 数値積分の積分点数	Integral PoinT		ek4nod
IX,XY 各方向の積分点	Integral X,Integral Y		ek4nod
DNDX(NFD ,LELM(KEL)) 全体座標系での微分	$\left\{ \frac{\partial N_i^e}{\partial x} \right\} = [J]^{-1} \left\{ \frac{\partial N_i^e}{\partial \xi} \right\}$	CI(2,2),D DNDXI(NFD,4) ZCRD(NFD,3)	ek4nod ek3nod dfshp4,stab
DNDXI(NFD,4) 局所座標系での微分	$\frac{\partial N_i^e}{\partial \xi} \frac{\partial N_i^e}{\partial \eta}$	XI,ETA	dfshp4 dlshp4
AI(IPT,IPT) 数値積分での座標値		IX,IY	ek4nod blkdat000
WI(IPT,IPT) 数値積分での重み	Weight	IX,IY	ek4nod blkdat000
WK(NFD,4) 局所座標系での $\mathbf{K}_e$ の中身	$\frac{\partial N_i^e}{\partial \xi} \frac{\partial N_j^e}{\partial \xi} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial N_j^e}{\partial \eta}$	DNDX(NFD,4)	ek4nod
EK(NFD,LELM(KEL)) 全体座標系の $\mathbf{K}_e$	$k \int \int \left\{ \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right\} d\Omega$	DETJ,WK(NFD,4) WI(IPT,IPT),S RK(KMAT)	ek4nod ek3nod gmatrix
WA(NFD,4) 局所座標系での $\mathbf{A}_e$ の中身	$N_j^e \hat{v}_x \frac{\partial N_i^e}{\partial \xi} + N_j^e \hat{v}_y \frac{\partial N_i^e}{\partial \eta}$	V(NFD),DNDX(NFD,4) RN(NFD,LELM(KEL))	ek4nod

記号 意味	表すもの	関連記号, 引数	主な 登場ルーチン
EA(NFD,LELM(KEL)) 全体座標系での $\mathbf{A}_e$	$\int \int \left\{ N_j^e \hat{v}_x \frac{\partial N_i^e}{\partial x} + N_j^e \hat{v}_y \frac{\partial N_i^e}{\partial y} \right\} d\Omega$	DETJ,WA(NFD,4) V(NFD),WI(IPT,IPT) S,DNDX(NFD,4) RN(NFD,LELM(KEL))	ek4nod ek3nod gmatrix
WA_DE(NFD,4) 局所座標系での $\mathbf{A}_{\delta_e}$ の中身	$\left(\hat{v}_x \frac{\partial N_j^e}{\partial \xi} + \hat{v}_y \frac{\partial N_j^e}{\partial \eta} \right) \left(\hat{v}_x \frac{\partial N_i^e}{\partial \xi} + \hat{v}_y \frac{\partial N_i^e}{\partial \eta} \right)$	V(NFD),DNDX(NFD,4)	ek4nod
EA_DE(NFD, LELM(KEL)) 全体座標系での $\mathbf{A}_{\delta_e}$	$\tau_e \int \int \left\{ \left(\hat{v}_x \frac{\partial N_j^e}{\partial x} + \hat{v}_y \frac{\partial N_j^e}{\partial y} \right) \left(\hat{v}_x \frac{\partial N_i^e}{\partial x} + \hat{v}_y \frac{\partial N_i^e}{\partial y} \right) \right\} d\Omega$	DETJ,WA(NFD,4) V(NFD),WI(IPT,IPT) S,DNDX(NFD,4) RN(NFD,LELM(KEL)) TAU_E	ek4nod ek3nod gmatrix
WK_DE(NFD,4) 局所座標系での $\mathbf{K}_{\delta_e}$ の中身	$\frac{\partial N_i^e}{\partial \xi} \frac{\partial N_j^e}{\partial \xi} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial N_j^e}{\partial \eta}$	DNDX(NFD,4)	ek4nod
EK_DE(NFD, LELM(KEL)) 全体座標系での $\mathbf{K}_{\delta_e}$	$\tau_{LSIC} \int \int \left\{ \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right\} d\Omega$	DETJ,WA(NFD,4) WI(IPT,IPT) S,DNDX(NFD,4) TAU_L	ek4nod ek3nod gmatrix
C(2,2) [J] の成分	$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}$	DNDXI(NFD,4) ZCRD(NFD,4)	dfshp4
DETJ [J]	$ [J] = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta}$	C(2,2)	dfshp4
CI(2,2) [J] ⁻¹ の成分	$[J]^{-1} = \frac{1}{ [J] } \begin{bmatrix} \frac{\partial x}{\partial \eta} & -\frac{\partial y}{\partial \eta} \\ -\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \end{bmatrix}$	C(2,2),DETJ	dfshp4
XI,ETA 局所座標系での横軸, 縦軸	ξ, η	AI(IPT,IPT)	dlshp4
RN(NFD,4) 内挿関数	N_i^e	XI,ETA	shp4
V(NFD) 内挿近似した流速	$\hat{v}_x, \hat{v}_y$	XI,ETA,VE ZV(NFD,LELM(KEL))	shp4,ek3nod
S 三角形要素の面積	$S^e = \frac{1}{2} \begin{bmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{bmatrix}$	ZCRD(NFD,3)	ek3nod
D 三角形要素の面積の二倍	$D^e = 2S^e$	S	ek3nod
TAU_E 安定化パラメータ τ_e	$\tau_e = \frac{h_e}{2 \mathbf{v} } \left(\coth P_e - \frac{1}{P_e} \right)$	RK(NS) VE,HE,PE	stab
TAU_L 安定化パラメータ τ_{LSIC}	$\tau_{LSIC} = \begin{cases} \frac{ \mathbf{v} h_e}{2} \frac{P_e}{3} & (P_e \leq 3) \\ \frac{ \mathbf{v} h_e}{2} & (P_e > 3) \end{cases}$	VE,HE,PE	stab
PE ペクレ数 P_e	$P_e = \frac{ \mathbf{v} h_e}{2k}$	RK(NS),VE,HE	stab
VE 流速の大きさ $ \mathbf{v} $	$ \mathbf{v} = \sqrt{\hat{v}_x^2 + \hat{v}_y^2}$	V(NFD)	stab
HE 要素サイズ h_e	$\begin{cases} 2 \left(\sum_{\alpha=1}^{n_{en}} \left \frac{\hat{v}_i}{ \mathbf{v} } \frac{\partial N_{\alpha}^e}{\partial x_i} \right \right)^{-1} & (\mathbf{v} > 0.001) \\ \text{要素の辺の長さの平均} & (\mathbf{v} \leq 0.001) \end{cases}$	VE,V(NFD) HEN(LELM(KEL)) DNDX(NFD, LELM(KEL))	stab
HEN(LELM(KEL)) 要素の一辺長		ZCRD(NFD, LELM(KEL)),HE	stab

A.6 式に登場する記号

本論文で登場する主要な記号について表 A.3 にまとめて掲載した。

表 A.3: 本論文内で登場する記号の意味

記号	意味	単位	定義や条件
T	温度	K	位置に依存する未知関数 $T(x, y)$
$\hat{T}$	内挿近似した温度	K	$\hat{T} = \sum_{i=1}^{n_{en}} N_i^e T_i^e$
T_i^e	要素 e の要素節点番号 i^* での温度	K	未知定数
x, y	空間座標軸	m	対象領域の位置を示す
x_i^e, y_i^e	要素 e の要素節点番号 i^* での座標値	m	
t	時間軸	s	対象領域の時刻を示す
$\mathbf{v}$	流速度ベクトル	m/s	$\mathbf{v} = (v_x, v_y)$
v_x, v_y	x, y 方向の流速	m/s	各節点で与えられ既知の定数とする
$\hat{v}_x, \hat{v}_y$	内挿近似した流速	m/s	$\hat{v}_x = \sum_{i=1}^{n_{en}} N_i^e v_{xi}^e, \hat{v}_y = \sum_{i=1}^{n_{en}} N_i^e v_{yi}^e$
v_{xi}^e, v_{yi}^e	要素 e の要素節点番号 i^* での流速	m/s	既知の定数
$\mathbf{n}$	単位法線ベクトル	m	$\mathbf{n} = (n_x, n_y)$
Q	熱量	J	
k	温度伝導率 (熱拡散率, 温度拡散率)	m ² /s	$k = \frac{\kappa}{\rho c_p}$
κ	熱伝導率	W/(m·K)	材質内で一定値で等方性する
ρ	密度	kg/m ³	材質内で一定値とする
c_p	定圧比熱容量	J/(kg·K)	材質内で一定値とする
f	発熱率	W/m ³	一定値とする
q_i	熱流速 (Heat Flux)	W/m ²	$q_i = -\kappa \frac{\partial T}{\partial x_i}$
w	重み関数	-	$w = w(x, y)$
$\hat{w}$	Galerkin 法で近似した重み関数	-	$\hat{w} = \sum_{i=1}^{n_{en}} N_i^e w_i^e$
$\tilde{w}$	Petrov-Galerkin 法で近似した重み関数	-	$\tilde{w} = \hat{w} + \tau_e \mathbf{v} \cdot \nabla \hat{w} = \hat{w} + \tau_e v_i \frac{\partial \hat{w}}{\partial x_i}$
w_i^e	要素 e の要素節点番号 i^* での重み	-	任意定数
P_e	Peclet (ペクレ) 数	-	$P_e = \frac{ \mathbf{v} h_e}{2k}$
h_e	要素サイズ	m	$h_e = \begin{cases} 2 \left(\sum_{\alpha=1}^{n_{en}} \left \frac{\hat{v}_i}{ \mathbf{v} } \frac{\partial N_\alpha^e}{\partial x_i} \right \right)^{-1} & (\mathbf{v} > 0.001) \\ \text{要素の辺の長さの平均} & (\mathbf{v} \leq 0.001) \end{cases}$ 本研究では $n_{en} = 3$ または 4
n_{en}	ある要素 e の節点数	-	
τ_e	安定化パラメータ	-	$\tau_e = \frac{h_e}{2 \mathbf{v} } \left(\coth P_e - \frac{1}{P_e} \right)$
τ_{LSIC}	衝撃捕捉項の安定化パラメータ	-	$\tau_{LSIC} = \begin{cases} \frac{ \mathbf{v} h_e P_e}{2} & (P_e \leq 3) \\ \frac{2}{ \mathbf{v} h_e} & (P_e > 3) \end{cases}$
$\mathbf{A}_e$	要素マトリックスでの移流行列	1/s	$\mathbf{A}_e = \int_{\Omega_e} N_j^e \left(\hat{v}_x \frac{\partial N_i^e}{\partial x} + \hat{v}_y \frac{\partial N_i^e}{\partial y} \right) d\Omega$ (行列の i, j 成分を示した)
$\mathbf{D}_e$	要素マトリックスでの拡散行列	1/s	$\mathbf{D}_e = k \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega$ (行列の i, j 成分を示した)
$\mathbf{A}_{\delta_e}$	要素マトリックスでの安定化行列	1/s	$\mathbf{A}_{\delta_e} = \tau_e \int_{\Omega_e} \left(\hat{v}_x \frac{\partial N_j^e}{\partial x} + \hat{v}_y \frac{\partial N_j^e}{\partial y} \right) \left(\hat{v}_x \frac{\partial N_i^e}{\partial x} + \hat{v}_y \frac{\partial N_i^e}{\partial y} \right) d\Omega$ (行列の i, j 成分を示した)

記号	意味	単位	定義や条件
D_{δ_e}	要素マトリックスでの衝撃捕捉行列	1/s	$D_{\delta_e} = \tau_{LSIC} \int_{\Omega_e} \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega$ (行列の i, j 成分を示した)
T_e	要素マトリックスでの温度ベクトル	K	T_i (ベクトルの i 成分を示した)
w_e	要素マトリックスでの重みベクトル	K	w_i (ベクトルの i 成分を示した)
F	熱フラックスベクトル	K	$kg_2 \int_{\Gamma_2} N_i^e d\gamma$ (ベクトルの i 成分を示した)
N_i^e	ある要素 e での内挿関数	-	$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i), \quad i = 1 \sim 4$ (四角形要素が正規化された局所座標系の場合)
L_i^e	ある要素 e での形状関数	-	$L_i^e = a_i^e + b_i^e x + c_i^e y \quad (i = 1 \sim n_{en})$
ξ, η	局所座標系での座標軸	m	$x = \sum_{i=1}^4 N_i^e(\xi, \eta) x_i^e, \quad y = \sum_{i=1}^4 N_i^e(\xi, \eta) y_i^e,$
$[J]$	ヤコビアン	-	$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$
D^e	三角形要素 e の 2 倍の面積	-	$D^e = \begin{bmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{bmatrix}$ $D^e/2$
S^e	三角形要素 e の面積	-	
Ω	領域を表す記号	-	
Ω_e	領域内のある要素 e の領域を表す	-	
Γ	境界を表す記号	-	$\Gamma = \Gamma_1 \cup \Gamma_2$
g_1	第一種境界条件値	K	一定値とする
g_2	第二種境界条件値	K/m	一定値とする
$\coth$	双曲線余接関数 (ハイパボリックコタンジェント)	-	$\coth x = \frac{1}{\tanh x}$
∇	演算子 (ナブラ)	-	$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$
Δ	演算子 (ラプラシアン)	-	$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

A.7 用語集

本論文で登場した専門用語について主要なものについて表 A.4 にまとめた。

表 A.4: 本論文に登場した主要な用語

用語	説明
Galerkin 法	近似関数と同じ基底関数を重み関数に用いる方法。
Green の公式	多変数関数における部分積分の公式のこと。
Petrov-Galerkin 法	近似関数と異なる基底関数を重み関数に用いる方法。
SUPG 法	Upwind Petrov-Galerkin 法を多次元へ拡張したもの。多次元空間での移流拡散問題において流れの上流方向に重みを取ることで解の安定化を図る。
Upwind Galerkin 法	移流拡散問題における解の不安定性の問題に対して、風上差分法と同じになるように拡散係数に人工拡散係数を加える離散化手法。過剰な人工拡散が入ることではまった解が求まる。
Upwind Petrov-Galerkin 法	Upwind Galerkin 法での不具合を解決した手法。一次元定常移流拡散問題において最適な人工拡散係数を加え、全ての項において同じ重み関数 $\tilde{w}$ を適用した方法のこと。
アイソパラメトリック要素	近似関数を近似するときに使った内挿関数と同じ基底関数を座標変換に用いた要素のこと。アイソパラメトリック要素を用いることで、近似関数を決めた時のように、全体座標 x, y もある要素 e の要素節点番号 i^* での座標値 x_i^e, y_i^e と内挿関数をかけたもので表すことができる。
重み付き残差法	微分方程式を近似的に解く手法。ある式に対して近似解を当てはめる。そのときに生じる誤差（残差）に対して、重みをかけて全体で足し合わせて最小化することで適切な解を求める。
重み付き残差方程式	重み付き残差法を適用した式のこと。
ガウスの消去法	連立一次方程式を組織的に解く手法。
ガウス・ルジャンドル法	任意の関数の積分を $-1 \sim +1$ の範囲内で決められた座標での関数値に重みをかけたものの和として求める数値積分法。
基底関数	近似関数を作る時に使う基となる関数。
近似関数	未知関数を別の関数で近似した関数のこと。
形状関数（試験関数）	要素内の面積比または体積比を表す関数。
剛性行列	二階微分項からなる領域積分の行列のことを慣習としてこう呼ぶ。
弱形式	部分積分を施して微分の階数が弱められた式。
節点	要素の端点のこと。
節点番号	節点につける番号。
直接剛性法	要素ごとに求めた値の和を取ることで全体の近似方程式を組み立てること。
内挿関数（補間関数）	物理量の割合を示す重み関数。節点と節点の間のデータを埋める数値を与える関数。領域内で線形に変化する場合は形状関数と等しくなる。なお、補間と内挿は同じ意味の言葉である。
フーリエの法則	熱の拡散についての法則である。熱は温度の高いところ低いところへ移動し、その移動量は温度差に比例するという法則である。
有限要素法	微分方程式を近似的に解く数値解析の手法。数学的には Galerkin 法を用いた重み付き残差法という位置づけになる。領域を要素に分割し、要素ごとに式を立て、領域全体は全要素について足し合わせることで表現する。
有限要素方程式	重み付き残差法と Galerkin 法を適用することで得られる離散化された式のこと。
要素	領域を小領域に分割したもののこと。
要素節点番号	各要素毎につける節点番号。節点番号は任意につけられるため、ある要素についての節点番号の対応を考慮することが困難となる。そこで、式などで表現する際に表記を簡単にするために要素節点番号が用いられる。反時計まわりに番号付けを行わないといけない。
要素番号	要素につける番号。