

Title	Element-Free Galerkin Methodに適応した並列アルゴリズムの開発
Author(s)	渡邊, 正宏
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1141">http://hdl.handle.net/10119/1141</a>
Rights	
Description	Supervisor:松澤 照男, 情報科学研究科, 修士

修士論文

Element-Free Galerkin Method に適応した  
並列アルゴリズムの開発

指導教官 松澤 照男教授

北陸先端科学技術大学院大学  
情報科学研究科情報システム学専攻

渡邊 正宏

1998 年 2 月 13 日

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>Element-Free Galerkin Method</b>	<b>3</b>
2.1	EFGM の概要	3
2.1.1	手順	4
2.2	EFGM における基底関数	4
2.2.1	Moving Least-Squares Method	4
2.2.2	基底関数の微分	8
2.2.3	MSLM における重み関数 (Weight Function)	8
2.3	支配方程式の Galerkin 法による重みつき残差方程式化と離散化	9
2.3.1	1次元 Burgers 方程式	10
2.3.2	Poisson 方程式	12
2.3.3	2次元非圧縮粘性流れのための支配方程式	12
<b>3</b>	<b>予備実験</b>	<b>15</b>
3.1	常微分方程式への適応	15
3.1.1	計算条件と目的	15
3.1.2	結果	16
3.1.3	重み関数による精度の違い	18
3.1.4	基底関数の最高次数の違いによる精度の評価	18
3.2	1次元 Burgers 方程式への適用	19
3.2.1	問題と計算の条件	20
3.2.2	結果	21

3.2.3	台半径と基底関数 . . . . .	21
3.2.4	不均一な節点配置 . . . . .	22
3.2.5	1次元問題に対する結果と問題点 . . . . .	23
3.3	2次元での利用と不規則な節点配置 . . . . .	25
3.3.1	問題と節点配置について . . . . .	25
3.3.2	均一な節点配置による結果 . . . . .	26
3.3.3	不均一な節点配置の影響 . . . . .	27
3.4	可変な台半径の決定について . . . . .	29
3.5	可変な台半径を用いた解析結果 . . . . .	29
3.6	不均一な節点配置による結果 . . . . .	32
<b>4</b>	<b>EFGM の流れへの適応</b> . . . . .	<b>34</b>
4.1	計算条件と圧力について . . . . .	34
4.2	結果 . . . . .	35
4.3	流れ問題に対する EFGM の利用について . . . . .	38
<b>5</b>	<b>EFGM の並列化について</b> . . . . .	<b>39</b>
5.1	EFGM 並列化の目的 . . . . .	39
5.2	並列化の手順 . . . . .	39
5.3	領域の分割について . . . . .	40
5.3.1	EFGM での領域の分割 . . . . .	40
5.3.2	実験と結果 . . . . .	41
5.4	考察 . . . . .	42
<b>6</b>	<b>考察と今後</b> . . . . .	<b>43</b>
6.1	節点配置 . . . . .	43
6.2	負荷分散を考慮した並列処理について . . . . .	45
<b>7</b>	<b>まとめ</b> . . . . .	<b>46</b>

# 目 次

2.1	EFGM の内挿手順 (1 次元の場合)	5
2.2	EFGM の内挿手順 (2 次元の場合)	6
2.3	Weight Function	9
3.1	常微分方程式の解析解	16
3.2	Burgers 方程式の解	21
3.3	基底関数の最高次数 1 $\sup=0.065$	22
3.4	基底関数の最高次数 2 $\sup=0.066$	22
3.5	基底関数の最高次数 1 $\sup=0.1$	23
3.6	基底関数の最高次数 2 $\sup=0.1$	23
3.7	不均一な節点配置による解析解と厳密解	24
3.8	節点配置	24
3.9	節点を加えた場合の解析解	24
3.10	有限要素法のアイソパラメトリック要素	26
3.11	Poisson equation's result	27
3.12	不均一な節点配置	28
3.13	固定な台半径による結果	28
3.14	可変な台半径	30
3.15	可変な台半径作成のアルゴリズム	31
3.16	可変な台半径による結果	31
4.1	正方キャピティ流れの境界条件	35
4.2	Cavity Flow solution EFGM $Re=10$	36
4.3	Cavity Flow solution FEM $Re=10$	37

5.1	FEM の並列計算 2PE の場合 . . . . .	41
5.2	領域の分割 . . . . .	42
6.1	2 次元的にランダムに節点を配置した場合 . . . . .	44
6.2	Stokes 近似の結果 . . . . .	44
6.3	節点を加えた場合 . . . . .	44
6.4	節点を加えた場合の流れ . . . . .	44

# 第 1 章

## はじめに

コンピュータの進歩により、数値解析の分野ではますます大規模な問題を扱うようになってきている。特に複雑な流れ場や構造解析で有限要素法を用いて大規模問題を解析する事が多い。有限要素法は、複雑な構造物や流れの問題でも、その形状や境界条件さえはっきりしていれば、その挙動を高い精度でシミュレートすることが可能である [11]。さらに、超並列計算機を用いて並列処理を行なう事によって、精度良く高速に解を得られるように、その計算手法やアルゴリズムが研究が進められている。

有限要素法は、多くの計算量を必要とするが、基底関数の計算などほとんどが整理された、コンピュータに適した手法であるといえる。ただし、解析モデル、メッシュ分割、計算条件などが正しくなければ精度の良い結果は得られない事が有限要素法の欠点である。特に、メッシュ分割の善し悪しが、結果の善し悪しを直接左右する事が知られている。解析精度の良くなるメッシュの準備などが研究されているが、実際の解析では経験により思考錯誤する事も多いようである。よって、コンピュータの発展により解析の計算速度は向上しているが、計算条件を設定する準備段階に多大な時間が費やされている事になる。今後ますます、問題が大規模化・複雑化していくことが考えられるため、精度の良いメッシュの準備は大きな問題となる。近年研究されている領域の自動分割手法を用いれば 解析実施者は領域の定義のみを行ない、解析に必要な要素分割や格子分割は自動的に生成される事になる。

その一方で、解析の前準備の複雑さを軽減するために、要素分割を必要としないメッシュレス解法の研究が盛んに行なわれている。

その中で、Element-Free Galerkin Method(以下 EFGM) は、節点準備から有限要素解析に整合性よく結び付けることのできるメッシュレス法である。EFGM は、一般の有限要素法に比べ内挿関数の計算に時間が多くかかる。それは、積分点周辺の節点を検索する必要があるためである。積分点毎に検索が行なわれるため、節点数が多いほど時間を必要とする。各積分点毎に、その周辺節点の情報を記憶すれば検索時間を浪費しないが、メモリを消費するのであまり得策ではないと考えられる。

本研究では、EFGM の並列化を最終目標として研究を進める。EFGM を並列化すれば、積分点周辺の検索時間、係数行列の作成時間が短縮されることは、容易に想像されるからである。しかし、それ以前に EFGM が持つ問題点として、自由な節点配置と台半径の関係が問題である事が実験より分かった。EFGM は台半径と呼ばれる距離によって、積分点周辺の節点を選びだしている。この台半径が大きければ節点は多く含まれることになる。しかし、節数があまりに多く含まれると、解が得られる許容誤差を越えてしまう。そこで本論文では、EFGM で台半径を積分点毎に逆に節点数から制御する方法を提案した。この可変な台半径により、固定な台半径と比べて簡単に台半径を決定することができ、許容範囲内で解を得ることが容易となった。本研究では並列化に至る前に、1次元の非線形問題として Burgers 方程式を、2次元の線形問題として Poisson 方程式を、そして2次元の非圧縮流れの問題として Navier-Stokes 方程式と連続の式を EFGM によって解析した。また、合わせて可変な台半径についても実際に適応しながら検証を進めた。最終的な目標である、EFGM の並列化についても2次元非圧縮定常流れを計算する事で、検証を行なった。

EFGM としての並列化は、基底関数の計算の並列化に他ならない。なぜなら、基底関数の計算を除けば、Galerkin 法との計算上の手続きに違いがないからである。ただし、節点配置が自由である事、台半径にて含まれる節点が変わることが、並列計算を行なう上で問題である。さらに、負荷分散という観点から領域分割が重要な問題になる。領域毎の節点密度に応じた領域分割の方法について考える。



## 第 2 章

# Element-Free Galerkin Method

### 2.1 EFGM の概要

ここでは、Element-Free Galerkin Method について、概要、手順に渡って説明する。これまでの有限要素法や差分法などの計算スキームでは、解析前に計算領域全体に節点のその連結である Cell の配置を行なう。正方キャビティ流れ、管内流れなどの単純な流れであれば、以上の前準備は簡単であるが、複雑な形状をもつ流れ場に対しての前準備は、対象に応じて Cell の配置、形状、サイズなどを考慮する必要がある。Cell の配置は解析によって大きく影響する場合がある。よって、解析前の準備に多くの時間を費やすこととなる。

Element-Free Galerkin Method は、有限要素法である Galerkin 法に整合性良く結び付けることのできる Cell を必要としない計算スキームで、節点間の連結情報を必要とせず、計算領域に節点のみを配置することで、計算対象である偏微分方程式を解くことができる。

Element-Free Galerkin Method(以下 EFGM) は、有限要素法である Galerkin 法に整合性良く結び付く。よって解析対象となる支配方程式は重みつき残差方程式として弱形式化される。EFGM が有限要素法と異なる点は基底関数の計算方法である。EFGM では、Moving Least-Squares Method という方法で基底関数を求める。この手法によって、任意の位置にある節点から基底関数を算出することができる。以下より、その基底関数の算出について述べる。

### 2.1.1 手順

EFGM による解析の手順は、次のような順序で行なわれる。

概略を示した図を図 (2.1)(2.2) に示す。図 (2.2)1. のように、解析領域に節点を配置する。節点の配置は任意とする。次に解析領域を Background Cell という規則的な格子状の領域に分割する。(図 (2.2)2.) この Cell は、節点の位置に関係なく分割することができる。この Cell それぞれについて、ガウス積分を行なう。(図 (2.2)3.) 図 (2.2) の最下図でガウス積分の積分点を  $\times$  とすると、以下の Moving Least-Squares Method(以下 MLSM) により、積分点での近似関数値が求められる。(図 (2.2)4.)

- (1) ある積分点を中心に、重み関数の台 (support) の半径に含まれる節点から、それぞれの距離に重み関数によって重みを算出する。
- (2) 重み値とその節点位置を用い、MLSM によって積分点での近似関数値を算定する。この結果、評価点での  $u(= u_I)$  が求められる。
- (3) 節点値に関して、Galerkin 法により作成された重みつき残差方程式を計算する。Galerkin 法を用いるので、重み関数  $W(x)$  の基底関数は、 $u$  の基底関数と同じである。

なお、境界上の扱いについては、節点を境界に沿って配置し有限要素法同様に Dirichlet 型境界条件は、重みつき残差方程式を求める事で最終的に得られる係数行列と、右辺ベクトルの境界と対応した行に代入すればよいし、Neuman 型は重みつき残差方程式として解析すれば良い [5]。以下、基底関数の定式化について説明する。

## 2.2 EFGM における基底関数

### 2.2.1 Moving Least-Squares Method

MSLM とは、以下のような手法である。領域内の任意の評価点  $\boldsymbol{x}$  を、次式のように近似関数  $u^h(\boldsymbol{x})$  として表す。

$$u^h(\boldsymbol{x}) = \sum_j^m p_j(\boldsymbol{x}) a_j(\boldsymbol{x}) \equiv \boldsymbol{p}^T(\boldsymbol{x}) \boldsymbol{a}(\boldsymbol{x}) \quad (2.1)$$

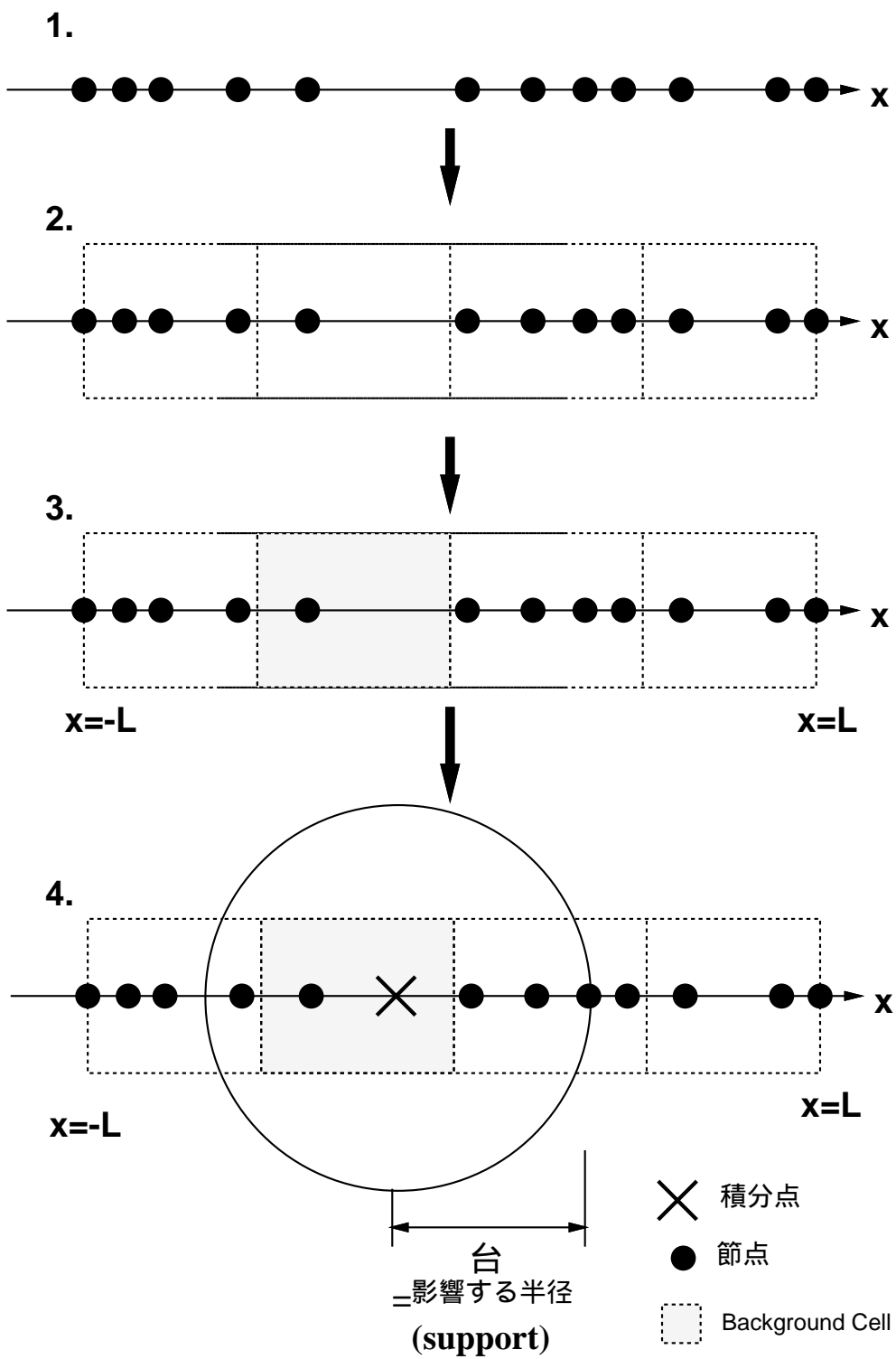


図 2.1: EFGM の内挿手順 (1次元の場合)

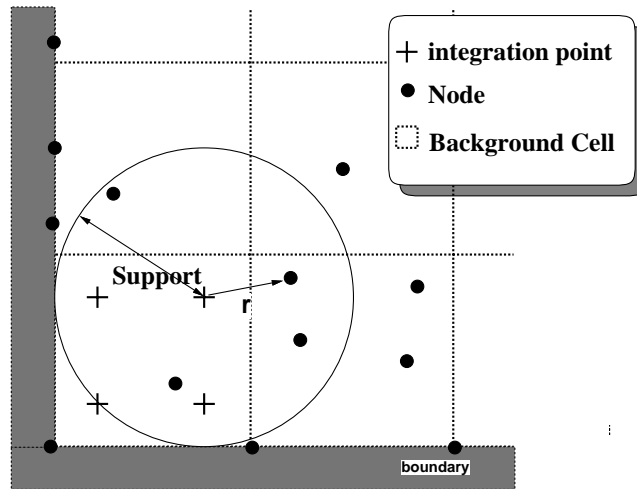


図 2.2: EFGM の内挿手順 (2次元の場合)

ただし、 $p_1(\boldsymbol{x}) = 1$ 、 $p_j(\boldsymbol{x})$  は多項式で、例えば $x$ が1次の ( $\boldsymbol{x} = [x]$ ) 場合、 $m = 2$  で

$$\boldsymbol{p}^T(\boldsymbol{x}) = [1, x] \quad (2.2)$$

$m = 3$  で

$$\boldsymbol{p}^T(\boldsymbol{x}) = [1, x, x^2] \quad (2.3)$$

である。また、 $x$ が2次 ( $\boldsymbol{x} = [x, y]$ ) の場合、 $m = 3$  で

$$\boldsymbol{p}^T(\boldsymbol{x}) = [1, x, y] \quad (2.4)$$

$m = 6$  で

$$\boldsymbol{p}^T(\boldsymbol{x}) = [1, x, y, x^2, xy, y^2] \quad (2.5)$$

である。

係数 $\boldsymbol{a}(\boldsymbol{x})$  は、 $\boldsymbol{x}$  の関数になっている。評価点 $\boldsymbol{x}$ での、係数 $\boldsymbol{a}(\boldsymbol{x})$  を求めるため、以下の評価関数  $J$ を用意する。

$$J = \sum_I^n w(\boldsymbol{x} - \boldsymbol{x}_I) [\boldsymbol{p}^T(\boldsymbol{x}) \boldsymbol{a}(\boldsymbol{x}) - u_I]^2 \quad (2.6)$$

ただし、 $I$ は節点の INDEX、 $w_I$ は点 $\mathbf{x}_I$ の点 $\mathbf{x}$ に対しての重みである。評価関数  $J$ は、節点量  $u_I$ と近似関数  $u^h(\mathbf{x})$  の誤差の重みつき 2 乗 ノルムで、この評価式を最小化する未定係数  $\mathbf{a}(\mathbf{x})$ は、評価式  $J$ の和を行列で表現する事によって、

$$\begin{aligned} J &= [([D]^T \mathbf{a})^T - \mathbf{u}^T][W][([D]^T \mathbf{a}) - \mathbf{u}] \\ &= [\mathbf{a}^T [D][W] - \mathbf{u}^T [W]][([D]^T \mathbf{a}) - \mathbf{u}] \\ &= 2\mathbf{a}^T [[D][W][D]^T \mathbf{a} - [D][W]\mathbf{u}] \end{aligned}$$

ただし、

$$[D] = [p(\mathbf{x}_1), p(\mathbf{x}_2), \dots, p(\mathbf{x}_n)] \quad (2.7)$$

$$[W] = \text{diag}(w(\mathbf{x} - \mathbf{x}_1), w(\mathbf{x} - \mathbf{x}_2), w(\mathbf{x} - \mathbf{x}_3), \dots, w(\mathbf{x} - \mathbf{x}_n)) \quad (2.8)$$

$$\mathbf{u}^T = [u_1, u_2, \dots, u_n] \quad (2.9)$$

ここで、 $J = 0$  とし、 $\mathbf{a}$ について解くと、

$$\mathbf{a} = [A]^{-1}[B]\mathbf{u} \quad (2.10)$$

ただし

$$[A] = [D][W][D]^T \quad (2.11)$$

$$[B] = [D][W] \quad (2.12)$$

(3) 式を (1) 式に代入すると、以下のように近似式が表せる。

$$\mathbf{u}^h = \sum_I^n \sum_j^m p_j(\mathbf{x})([A(\mathbf{x})]^{-1}[B(\mathbf{x})])_{jI} u_I \quad (2.13)$$

以上の式 (2.13) によって、積分点における値を内挿する事ができる。この基底関数の微分については、以下の節にて述べる。

## 2.2.2 基底関数の微分

基底関数の 1 次微分は、式 (2.13) の右边を  $[A][B]$  が空間座標の関数であることに注意して以下の様に求める [1][3]。

$$\frac{du^h(x)}{dx} = \left( \frac{d\{\mathbf{p}\}^T}{dx} [A]^{-1} [B] + \{\mathbf{p}\}^T \frac{d[A]^{-1}}{dx} [B] + \{\mathbf{p}\}^T [A]^{-1} \frac{d[B]}{dx} \right) \{u\} \quad (2.14)$$

なお、 $[A]^{-1}$  の微分には、次式を用いる。

$$\frac{d[A]^{-1}}{dx} = -[A]^{-1} \frac{d[A]}{dx} [A]^{-1} \quad (2.15)$$

以上は、1 次元の場合となっているが、2 次元には容易に拡張できる。実際に微分されるのは、距離の関数である MSLM の重み関数と、多項式のベクトル  $p$  のみである。2 次元の場合、 $x, y$  で微分すればよい。

## 2.2.3 MSLM における重み関数 (Weight Function)

MSLM では、Weight Function を用い解を滑らかに近似する。この Weight function は、重みつき残差法での重み関数ではない。この Weight function により、有限要素法の基底関数の必要条件、

$$\sum_i^m N_i(\mathbf{x}) = 1 \quad (2.16)$$

が満たされる [5]。Weight function は 3 種類提案 [1] されている。 $\rho$  は台の半径  $x$  は評価点、 $x_I$  は積分点とすると

Conical weight Function:

$$w(x - x_I) = 1.0 - \left( \frac{x - x_I}{\rho} \right)^2 \quad (2.17)$$

Exponential weight function:

$$w(x - x_I) = \frac{e^{-((x-x_I)/c)^2} - e^{-(\rho/c)^2}}{1 - e^{-(\rho/c)^2}} \quad (2.18)$$

Quartic Spline function:

$$w(x - x_I) = 1.0 - 6.0 \left( \frac{x - x_I}{\rho} \right)^2 + 8.0 \left( \frac{x - x_I}{\rho} \right)^3 - 3.0 \left( \frac{x - x_I}{\rho} \right)^4 \quad (2.19)$$

Weight function をグラフに表すと、図 (2.3) のようになる。ただし、Exponential weight function の定数  $c = 0.5$  とした。この Weight function については、それぞれを用いた場合の予備実験を行なう。

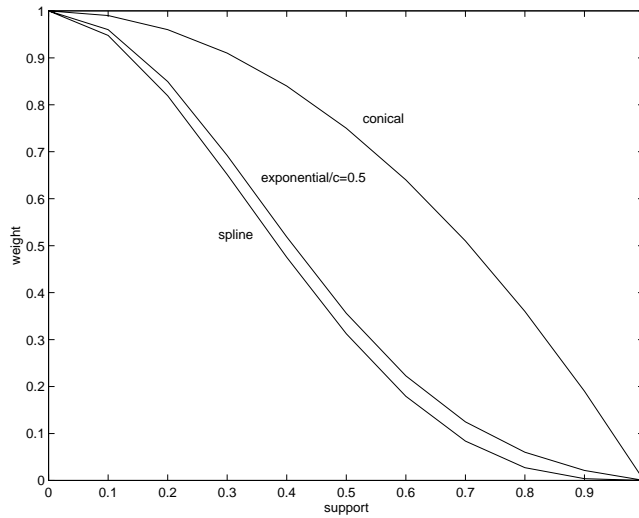


図 2.3: Weight Function

## 2.3 支配方程式の Galerkin 法による重みつき残差方程式化と離散化

前述した通り、EFGM では重みつき残差法によって弱形式化された支配方程式を解析する。ここでは、解析対象である支配方程式を重みつき残差方程式として表し、予備実験と、流れの解析のための準備を行なう。なお、境界条件や厳密解はそれぞれ解析を行なう章で提示する。

以下のような微分方程式系を考える。領域  $D$  で与えられた  $u(x)$  の微分方程式系を

$$\Omega(u) = 0 \quad x \in D \quad (2.20)$$

境界  $B$  における境界条件 (自然境界条件) を

$$\Gamma(u) = 0 \quad x \in B \quad (2.21)$$

とする。これらを Galerkin 法に基づき重みつき残差法を用いて弱形式化を行ない、離散方程式に書き換える。変数  $u$  を以下のように基底関数  $N_i$  を用いて表すと、

$$u = \sum_i N_i u_i \quad (2.22)$$

Galerkin 法により、重み関数  $w$  は式 (2.22) と同じ基底関数となる。

$$w = \sum_j N_j w_j \quad (2.23)$$

この時、解くべき離散方程式は、以下のようになる。

$$\begin{aligned} \int_D \sum_j N_j w_j \boldsymbol{\Omega} \left( \sum_i N_i u_i \right) d\mathbf{D} \\ + \int_B \sum_j N_j w_j \boldsymbol{\Gamma} \left( \sum_i N_i u_i \right) d\mathbf{B} = 0 \end{aligned} \quad (2.24)$$

これが、重みつき残差方程式の一般的な表現である。この離散化した方程式は EFGM で用いるが、有限要素法として用いることも出来る。この部分が EFGM が FEM と整合性よく結び付くメッシュレス法である理由である。これより解析の対象とする、

1. 1次元常微分方程式
2. 1次元 Burgers 方程式
3. 2次元 Poisson 方程式
4. 定常非圧縮粘性流れの支配方程式

についてそれぞれ重みつき残差方程式を求める。なお、1. の常微分方程式の重みつき残差方程式は、参考文献 [1] に記されているので省略する。なお 2. 以降の方程式については、その後の解析のため重み残差方程式は代数化し表記した。

### 2.3.1 1次元 Burgers 方程式

1次元の Burgers 方程式について離散方程式を求める。

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (2.25)$$

重みつき残差方程式を求めると、以下のようになる。

$$A_{ij} u_j^{(n+1)} + (1 - \theta) \Delta t \{ B_{ijk}^x u_j^{(n+1)} + C_{ij} \} u_k^{(n+1)} = A_{ij} u_j^{(n)} - \theta \Delta t \{ B_{ijk} u_j^{(n)} + C_{ij} \} u_k^{(n)} \quad (2.26)$$

ただし、

$$A_{ij} u_j = \int_D \sum_i \sum_j N_i w_i N_j u_j dD \quad (2.27)$$



$$B_{ijk}^x u_j = \int_D \sum_i \sum_j \sum_k N_i w_i N_j u_j \frac{\partial N_k}{\partial x} u_k dD \quad (2.28)$$

$$C_{ij} u_j = \int_D \sum_i \sum_j \frac{\partial N_i}{\partial x} w_i dD \frac{\partial N_j}{\partial x} u_j \quad (2.29)$$

で、 $\theta$ は $\theta = 1$ で陽解法、 $\theta = 0$ で陰解法、 $\theta = \frac{1}{2}$ でCrank-Nicolson法となる。 $\nu$ は粘性係数、各行列 $A, B, C$ は全要素について重ね合わせたものを表している。ここで、式(2.26)の非線形項を解くため、Newton-Raphson法を用いる。式(2.26)の右辺項を左辺に移項した次式を $F_i^{(n+1)}$ とする。

$$\begin{aligned} F_i^{(n+1)} &= A_{ij} u_j^{(n+1)} + (1 - \theta) \Delta t \{ B_{ijk}^x u_j^{(n+1)} + C_{ij} \} u_k^{(n+1)} \\ &\quad - A_{ij} u_j^{(n)} + \theta \Delta t \{ B_{ijk} u_j^{(n)} + C_{ij} \} u_k^{(n)} = 0 \end{aligned} \quad (2.30)$$

ここで $F_i^{(n+1)}$ の $u_j$ に関する偏微分を求めると

$$J_{ij}^{(n+1)} = \frac{\partial F_i^{(n+1)}}{\partial u_j^{(n+1)}} = A_{ij} u_j^{(n+1)} + (1 - \theta) \Delta t \{ B_{ijk}^x u_j^{(n+1)} + C_{ij} \} \quad (2.31)$$

従って、 $r$ を反復回数とすると次のように反復法が定義できる。

$$J_{ij}^{(n+1),(r)} \Delta u_j = -F_i^{(n)} \quad (2.32)$$

ただし

$$\Delta u_j = u_j^{(n+1),(r+1)} - u_j^{(n+1),(r)} \quad (2.33)$$

以上、式(2.33)を繰り返し用い解くことで反復解を求め、時間進行を行なう。なお、反復解法の初期値はNewton-Raphson法の性質上、解にある程度近い値にとっておく事が必要であり、そうでない場合には発散する恐れもある。今回、初期値は式(2.26)から直接求める事とする。ただし、左辺の移流項の係数は前のステップで得られた $u^{(n)}$ を用いる。従って、

$$\begin{aligned} A_{ij} u_j^{(n+1)} + (1 - \theta) \Delta t \{ B_{ijk}^x u_j^{(n)} + C_{ij} \} u_k^{(n+1),(0)} \\ = A_{ij} u_j^{(n)} - \theta \Delta t \{ B_{ijk}^x u_j^{(n)} + C_{ij} \} u_k^{(n)} \end{aligned} \quad (2.34)$$

となる。

### 2.3.2 Poisson 方程式

Poisson 方程式は、線形の方程式である。以下の、2次元の Poisson 方程式について離散式を求める。

$$\nabla^2 \mathbf{u} = \mathbf{Q} \quad (2.35)$$

1次元 Burgers 方程式同様に Galerkin 法によって

$$C_{ij}^x u_j + C_{ij}^y v_j = \mathbf{Q} \quad \mathbf{Q} = 0 \quad (2.36)$$

Burgers 方程式の拡散項同様に

$$C_{ij}^x u_j = \int_D \sum_i \sum_j \frac{\partial N_i}{\partial x} w_i \frac{\partial N_j}{\partial x} u_j dD \quad (2.37)$$

$$C_{ij}^y v_j = \int_D \sum_i \sum_j \frac{\partial N_i}{\partial x} w_i \frac{\partial N_j}{\partial x} v_j dD \quad (2.38)$$

### 2.3.3 2次元非圧縮粘性流れのための支配方程式

流れのための支配方程式は、Navier-Stokes 方程式と連続の式である。Navier-Stokes 方程式は、非線形性をもつ速度と圧力の3変数の方程式である。2式を無次元化した場合、以下のようなになる。

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = -\text{grad } \mathbf{p} + \frac{1}{Re} \Delta \mathbf{u} \quad (2.39)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2.40)$$

この2式について、Navier-Stokes 方程式を速度成分  $u, v$  に分け、重み残差方程式を作成すると以下の様な3つの式となる。

$$B_{ijk}^x u_j u_k + B_{ijk}^y v_j u_k - E_{ij}^x p_j + \frac{1}{Re} C_{ij}^x u_j = 0 \quad (2.41)$$

$$B_{ijk}^x u_j v_k + B_{ijk}^y v_j v_k - E_{ij}^x p_j + \frac{1}{Re} C_{ij}^y v_j = 0 \quad (2.42)$$

$$G_{ijk}^x u_j + G_{ijk}^y v_j = 0 \quad (2.43)$$

ただし、

$$E_{ij}^x p_j = \int_D \frac{\partial N_i}{\partial x} N_j p_j dD \quad (2.44)$$

$$E_{ij}^y p_j = \int_D \frac{\partial N_i}{\partial y} N_j p_j dD \quad (2.45)$$

$$G_{ij}^x u_j = \int_D N_i w_i \frac{\partial N_j}{\partial x} u_j dD \quad (2.46)$$

$$G_{ij}^y u_j = \int_D N_i w_i \frac{\partial N_j}{\partial y} u_j dD \quad (2.47)$$

これら式 (2.41)、(2.42)、(2.43) を行列の形で表すと、以下のようになる。

$$\begin{bmatrix} B_{ijk}^x u_j + B_{ijk}^y v_j + \frac{1}{Re} C_{ij} & 0 & -E_{ij}^x \\ 0 & B_{ijk}^x u_j + B_{ijk}^y v_j + \frac{1}{Re} & -E_{ij}^y \\ G_{ij}^x & G_{ij}^y & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = 0 \quad (2.48)$$

この式は Burgers 方程式同様、非線形項を含んでいるため Newton-Raphson 法を適用し、反復計算を行なう。そのため、式 (2.31) のように方程式の変数による偏微分を行なう。式 (2.48) を  $F_{(\alpha)} = 0$  とすると、式 (2.33) における  $J_{ij}$  は、( $\alpha$ ) は式 (2.41)=1、(2.42)=2、(2.43)=3 を示す)

$$\begin{bmatrix} \frac{\partial F_{(1)}}{\partial u_j} & \frac{\partial F_{(1)}}{\partial v_j} & \frac{\partial F_{(1)}}{\partial p_j} \\ \frac{\partial F_{(2)}}{\partial u_j} & \frac{\partial F_{(2)}}{\partial v_j} & \frac{\partial F_{(2)}}{\partial p_j} \\ \frac{\partial F_{(3)}}{\partial u_j} & \frac{\partial F_{(3)}}{\partial v_j} & \frac{\partial F_{(3)}}{\partial p_j} \end{bmatrix} \quad (2.49)$$

$$= \begin{bmatrix} B_{ijk}^x u_k + B_{ikj}^x u_k + B_{ikj}^y v_k + \frac{1}{Re} C_{ij} & B_{ij}^y u_k & -E_{ij}^x \\ B_{ijk}^x v_k & B_{ikj}^x u_k + B_{ijk}^y v_k + B_{ikj}^y v_k + \frac{1}{Re} & -E_{ij}^y \\ G_{ij}^x & G_{ij}^y & 0 \end{bmatrix} \quad (2.50)$$

となる。よって、式 (2.33) のように反復解法を行なうことができ、式 (2.41)(2.42)(2.43) を解くことができる。

なお、反復解法の初期値であるが、Newton-Raphson 法のため、初期値として正解に近い値を求めなければならない。今回、問題の設定として、レイノルズ数を 10 程度の低い

値としている。レイノルズ数が低い場合、Navier-Stokes 方程式の移流項を省き、流れを近似することができる。これを Stokes 近似という。よって Stokes 近似による方程式を解き、反復解法の初期値とする。

Stokes 近似によって支配方程式は以下の 2 式になる。

$$Re \operatorname{grad} p - \nabla^2 \mathbf{u} = 0 \quad (2.51)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2.52)$$

重み残差方程式は以下のようになる。

$$-E_{ij}^x p_j + \frac{1}{Re} (C_{ij}^x u_j + C_{ij}^y u_j) = 0 \quad (2.53)$$

$$-E_{ij}^x p_j + \frac{1}{Re} (C_{ij}^x v_j + C_{ij}^y v_j) = 0 \quad (2.54)$$

$$G_{ij}^x u_j + G_{ij}^y v_j = 0 \quad (2.55)$$

以上の式を、変数について行列化すれば

$$\begin{bmatrix} \frac{1}{Re} C_{ij} & 0 & -E_{ij}^x \\ 0 & \frac{1}{Re} C_{ij} & -E_{ij}^y \\ G_{ij}^x & G_{ij}^y & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = 0 \quad (2.56)$$

以上の連立方程式を解くことで、反復解法の初期値とする。

解析対象とする、全ての支配方程式を離散化した。次章より、EFGM による実験を行なう。

## 第 3 章

### 予備実験

#### 3.1 常微分方程式への適応

EFGM を 1 次元のモデル問題に用いることにより、使用方法や信頼性について検証を行なう。各境界条件に対応できるかについても確認する。

モデル問題として、常微分方程式の境界値問題を解析する [1]。方程式は、

$$\frac{d^2u}{dx^2} = \sin(\pi x) \quad 0 \leq x \leq 1 \quad (3.1)$$

境界条件

$$u(x) = 0 \quad \text{at } x = 0 \quad (3.2)$$

$$\frac{du(x)}{dx} = \frac{1}{\pi} \quad \text{at } x = 1 \quad (3.3)$$

厳密解は、

$$u_{exact} = -\frac{\sin(\pi x)}{\pi^2} \quad (3.4)$$

である。

##### 3.1.1 計算条件と目的

計算の条件は、

1. 節点数 21。  $\Delta x = 0.1$
2. Background Cell 分割数 20。

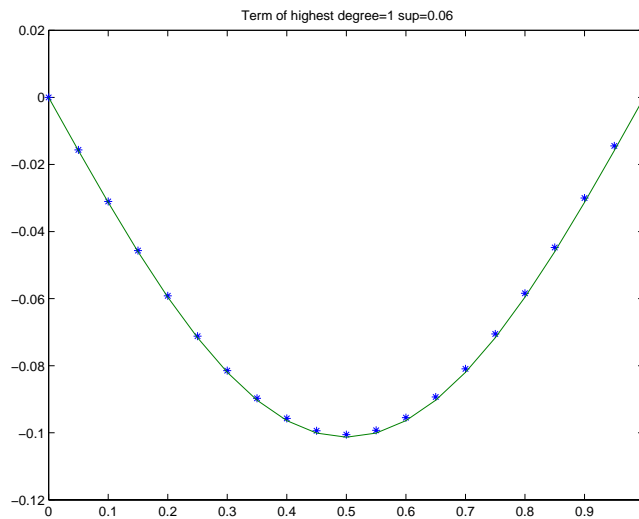


図 3.1: 常微分方程式の解析解

3. 近似関数の最高次数を、1 次、2 次、3 次とする。

4. Weight function を、図 (2.3) の conical, exp, spline の 3 つとする。

とする。条件 1. のように当間隔で節点を配置することを、以下「均一な配置」とする。台半径は、精度が得られるように調整する。

行なう実験は、

1. Weight function による精度の違い
2. 基底関数の最高次数の違いによる精度の違い

の 2 つとする。上述の 2 種類の実験を行なうと共に、Dirichlet 型・Neuman 型の境界条件で計算可能かどうかの確認を行なう。

### 3.1.2 結果

まず解析の例として、厳密解と近似解を図 (3.1) に示す。この時、基底関数の最高次数 1、台半径 0.06 で、MSLM の重み関数は Spline 関数である。図 (3.1) において、直線が厳密解、点が解析解である。表に解析解と、節点毎での絶対誤差を表示する。

表 (3.1.2) 解析解と誤差

$x$	厳密解	EFGM	絶対誤差
0	0	0	0
0.05	-0.0159	-0.0156	0.0002
0.1	-0.0313	-0.0310	0.0003
0.15	-0.0460	-0.0457	0.0003
0.2	-0.0596	-0.0592	0.0004
0.25	-0.0716	-0.0712	0.0004
0.3	-0.0820	-0.0815	0.0005
0.35	-0.0903	-0.0897	0.0006
0.4	-0.0964	-0.0957	0.0006
0.45	-0.1001	-0.0994	0.0007
0.5	-0.1013	-0.1006	0.0008
0.55	-0.1001	-0.0992	0.0008
0.6	-0.0964	-0.0955	0.0009
0.65	-0.0903	-0.0893	0.0010
0.7	-0.0820	-0.0809	0.0010
0.75	-0.0716	-0.0705	0.0011
0.8	-0.0596	-0.0584	0.0012
0.85	-0.0460	-0.0448	0.0012
0.9	-0.0313	-0.0300	0.0013
0.95	-0.0159	-0.0145	0.0014
1.0	0	0.0016	0.0016

この時の最大の絶対誤差は、 $1.6 \times 10^{-3}$ であり、よく近似していることが分かる。

しかし、 $x = 1$ での Neuman 型の境界条件では、誤差が見られる。これは、基底関数の次数を上げることで、誤差を小さくすることが可能である [5]。基底関数の次数を 2 に上げて、解析を行なうと次の表のようになる。

表 (3.1.2) Neuman 型の境界の誤差

$x$	厳密解	EFGM	絶対誤差
1.0	0	0.0006	0.0006

よって、基底関数の次数を上げることで、Neuman 型の境界条件を導入した境界の精度を上げることができる。

### 3.1.3 重み関数による精度の違い

重み関数別に、台半径別に絶対誤差での最大値を表に記した。なお、この時の基底関数の最高次数は 2 である。台半径の調整は、0.01 ずつ行なう。

表 (3.1.3) 重み関数による絶対誤差

台半径	Conical	Exp	Spline
0.09	0.0139	0.0148	0.0157
0.10	0.0049	0.0049	0.0031
0.11	0.0136	0.0135	0.0036
0.12	0.0299	0.0297	0.0066
0.13	0.0114	0.0116	0.0124
0.14	0.0288	0.0279	0.0182

表 (3.1.3) より、もっともよい精度が得られたのが Spline 関数であることが分かる。台半径の調整をさらに行なうことで、どの Weight function でも少しずつ精度を向上できる可能性があるが、以後の実験ではこの問題でもっとも精度の良かった Spline 関数を用いることとする。

### 3.1.4 基底関数の最高次数の違いによる精度の評価

基底関数の最高次数を変えて解析を行なう。ある台半径での最大の絶対誤差を選びだし、表にすると以下のようなになる。重み関数は、Spline 関数である。台半径の調整は、0.01



つつ行なう。

表 (3.1.4) 基底関数の最高次数の違いによる誤差

台半径	1 次	台半径	2 次	台半径	3 次
0.05	0.0013	0.09	0.0148	0.15	0.0016
0.06	0.0016	0.10	0.0049	0.16	0.0039
0.07	0.0060	0.11	0.0135	0.17	0.0167
0.08	0.0081	0.12	0.0297	0.18	0.0447
0.09	0.0101	0.13	0.0116		
0.10	0.0114	0.14	0.0279		

それぞれ多項式の最高次数によって、台半径の範囲が異なるのは、例えば基底関数の最高次数が3次の場合で、解が得られなかった（発散した）場合を調べたところ、積分点に含まれる節点の数が4個以上ない場合があり、その時基底関数の逆行列が精度良く求められておらず、式 (2.13) の行列  $A$  が正則でなくなったためであることが分かった。よってそれぞれの基底関数によって、良い精度が得られる台半径が異なる事となる。各積分点毎に基底関数の最高次数+1個の節点が必要となることを示している。この1次元のモデル問題と計算条件では、基底関数の最高次数が2次の場合、節点を3個含むように台半径を設定する事が難しくなるため、他の基底関数に比べて精度が得られにくかったと考えられる。

また表より、台半径が大きくなるにつれ誤差が大きくなっていることが分かる。よって、台半径は解に大きな影響を及ぼすことが分かる。

## 3.2 1次元 Burgers 方程式への適用

常微分方程式では、各基底関数の精度と Weight function の違い、2つの境界条件での EFGM の精度について検討して来た。EFGM を1次元の非線形の問題に用いることにより、非線形問題に対する EFGM の信頼性を確認する。Burgers 方程式は Navier-Stokes 方程式のもつ非線形性の一面をもつため、モデル方程式としてスキームの検証に使用される。1次元的に不均一な節点配置を行ない、節点配置の影響を見る。

### 3.2.1 問題と計算の条件

以下の境界条件で、式 (2.26) の 1 次元 Burgers 方程式を解く。

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (3.5)$$

初期条件

$$u(x, 0) = \frac{2\pi(\frac{1}{4} \sin(\pi x) + \sin(2\pi x))}{1 + \frac{1}{4} \cos(\pi x) + \frac{1}{2} \cos(2\pi x)} \quad (0 \leq x \leq 1) \quad (3.6)$$

境界条件

$$u(0, t) = u(1, t) = 0 \quad (3.7)$$

厳密解は

$$u(x, t) = \frac{2\pi\nu(\frac{1}{4} \exp(-\nu\pi^2 t) \sin(\pi x)) + \exp(-4\nu\pi^2 t) \sin(2\pi x)}{1 + \frac{1}{4} \exp(-\nu\pi^2 t) \cos(\pi x) + \frac{1}{2} \exp(-4\nu\pi^2 t) \cos(2\pi x)}$$

実験の条件は、以下に基づく。

粘性係数  $\nu = 1.0$

基底関数に用いられる多項式の最高次数を 1 次と 2 次とする。

節点配置は規則的なものとし、節点数は 30 とする。

台半径は誤差の少ないものに調整する。ただし、台半径に含まれる節点数は必ず多項式の最高次数+1 以上である。

Background cell 幅 ( 計算領域を等分割 ) は 40。ただし、節点数  $\leq$  総積分点数。

常微分方程式の場合でもそうであったが、台半径については指定の仕方などは現在のところ限定されていない。よって、節点数と積分点を固定して何度も計算を行ない、もっとも精度が得られるよう調整をする事とする。なお、Background Cell は増やせば精度が向上すると思われる [1]。しかし、節点数の影響を見る事が目的であるため Cell 幅は固定とする。

行なう実験は、

1. 基底関数を 2 種類使用し、それぞれ精度、使用上の利点があるか検証する。

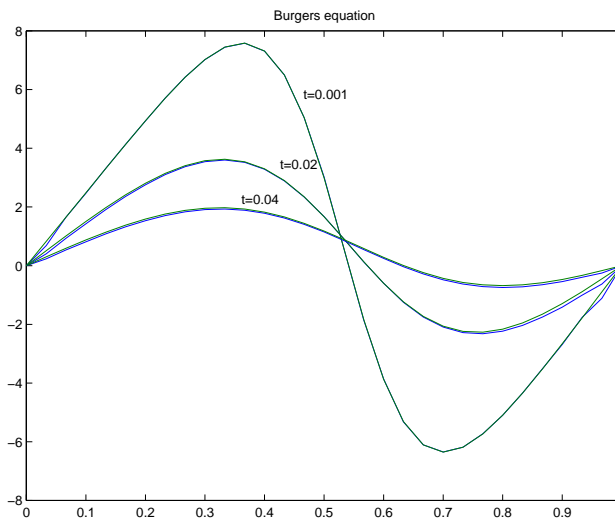


図 3.2: Burgers 方程式の解

2.  $\Delta x$  が一定でない、「不規則な節点配置」について解析を行なう。

非線形方程式でも常微分方程式で行なったように、基底関数の違い・台半径の違いによる性質が現れるか検討する。また、ここで、EFGM 本来の特徴の1つである自由な節点配置について検討する。 $\Delta x$  が一定でない任意の位置への節点配置によって、解がどのように変化するか検討する。

### 3.2.2 結果

結果は図(3.2)となる。各時間ステップにおける解析解と厳密解を表示しているが、境界付近に他の部分に比べて誤差が見られる他は良く一致している。よって、非線形問題に対しても EFGM が解析可能であることが分かる。

### 3.2.3 台半径と基底関数

均一な節点配置で、基底関数の最高次数の違いが、台半径を変化させることでどのように解が変化するか、図(3.3)(3.4)(3.5)(3.6)に示す。また厳密解との相対誤差を調べると表(3.2.3)のようになる。

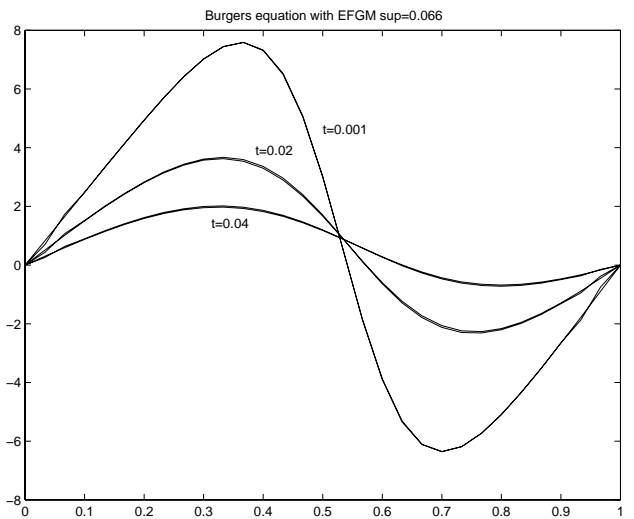
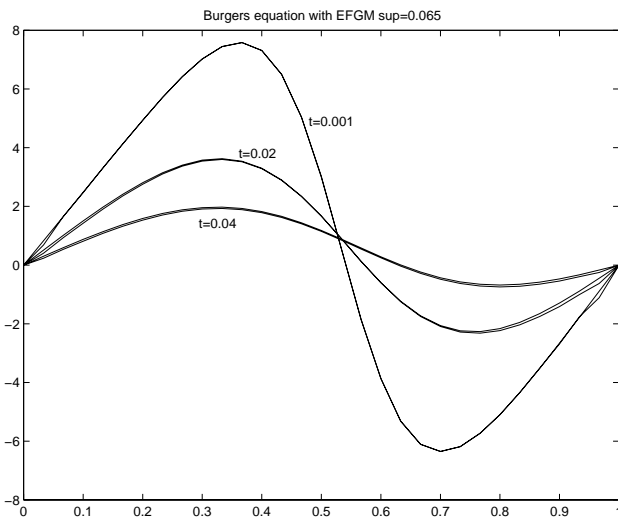


図 3.3: 基底関数の最高次数 1  $\text{sup}=0.065$

図 3.4: 基底関数の最高次数 2  $\text{sup}=0.066$

表 (3.2.3) 基底関数、台半径による誤差

基底関数の最高次数	節点数	台半径	誤差 (最大)
1	31	0.08	0.1640
1	31	0.066	0.1415
2	31	0.1	0.2627
2	31	0.069	0.1989

台半径を調整を行ない、良いと思われる台半径を基底関数の最高次数が 1 次するとき 0.066、2 次するとき 0.069 とした。行なった場合とでは、相対誤差は最高次数が 2 の場合 7 % 程度も異なる事が分かる。

台半径を近似関数が厳密解に良く近似するように調整したものは、基底関数の次数が 1 次 2 次とも良く近似している。しかし、もっとも良い解を導き出せた台半径より大きくすると、解が振動する事が分かる (図 (3.5) 図 (3.6))。この問題のように、節点数の少ない問題であれば台半径の調整は、解を見ながら行なうことが可能であるが、本来目的にしている大規模な問題に対し、解析をおこなうことは難しいと考えられる。

### 3.2.4 不均一な節点配置

EFGM は、節点を自由に配置できる事が特徴である。ここでは、一次元の問題に対して任意に節点配置を行ないその検証をする。図 (3.8) のように、節点配置を  $\Delta x$  が一定で

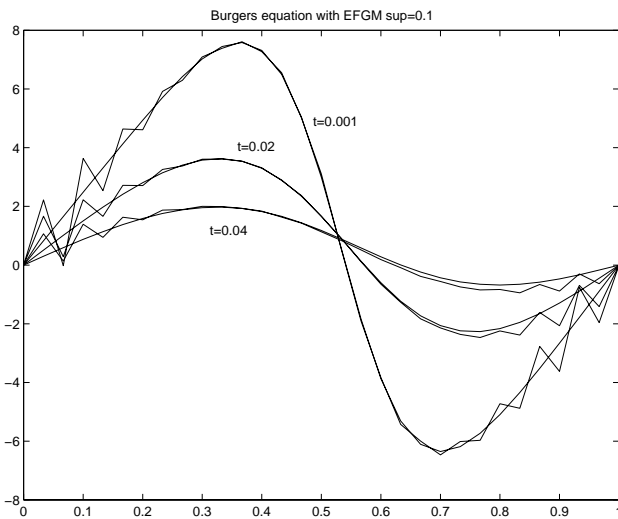


図 3.5: 基底関数の最高次数 1  $\text{sup}=0.1$

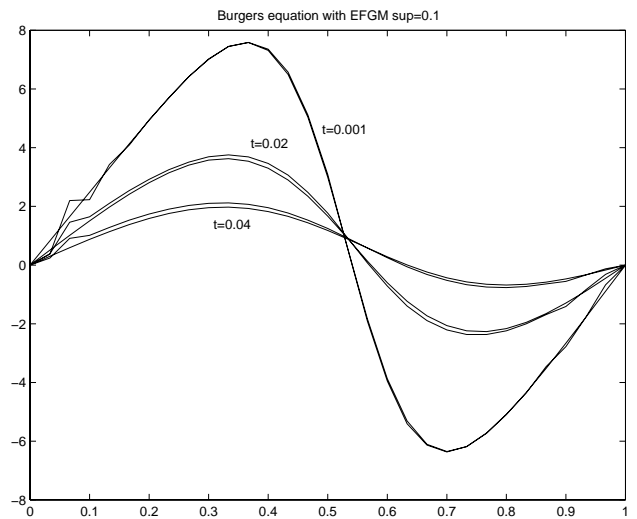


図 3.6: 基底関数の最高次数 2  $\text{sup}=0.1$

ない、すなわち不均一な節点配置によって解析を行なう。結果は、図 (3.7) のようになる。この時の基底関数の最高次数は、1 次である。初期ステップでは良く近似しているが、ステップが進むにつれ近似解が厳密解に一致しない部分が存在する。これは、局所的に節点数が不足していること、各領域で (積分点毎に) 節点が偏っているためだと思われる。しかしながら、節点数 15 個で任意に節点を配置し、非線形問題をある程度の精度で解析できることは、EFGM の特徴の一つと言える。また、この精度の悪い部分に節点をさらに任意に配置し、精度を上げることは可能である。

精度が不足している箇所、 $x = 0.4, 0.4875, 0.56$  に節点を配置すると、解析解は図 (3.2.4) のようになる。図 (3.8) の節点配置と、 $x = 0.4, 0.4875, 0.56$  の 3 節点を新たに加えた場合の  $t = 0.060$  での最大の相対誤差を比べると以下ようになる。

表 (3.2.4) 節点をさらに加えた場合の相対誤差

	加える前	加えた後
相対誤差	0.6359	0.4385

このように、節点を増やすことで精度を上げることは可能である。

### 3.2.5 1次元問題に対する結果と問題点

以上の常微分方程式・1次元 Burgers 方程式の結果より、1次元問題に対する EFGM の性質として以下のことが挙げられる。

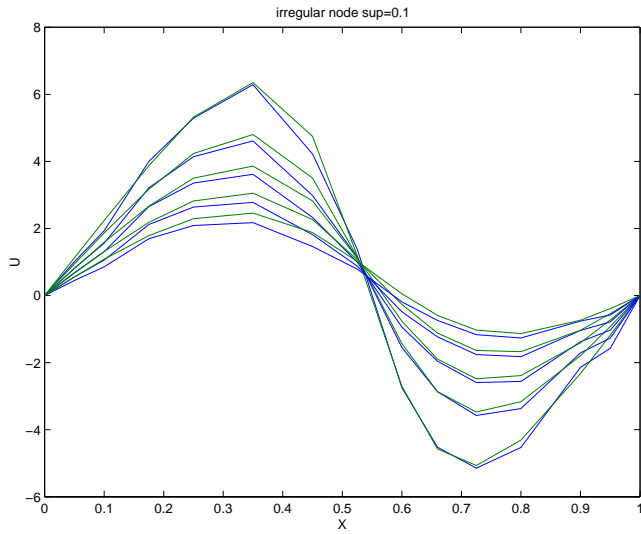


図 3.7: 不均一な節点配置による解析解と厳密解

x	x	x
0.0	0.35	0.725
0.05	0.45	0.8
0.10	0.525	0.9
0.175	0.6	0.95
0.25	0.66	1.0

図 3.8: 節点配置

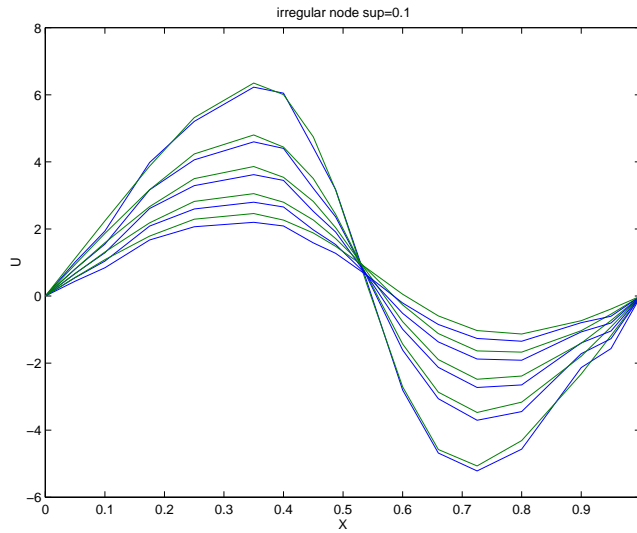


図 3.9: 節点を加えた場合の解析解

- (1) 台半径は節点配置により調整が必要である。台半径が大き過ぎると誤差が大きくなる。
- (2) 節点は自由に配置できるが、解を高精度化するにはある程度節点数を多くしなければならない。

また、(1) と (2) より、節点の密度の偏りが領域毎で大きいと、台半径が大き過ぎる場合がでてくる。1次元の問題ではそれほど影響が見られないが、2次元の場合顕著に現れると思われる。

### 3.3 2次元での利用と不規則な節点配置

#### 3.3.1 問題と節点配置について

1次元で、台半径について調べることができた。さらに Poisson 方程式を解くことで、2次元での台半径の影響、さらに節点配置の影響をみる。

Poisson 方程式については、すでに式 (2.35)、その離散化 (式 (2.36)) は示した通りである。以下に、境界条件を示す。

$$u(x, 0) = \sin(\pi x), u(x, 1) = \sin(\pi x) \exp(-\pi), u(0, y) = u(1, y) = 0 \quad (3.8)$$

厳密解は

$$u(x, y) = \sin \pi x \exp(-\pi y) \quad (3.9)$$

計算条件は

基底関数の最高次数は 2 次

Background cell の分割数は  $20 \times 20$

節点は均一のものとは不均一なものとする。

(1) 均一な節点は  $26 \times 26 = 676$  個

(2) 不均一な節点配置は図 (3.12) の位置とし節点数は  $21 \times 21 = 441$  個。

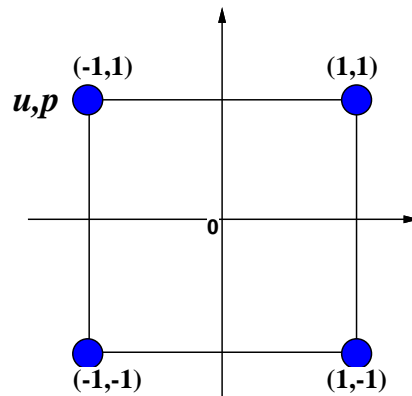


図 3.10: 有限要素法のアイソパラメトリック要素

なお、結果の中で信頼性の確認のため、一般の有限要素法と精度を比べている。有限要素法の要素は isoparametric 要素 [10] を採用した ( 図 3.10 ) 形状関数を、以下に示す。

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \quad i = 1 \sim 4 \quad (3.10)$$

この形状関数を用いた場合の節点座標の写像や導関数の求め方などは [10] などが詳しい。この要素を用いて、FEM の節点配置及び Cell 位置は EFGM と同じとなるようにした。以後 FEM で演算を行なう際にはすべてこの要素を用いる。

### 3.3.2 均一な節点配置による結果

均一な節点配置による結果は、図 (3.11) の様になる。グラフの形状をわかりやすくするために、節点の隣どうしを線で結んだ。FEM で行なった場合と、EFGM で行なった場合の相対誤差を以下に示す。

表 (3.3.2) FEM と EFGM での解析結果

相対誤差	FEM	EFGM
最大値	0.0023	0.1926
平均値	0.0397	0.0629

FEM と比べると劣るものの、相対誤差の平均値を見ても大体同程度で解を近似している事がわかる。



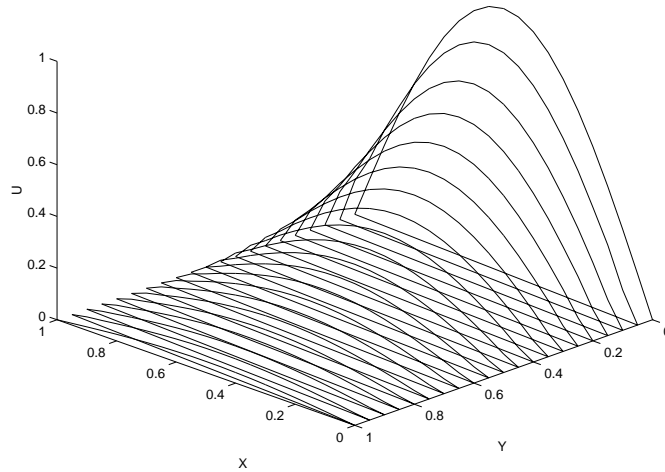


図 3.11: Poisson equation's result

### 3.3.3 不均一な節点配置の影響

図 (3.12) のように、節点を配置した。この節点配置は、 $x, y$  軸方向に意図的に密集した部分と、大きく隣の節点との間を空けたものである。2 次元的にランダムな分散を用いなかったのは、積分点それぞれでの節点の影響をわかりやすくするためである。この節点配置で解析をおこなうと、図 (3.13) のようになる。この時の台半径が 0.15、厳密解との相対誤差の最大値が 5.5646 平均値は 0.4175 である。特に、節点の集中している計算領域上部では、解の振動が大きい。1 次元でも見られたことであるが、台半径が大き過ぎるため、誤差が大きくなる。1 次元の場合よりこの影響は大きく、またこのような不均一な節点配置を行なおうとした場合、台半径の指定が困難であると繰り返し調整を行なわなければならないため、解析の準備のコストが増大する。また、節点数が多くなるにつれ台半径の調整を、より細かく行なわなければならない。そこで、台半径を距離ではなく節点数で積分点毎に調整を行なうことを考える。

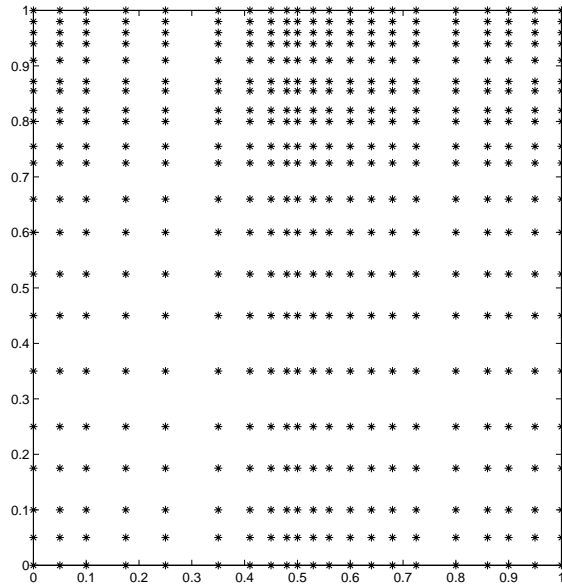


図 3.12: 不均一な節点配置

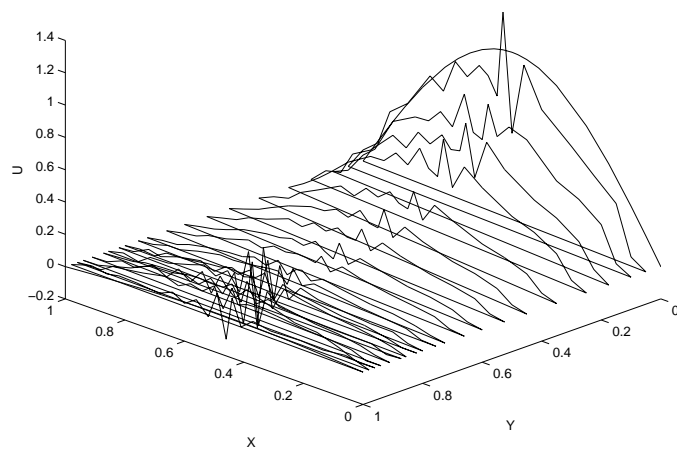


図 3.13: 固定な台半径による結果

### 3.4 可変な台半径の決定について

台半径を、領域の節点数により決めることで簡便に台半径を決定する方法を示す。影響する節点数の上限を ( $limit_n$ ) として、

$$r_{limit_n} = | \text{積分点} - \mathbf{x}_{limit_n} | \quad (3.11)$$

$$r_{limit_{n+1}} = | \text{積分点} - \mathbf{x}_{limit_{n+1}} | \quad (3.12)$$

とすれば

$$support = r_{limit_n} + a \times (r_{limit_n} - r_{limit_{n+1}}) \quad (3.13)$$

ただし  $a$  は定数 ( $0 \leq a \leq 1$ )。

もし、同じ距離に  $limit_n$  個目の節点が存在した場合、

$$limit_n = limit_{n+1} \quad (3.14)$$

として、台半径を決定する。定数  $a$  は、図 (3.14) に示す通り  $limit_n$  と  $limit_{n+1}$  の間隔をどの程度台半径に含めるか決定する。台半径により基底関数を求める際に、Weight Function により節点からの距離で重みづけを行なうが、台半径の端では重みの値が小さくなり、影響が十分にでない。そこで、 $a$  という定数を与えて、積分点からもっとも遠い節点の影響を十分に得ようとするものである。この定数の決定については適当で良いが、あまり大きくとると解の許容範囲内を越えることがあると考えられる。実際の計算を行なう際のアルゴリズムであるが、図 (3.15) に示す。なお、積分点からある節点までの距離の配列へ代入を行なう時は、距離がある値以下であるときのみにとするとソート時間の短縮になる。ある値の決定は節点数にもよるが、ある座標方向に計算領域の  $1/4$  程度として設定すれば、台半径の計算に問題は起きないと思われる。

### 3.5 可変な台半径を用いた解析結果

図 (3.12) の節点配置で可変な台半径を用いた場合の、Poisson 方程式の解を図 (3.16) に示す。この近似解と厳密解との相対誤差を以下の表 (3.5) に示す。

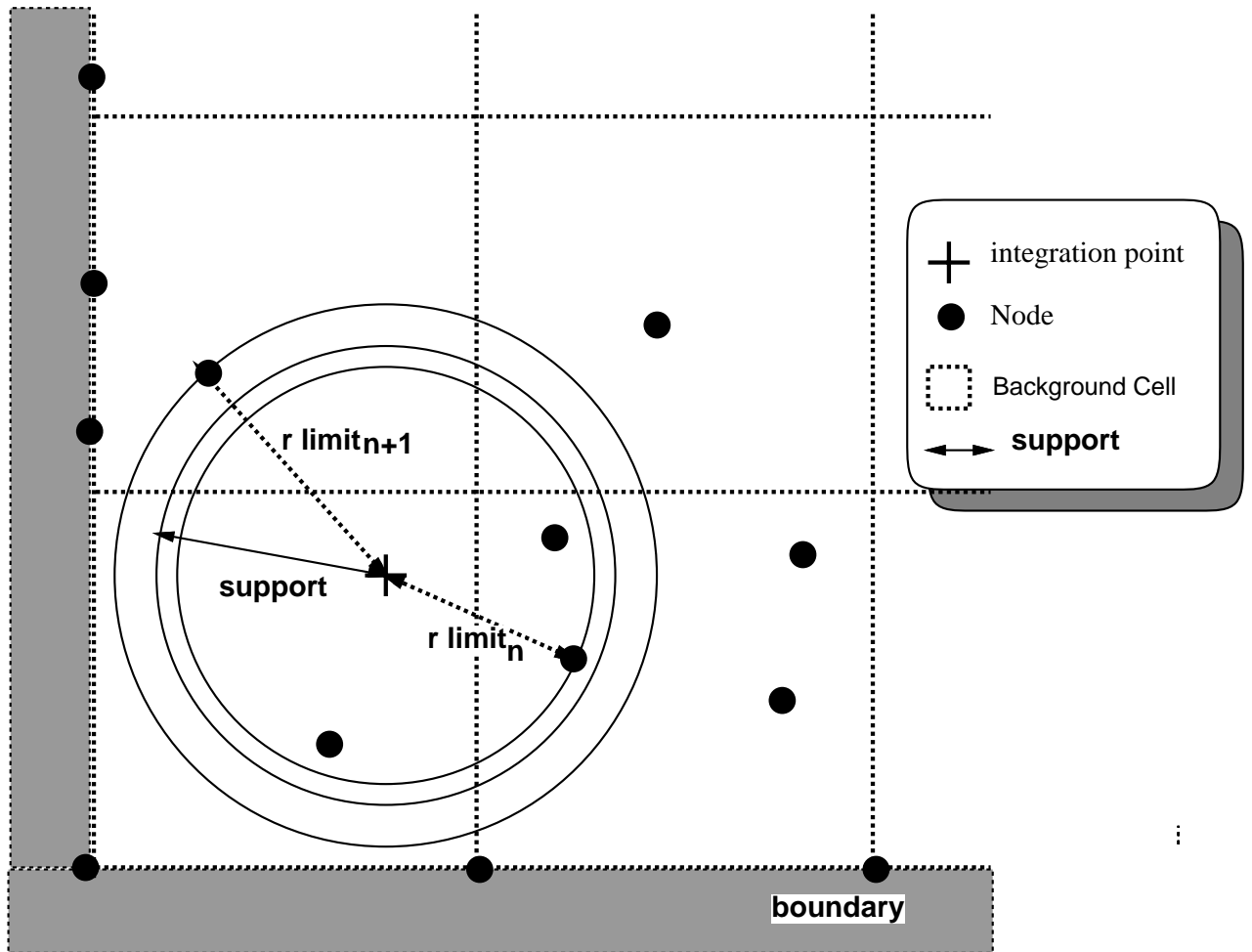


図 3.14: 可変な台半径

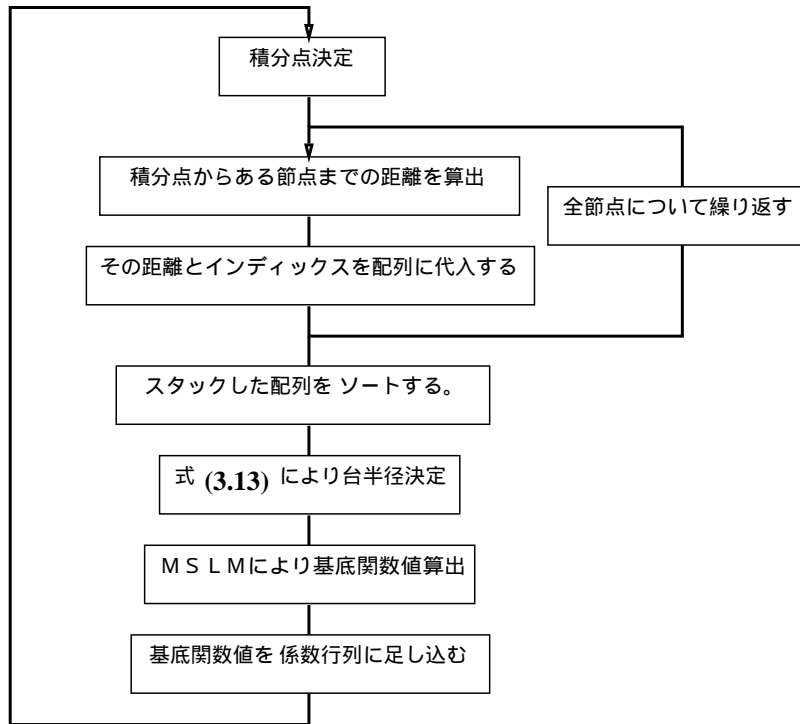


図 3.15: 可変な台半径作成のアルゴリズム

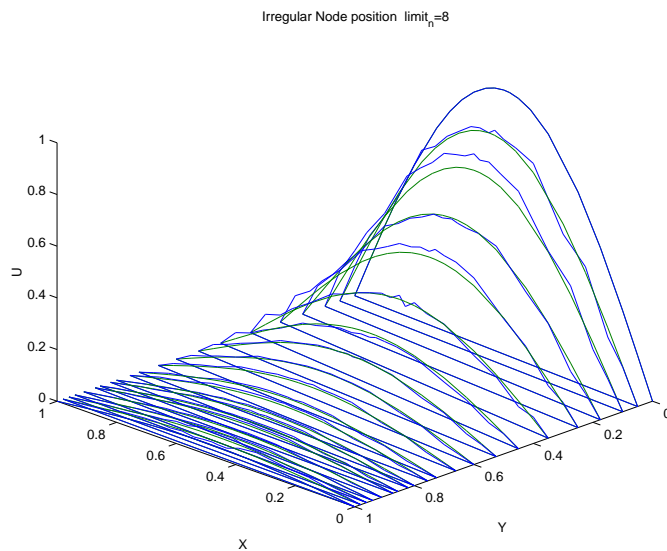


図 3.16: 可変な台半径による結果

表 (3.5) 可変な台半径による相対誤差

相対誤差	台半径固定	可変
最大値	5.5646	0.7572
平均値	0.4175(sup=0.15)	0.0764

台半径を固定し調整したものに比べると、最大値・平均値ともに固定な場合より精度が得られるようになった。この時の個数指定は 8 個、定数  $a = 0.1$  である。

次に、定数  $a$  の影響について試みる。表 (3.5) は、個数指定を 8 ~ 12 としている。

表 (3.5)  $a = 0.1$  の場合

相対誤差	$limit_n = 8$	9	10	11	12
最大値	0.7572	0.9213	1.1173	1.7487	2.4383
平均値	0.0764	0.0879	0.1138	0.1564	0.2364

$a = 0.2$  の場合

相対誤差	$limit_n = 8$	9	10	11	12
最大値	0.7620	0.9311	1.1467	1.8964	2.5167
平均値	0.0771	0.0910	0.1162	0.1676	0.2451

$a = 0.3$  の場合

相対誤差	$limit_n = 8$	9	10	11	12
最大値	0.7670	0.9411	1.1780	1.9892	2.5812
平均値	0.0776	0.0942	0.1186	0.1779	0.2533

表 (3.5) より定数  $a$  が増えることで誤差が増えていることがわかる。この結果は、これまで検討してきた台半径が大き過ぎる事と同じと考えられる。つまり、台半径が大きくなれば誤差が大きくなる。よって、 $a$  については 0.1 程度でよい。

### 3.6 不均一な節点配置による結果

2 次元的に不均一な節点配置について、検討をしてきた。その結果を見る限りは、均一な配置が良いように思われる。しかし、節点配置を均一にできるモデルは稀である上、それでは EFGM の最大のメリットである節点配置の自由さを失う。よって、解の変化が大きい場所に対して節点を適時増やすこと、それ以外の計算領域では均一に配置する方が良

いと考えられる。さらに、不均一な節点配置にとって問題なのは、図 (3.13) を見て理解できるように、節点密度が高いと思われる場所では台半径が大きすぎるため誤差を引き起こしているため精度が良くない点である。そこで、それらを同時に満たすため、可変半径を採用した。その結果、図 (3.12) のように解析上支障となりそうな節点配置においても、一応の結果を保証できる事を示した。

## 第 4 章

# EFGM の流れへの適応

### 4.1 計算条件と圧力について

これまでの予備実験を踏まえて、EFGM を流れに適応する。非圧縮流れを支配する方程式は、Navier-Stokes 方程式と連続の式である。Navier-Stokes 方程式は、予備実験でモデル問題とした、Burgers 方程式同様、非線形の方程式である。よってなんらかの反復計算を必要とする。ここでは、Burgers 方程式同様、Newton-Raphson 法を使用する。Newton-Raphson 法は、反復の初期解がある程度解に近い必要がある。そうでなければ、発散する可能性があるからである。Navier-Stokes 方程式には、流体の粘性を表すレイノルズ数が係数として含まれている。この、レイノルズ数が比較的小さい場合、方程式の移流項 (非線形項) は無視するところができ、圧力と拡散項のみで、線形の方程式として計算を行なうことができる。これを Stokes 近似という。問題の設定ではレイノルズ数は 10 と非常に低い。よって、反復解法の初期値として Stokes 近似した方程式の解を用いる。計算条件、モデル図は以下の図 (4.1) になる。

ここで、一般的な有限要素法における圧力について考える。一般の有限要素法では、基底関数の次数を流速を高次に、圧力を低次に取ると精度が良いとされている [5]。しかし、要素上で内挿点を取ることのできる有限要素法と違い、EFGM では節点が自由に配置され、また積分点毎で所有する節点数も異なり次数を下げる事が難しい。次数を下げると、低次の基底関数が高次の基底関数のための台半径に影響を受けて、誤差の許容範囲を越えてしまうため、解を得ることができない。台半径の大きさが、どの程度解に影響を与えるかはこれまで見てきた通りである。よって、圧力の精度が落ちるが流速と圧力を同じ節点



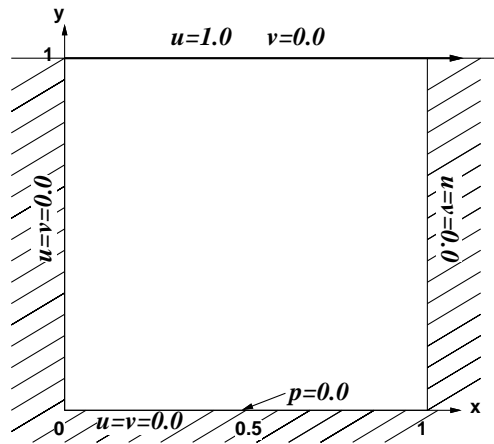


図 4.1: 正方キャビティ流れの境界条件

で設定し、同じ基底関数を用いる事とする。圧力を計算せずに、流速のみを計算するペナルティ関数法 [5] という方法も存在するが、今回は用いなかった。計算条件は、以下の通りである。

節点は、均一な配置とし、 $21 \times 21 = 441$  する。

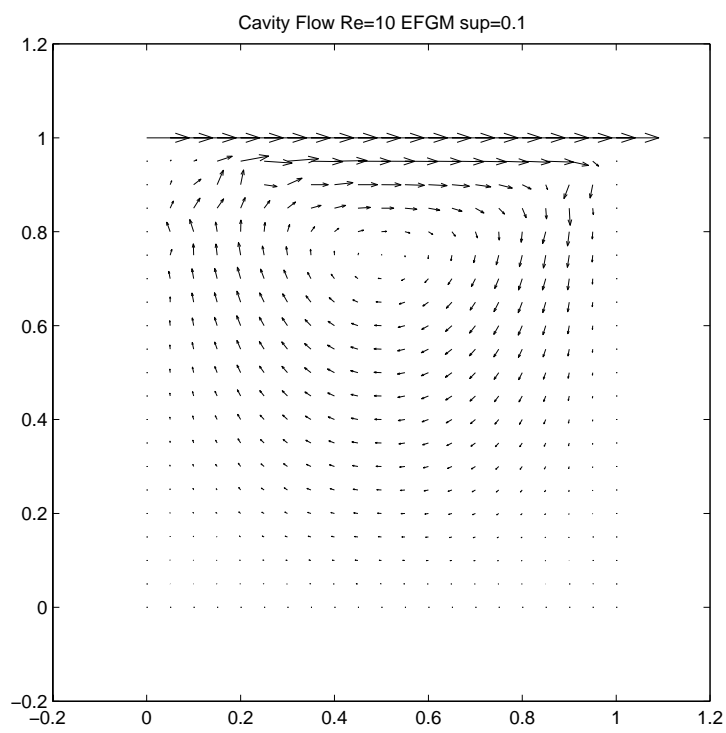
Background cell は  $20 \times 20$  とする。

台半径は固定とする。

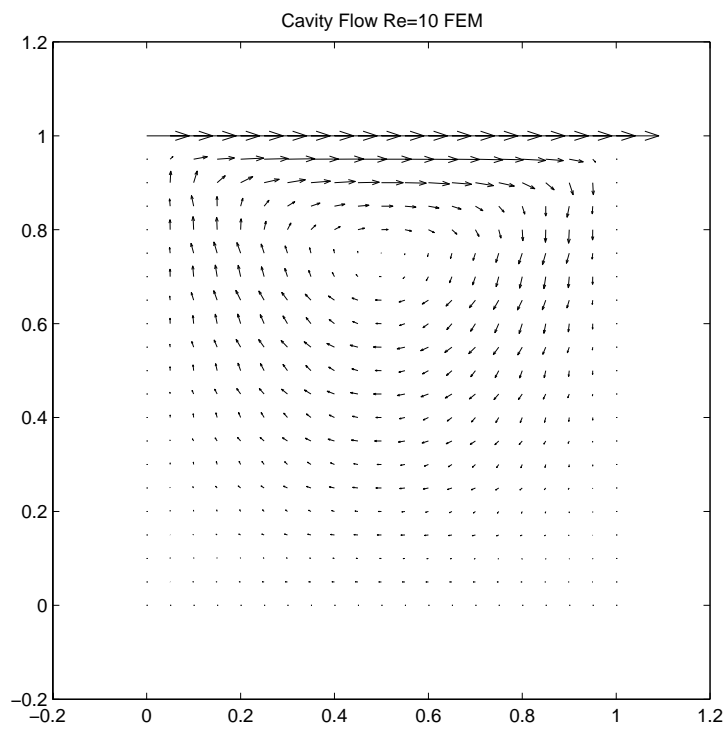
なお、有限要素法では Poisson 方程式で用いた isoparametric 要素を用いて解析を行なう。節点位置、Cell 位置とも上述した計算条件と同じとする。

## 4.2 結果

EFGM での解析の結果を図 (4.2)、FEM での解析結果を図 (4.3) とする。矢印は、速度の大きさと方向を表す。解の比較のため、有限要素法を用いた場合の解も示す。両方の解を比べると Y 座標 0.8 付近で渦が見られ、左上方の流れの他は良く一致していると考えられる。これは、左上方の境界付近の節点数が不足しているためだと思われる。しかし、EFGM によって流れを解析することが可能であることが分かる。



⊗ 4.2: Cavity Flow solution EFGM Re=10



⊠ 4.3: Cavity Flow solution FEM Re=10

### 4.3 流れ問題に対する EFGM の利用について

以上の結果より、EFGM での流れの解析が可能であることが分かった。この解析ではキャビティ流れを扱ったが、有限要素法では Cell を張りにくい境界をもつ問題や、局所的に解が高勾配になる問題に適用し、より EFGM での解析方法について確認、検討してゆく必要がある。

## 第 5 章

# EFGM の並列化について

### 5.1 EFGM 並列化の目的

これまでは、EFGM が実際の問題に適用可能かどうか、確認を繰り返してきた。これより、EFGM の計算時間について検討してゆく。EFGM は有限要素法に比べ、節点の探索に時間を消費する。それは、積分点毎に節点を探索するためである。もし、積分点毎で影響する節点を記録しておく、例えば 2 次元問題で基底関数の次数が 2 次とすると、最低 6 節点の Indx を記録しておくことになる。積分点数を乗じた分の記憶容量を消費するので得策ではないと思われる。非線形の問題では、非線形項を毎ステップ作成しなければならないため、さらに基底関数の計算時間の割合が増大する。このデメリットを解消するため EFGM を並列化することを試みる。

### 5.2 並列化の手順

EFGM の並列化の対象として、前章の定常非圧縮流れを並列化する。なお、問題の簡便化から Stokes 近似とした。以下の手順で並列化を行なう。

1. 節点の位置データを入力する。
2. Background cell に計算領域を分割をする。
3. 各 PE に割り当てた計算領域の節点位置と計算領域境界の情報を分配する。

4. 各 PE で係数行列の作成をする。

5. 各 PE の係数行列の要素とその反復解をメッセージパッシングしながら、反復解を求める。

この論文では、まず 4. 迄を並列化する事を考える。5. はメッセージパッシングする量、計算速度の観点から反復解法によって求めるべきであるが、まず領域の分割方法とその確認のため、直接法である LU 分解によって連立一次方程式を解く。なお、計算はシーケンシャルなマシンで行ない、各領域毎に係数行列を作成してそれを全体行列に合成し連立一次方程式を求める。

### 5.3 領域の分割について

isoparametric 要素を用いた有限要素法の場合、領域分割の仕方は以下のように行なうことが可能である。

1. 計算領域を PE 数で分割する。分割の基準は Cell である。

2. 計算領域境界上の節点についての基底関数の値を式 (5.1) のように定義する。

$$F_{ij} = \frac{pe_1 + pe_2 + \dots + pe_n}{PE_n} \quad (5.1)$$

ただし、 $pe_1, \dots, pe_n$  を各 PE での  $ij$  に対応する係数、 $PE_n$  は領域境界の PE 数である。以上の方法で、有限要素法で定常非圧縮粘性キャビティ流れを Stokes 近似した方程式を並列化し演算を行なうと、図 (5.1) のような演算結果となる。有限要素法では Cell 単位で領域を分割した。各計算領域の境界は、単純に足し合わせて平均化することで解析する事ができた。しかしながら EFGM の場合、台半径によって含まれる節点が変わるため、各計算領域の隣の節点を含んでしまう事が考えられる。そこで、以下のように領域を分割する。

#### 5.3.1 EFGM での領域の分割

領域の分割は、以下のように行なう。

(1) 領域を Background Cell 単位で PE 数に分割する。

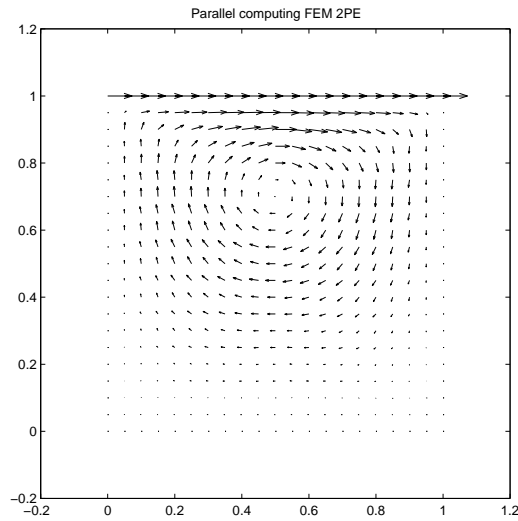


図 5.1: FEM の並列計算 2PE の場合

- (2) 領域境界から、となりの計算領域の台半径分の距離に入る節点を、その PE が探索する節点とする。(2) で得られた他の計算領域の節点については、実際にその領域の計算に使用されなくてもよい。

この領域分割を図 (5.2) に示す。PE1 の受け持つ領域内の節点に加え、PE2 の白い点で表された節点を、PE1 の計算領域の節点として、積分点のたびに検索を行なう。PE2 にとっても同様に黒い節点を自分の領域の節点として検索を行なう。計算領域境界上の節点は、どちらの計算領域にも含まれることになる。領域分割を Background Cell 単位としたのは、ある Background Cell 内の積分点が二つの領域にまたがって存在しないようにするためである。このようにして領域を分割する。

### 5.3.2 実験と結果

考察した領域の分割によって、解析を行なうことができた。基底関数計算の部分は、独立に演算が行なえるためほぼ PE 倍だけ早くなる。しかし、全体の連立方程式の並列化を行っていない上、全体行列作成のため各 PE の係数行列足し込みに時間が費やされている。

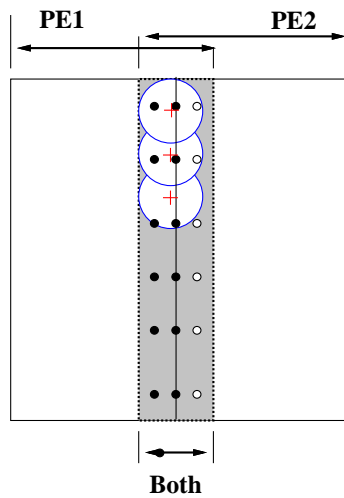


図 5.2: 領域の分割

表 (5.3.2) 領域分割を行なった場合の係数行列作成時間

	係数行列作成時間	全体の計算時間
1PE	229 (sec)	787 (sec)
2PE	89 (sec)	663 (sec)
4PE	42 (sec)	616 (sec)

表 (5.3.2) で 2PE の場合、高速化率が 2 倍以上出ている。これは、各積分点毎で節点を受け持ちの領域全体にわたって検索しているためである。計算領域全体の節点数が 441 個とすると、2PE の場合、積分点毎に全節点数の半分を検索するだけで済むからである。4PE の場合も同様である。

## 5.4 考察

計算領域を領域分割する指針を示すことができた。しかし、反復計算部分が並列化されていないため、この領域分割が引き起こすメッセージパッシング量が効率良く減らせるかなど、これよりさらに検討が必要である。



## 第 6 章

### 考察と今後

#### 6.1 節点配置

EFGM の 1 次元、2 次元問題と非圧縮粘性流れと並列計算について検討を行ってきた。以下で全体を通しての考察と今後の展望を述べる。以上の計算では、節点配置はプログラムによって与えた。しかし、精度の上からも、節点を画像上にマウスなどで置けるようにすると、良い結果を得やすいと思われる。そのうえで、可変な台半径を用いれば、比較的簡単に解析を行なうことが可能である。計算精度の必要な境界付近は、節点を増やすと良いと思われる。その際、台半径を調整する必要がある。もし、台半径を節点密度で調整せず、固定で行なう場合には、解析領域の単位面積あたりの節点密度の違いが、場所によってあまりに大きいと誤差が大きくなり解析を行なえない。単位面積あたりという視点は、積分点との密接な関わりをもっている。今回はこの節点密度の基準について調べていないが、今後調べてみたいと思う。

さて、2 次元の Poisson 方程式の解析では、不均一な節点配置として図 (3.12) を用いたが、以下はランダム関数を用いて節点配置を行ない、定常非圧縮粘性流れの解析を行なった。節点同士が局所的に接近していたり、ほぼ重なってしまうなどの問題が起こったため、この時は、可変な台半径を用いても解を得る事ができなかった。よって、全くランダムな場合で演算を行なうことはできない事が分かった。

均一な節点配置を、節点間の距離が均一であると定義してきたが、モデルの問題に対して、図 (6.3) のように変化の大きい境界周辺に点をさらに配置して解析をおこなうと、図 (6.4) のようになる。均一な配置で得た解析解より流れが FEM で得られた流れに近く

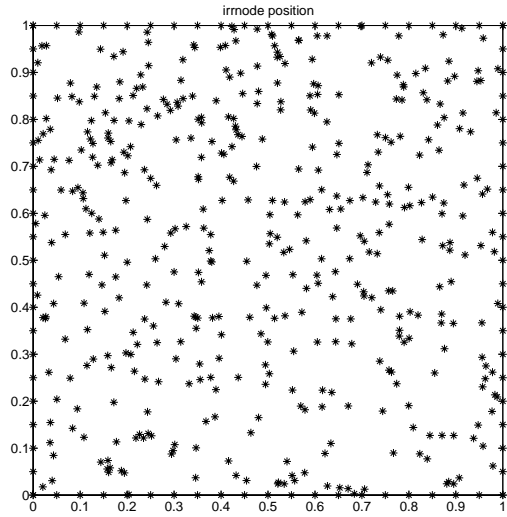


図 6.1: 2 次元的にランダムに節点を配置した場合

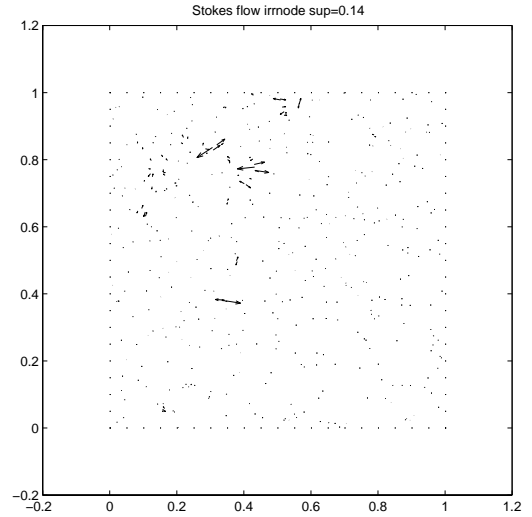


図 6.2: Stokes 近似の結果

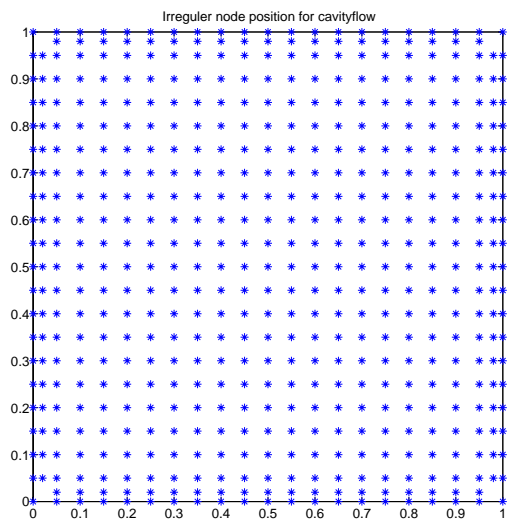


図 6.3: 節点を加えた場合

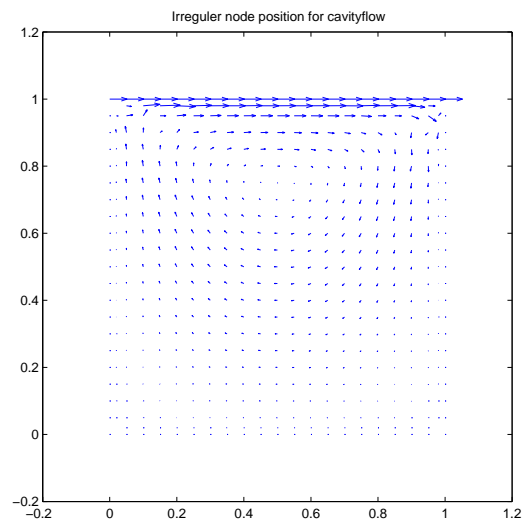


図 6.4: 節点を加えた場合の流れ

なっている事が分かる。この節点を加える作業は、節点をただ配置し、Indx も節点数の増化とともに増やしてつけることで解析が行なえる。有限要素法のように、節点連結情報を書き換える必要がないので、解析前の準備の労力はそれほど必要ない。以上の結果より、節点配置は均一に配置できる領域は均一に境界周辺の高勾配な領域にはそれに応じて節点を配置すると良い。

## 6.2 負荷分散を考慮した並列処理について

今回の領域分割では、負荷分散を考えに入れることができなかった。負荷分散は、台半径を基準として行なうことができると考えている。節点を配置し、固定な台半径、または可変な台半径で計算領域全体を積分点を中心に節点を探索する。そこで得られたスケール(固定な台半径であれば領域毎に含まれる節点数の偏り、可変な台半径であれば台半径の変化)によって、領域を Background Cell 単位で分割する。あまり細かい計算領域の細分化は、メッセージパッシング量を増やすので、Background Cell 単位の計算領域の分配について考えなければならない。今後はこの負荷分散について、検討し検証を行なう予定である。

# 第 7 章

## まとめ

本研究によって以下のことを行なった。

1 次元の常微分方程式と Burgers 方程式、2 次元の Poisson 方程式を解析することで EFGM の性質を示した。また、Burgers 方程式を解析することで非線形問題に対する EFGM の信頼性を示した。

台半径と節点配置との関係を実験的に示した。

節点数で決定する可変な台半径を提案し、不均一な節点配置に対して効果があることを示した。

流れの問題に EFGM を適応し、解析可能である事を示した。

並列計算を行なうための領域分割に対する指標を示した。

## 謝辞

本研究を行なうにあたり、至らない私を丁寧に御指導頂いた松澤照男教授に深く感謝致します。また、日頃から有益な助言や激励を頂いた古山彰一さん、杉山晃一さん、上野博芳さん、また、同室であった大原健一君ほか、3年間の学生生活を共にした松澤研究室の皆様へ感謝致します。

## 参考文献

- [1] 奥田洋司・長嶋利夫・矢川元基, エlementフリーガラーキン法に関する基礎的検討 (第1報) 常微分方程式への適用, 日本機会学会論文集 (A 編)61-590,2302-2308,1995.
- [2] 長嶋利夫・奥田洋司・矢川元基, エlementフリーガラーキン法に関する基礎的検討 (第2報)2次元ポテンシャル問題への適用, 日本機会学会論文集 (A 編)62-599,1746-1753,1996.
- [3] T.Belytschko・Y.Y.LU・L.GU, Element-Free Galerkin Methods,Intr.J.Numerical Methods in Eng.VOL.37,229-256.1994.
- [4] 高橋亮一, 応用数値解析, 朝倉書店,pp.58-60.167-168, Aug. 1993.
- [5] 矢川元基・吉村忍, 有限要素法, 培風館, pp.48-49,53-65.134-146,Sep.1991.
- [6] William H. Press・B.P.Flannery・S.A.Teukolsky・W.T.Vetterling, Numerical Recipes in C【日本語版】, 技術評論社,pp.55-65.278-281,Jun.1993.
- [7] Kenneth H.Huebner・Earl A .Thornton・Red G.Byrom, The Finite Element Method For Engineers third edition,INTERSCIENCE,pp.133-141,.1995.
- [8] 日本機会学会編, 流れの数値シュミレーション, コロナ社,pp.186-192,Sep.1988.
- [9] 数値流体力学編集委員会編, 非圧縮性流体解析, 東京大学出版会,pp.186-207,Feb.1995.
- [10] 戸川隼人, BASIC による有限要素法の基礎, サイエンス社,pp152-159,Feb.1985.
- [11] 東町高雄, 有限要素法のノウハウ, 森北出版株式会社,pp1-6,Mar.1993.
- [12] Ping Tong 著・矢川元基訳, エンジニアのための有限要素法, 共立出版,pp30-31,Sep.1983.